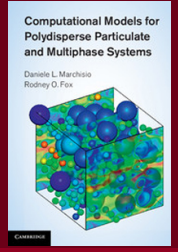
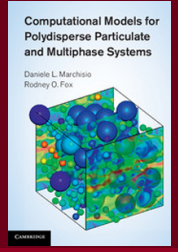


Computational Models for Polydisperse Particulate and Multiphase Systems

Rodney O. Fox
International Francqui Professor

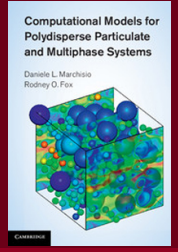
Department of Chemical and Biological Engineering
Iowa State University

February 21 – April 6, 2017



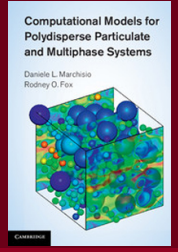
Lecture Topics

1. Introduction to Disperse Multiphase Flows
2. Mesoscale Description of Polydisperse Flows
3. Quadrature-Based Moment Methods
4. Generalized Population Balance Equation
5. Mesoscale Models for Physical and Chemical Processes
6. Solution Methods for Homogeneous Systems



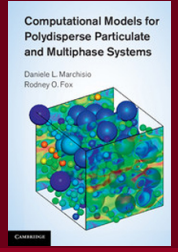
Lecture Topics

7. QBMM for Spatially Inhomogeneous Systems
8. High-Order, Realizable, Kinetic-Based, Finite-Volume Methods
9. Application to Fine-Particle Formation
10. Application to Bubbly Flows
11. Application to Gas–Particle Flows
12. Turbulence Modeling for Disperse Multiphase Flows



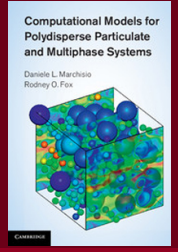
Part 1

INTRODUCTION TO DISPERSE MULTIPHASE FLOWS

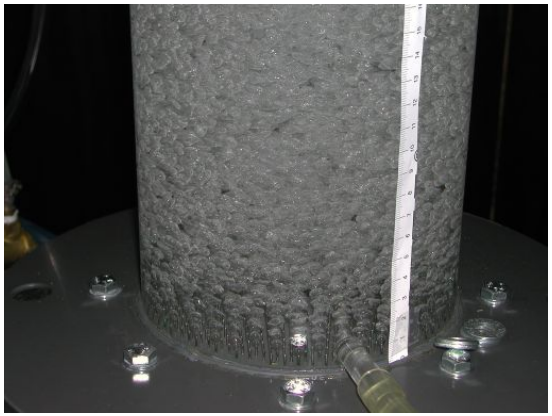


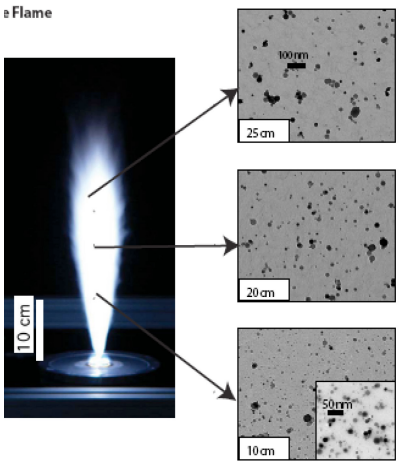
Working Definition of Polydisperse Particulate & Multiphase Systems

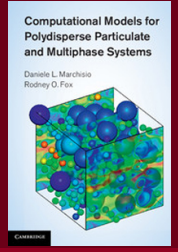
- Systems we consider are constituted by multiple phases of which one phase is **continuous**
- Others are **dispersed**, namely composed of discrete elements
- Discrete elements can be solid particles, droplets or bubbles
- Relevant properties of disperse phase(s), such as mass, momentum, or energy, change from element to element, generating **distributions** or **number density functions**
- Typical examples are crystal size distribution (CSD), particle size distribution (PSD), and particle velocity distribution (PVD)
- Elements of disperse phase(s) continuously evolve due to phase coupling (one-, two-, three- and four-way coupling)



Examples

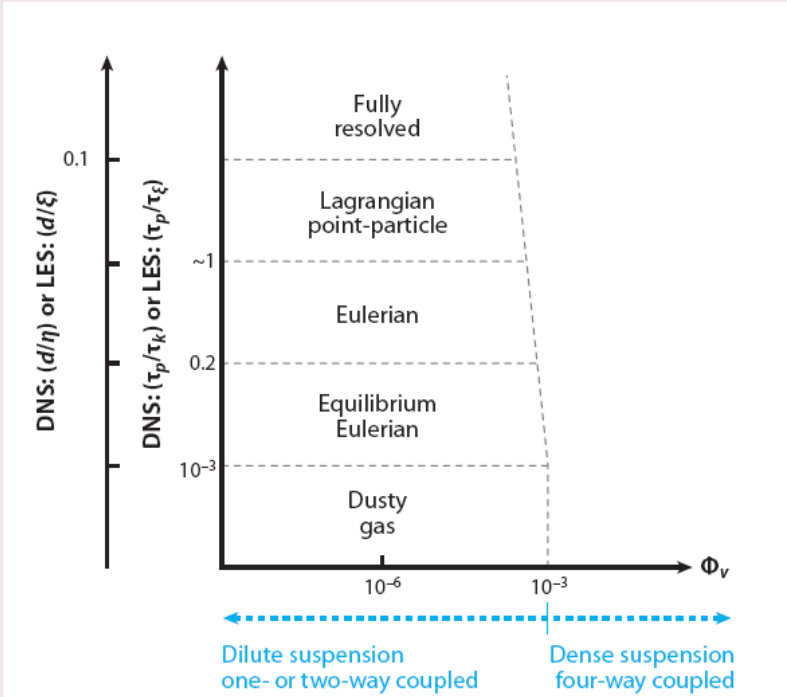




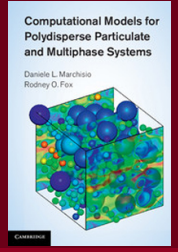


Overview of Different Simulation Methods

Treatment of disperse phase in turbulent fluid

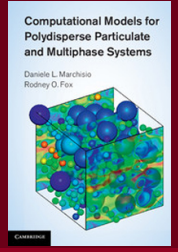


- **Fully resolved approach** (e.g. Discrete element method): see for example work of Subramaniam group at ISU
- **Lagrangian point-particle approach**: see for example work of Soldati group at the Univ. of Udine (Italy)
- **Eulerian approach**: disperse phase is treated as a continuum through concept of number density function



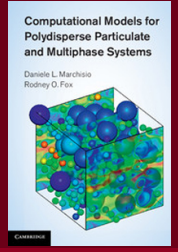
Working Definition of Polydisperse Particulate & Multiphase Systems

- Problem of describing evolution (in space and time) of these systems in Eulerian framework (in terms of number density functions or distributions) has been treated in many ways by different scientific communities
- Crystallization and precipitation (often neglecting spatial inhomogeneities): crystal or particle size, **population balance equation (PBE)**
- Evaporating (and non-evaporating) sprays: droplet surface area, **Williams–Boltzmann equation (WBE)**
- Aerosols and ultra-fine particles: particle mass, **particle dynamics equation (PDE)**
- Particulate systems involved in granular flows: particle velocity, **Boltzmann equation (BE)**
- Although these apparently different theoretical frameworks are referred to by different names, underlying theory (which has its foundation in classical statistical mechanics) is exactly the same!



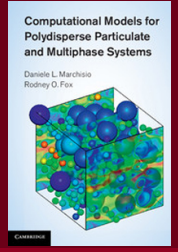
Working Definition of Polydisperse Particulate & Multiphase Systems

- Problem of describing evolution (in space and time) of these systems in Eulerian framework (in terms of number density functions or distributions) has been treated in many ways by different scientific communities
- Crystallization and precipitation (often neglecting spatial inhomogeneities): crystal or particle size, **population balance equation (PBE)**
- Evaporating (and non-evaporating) sprays: droplet surface area, **Williams–Boltzmann equation (WBE)**
- Aerosols and ultra-fine particles: particle mass, **particle dynamics equation (PDE)**
- Particulate systems involved in granular flows: particle velocity, **Boltzmann equation (BE)**
- Although these apparently different theoretical frameworks are referred to by different names, underlying theory (which has its foundation in classical statistical mechanics) is exactly the same!



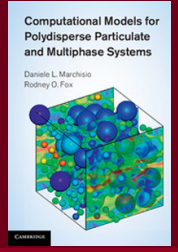
Working Definition of Polydisperse Particulate & Multiphase Systems

- Problem of describing evolution (in space and time) of these systems in Eulerian framework (in terms of number density functions or distributions) has been treated in many ways by different scientific communities
- Crystallization and precipitation (often neglecting spatial inhomogeneities): crystal or particle size, **population balance equation (PBE)**
- Evaporating (and non-evaporating) sprays: droplet surface area, **Williams–Boltzmann equation (WBE)**
- Aerosols and ultra-fine particles: particle mass, **particle dynamics equation (PDE)**
- Particulate systems involved in granular flows: particle velocity, **Boltzmann equation (BE)**
- Although these apparently different theoretical frameworks are referred to by different names, underlying theory (which has its foundation in classical statistical mechanics) is exactly the same!



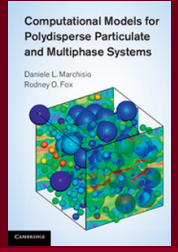
Working Definition of Polydisperse Particulate & Multiphase Systems

- Problem of describing evolution (in space and time) of these systems in Eulerian framework (in terms of number density functions or distributions) has been treated in many ways by different scientific communities
- Crystallization and precipitation (often neglecting spatial inhomogeneities): crystal or particle size, **population balance equation (PBE)**
- Evaporating (and non-evaporating) sprays: droplet surface area, **Williams–Boltzmann equation (WBE)**
- Aerosols and ultra-fine particles: particle mass, **particle dynamics equation (PDE)**
- Particulate systems involved in granular flows: particle velocity, **Boltzmann equation (BE)**
- Although these apparently different theoretical frameworks are referred to by different names, underlying theory (which has its foundation in classical statistical mechanics) is exactly the same!



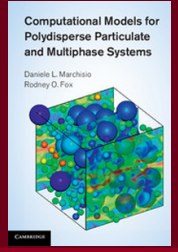
Working Definition of Polydisperse Particulate & Multiphase Systems

- Problem of describing evolution (in space and time) of these systems in Eulerian framework (in terms of number density functions or distributions) has been treated in many ways by different scientific communities
- Crystallization and precipitation (often neglecting spatial inhomogeneities): crystal or particle size, **population balance equation (PBE)**
- Evaporating (and non-evaporating) sprays: droplet surface area, **Williams–Boltzmann equation (WBE)**
- Aerosols and ultra-fine particles: particle mass, **particle dynamics equation (PDE)**
- Particulate systems involved in granular flows: particle velocity, **Boltzmann equation (BE)**
- Although these apparently different theoretical frameworks are referred to by different names, underlying theory (which has its foundation in classical statistical mechanics) is exactly the same!



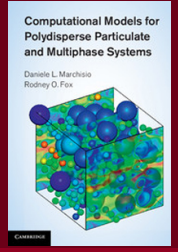
Working Definition of Polydisperse Particulate & Multiphase Systems

- Problem of describing evolution (in space and time) of these systems in Eulerian framework (in terms of number density functions or distributions) has been treated in many ways by different scientific communities
- Crystallization and precipitation (often neglecting spatial inhomogeneities): crystal or particle size, **population balance equation (PBE)**
- Evaporating (and non-evaporating) sprays: droplet surface area, **Williams–Boltzmann equation (WBE)**
- Aerosols and ultra-fine particles: particle mass, **particle dynamics equation (PDE)**
- Particulate systems involved in granular flows: particle velocity, **Boltzmann equation (BE)**
- Although these apparently different theoretical frameworks are referred to by different names, underlying theory (which has its foundation in classical statistical mechanics) is exactly the same!



Working Definition of Polydisperse Particulate & Multiphase Systems

- Problem of describing evolution (in space and time) of these systems in Eulerian framework (in terms of number density functions or distributions) has been treated in many ways by different scientific communities
- Crystallization and precipitation (often neglecting spatial inhomogeneities): crystal or particle size, **population balance equation (PBE)**
- Evaporating (and non-evaporating) sprays: droplet surface area, **Williams–Boltzmann equation (WBE)**
- Aerosols and ultra-fine particles: particle mass, **particle dynamics equation (PDE)**
- Particulate systems involved in granular flows: particle velocity, **Boltzmann equation (BE)**
- Although these apparently different theoretical frameworks are referred to by different names, underlying theory (which has its foundation in classical statistical mechanics) is exactly the same!



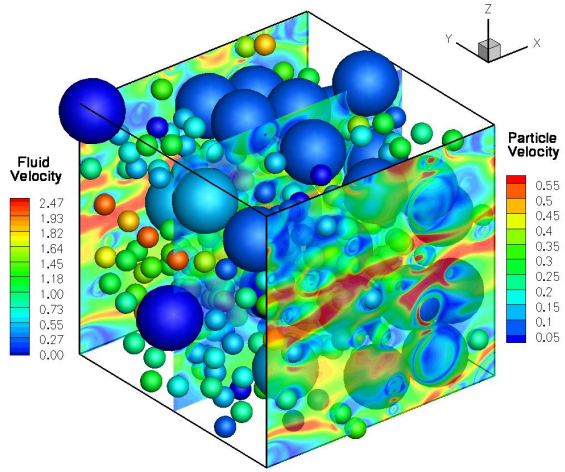
Microscale versus Mesoscale

- **Microscale description** resolves all details:

Fluid velocity: Navier–Stokes eq. no-slip BC on particle surface

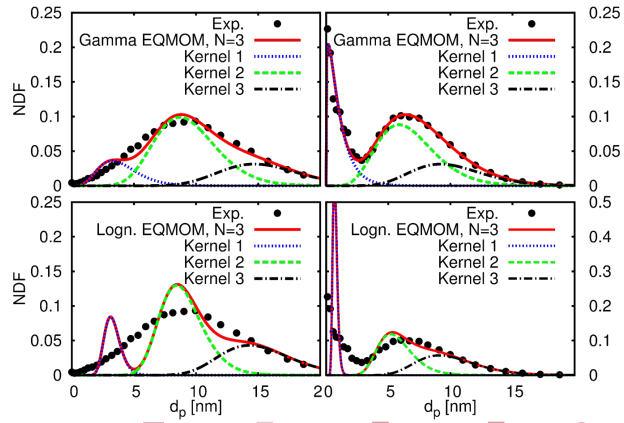
$$M_p^{(n)} \frac{d^2\mathbf{X}^{(n)}}{dt^2} = \text{Fluid force} + \text{collisions}$$

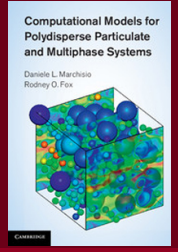
Very fine grid needed to predict forces



- **Mesoscale description** works with distributions:

Fluid velocity: Navier–Stokes eq. with coupling terms
Population balance equation for distribution





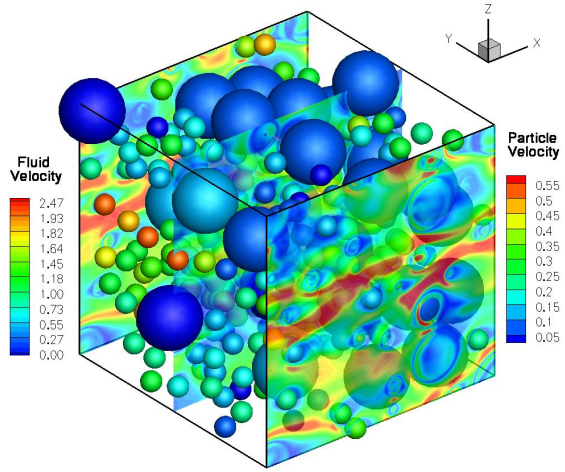
Microscale versus Mesoscale

- **Microscale description** resolves all details:

Fluid velocity: Navier–Stokes eq. no-slip BC on particle surface

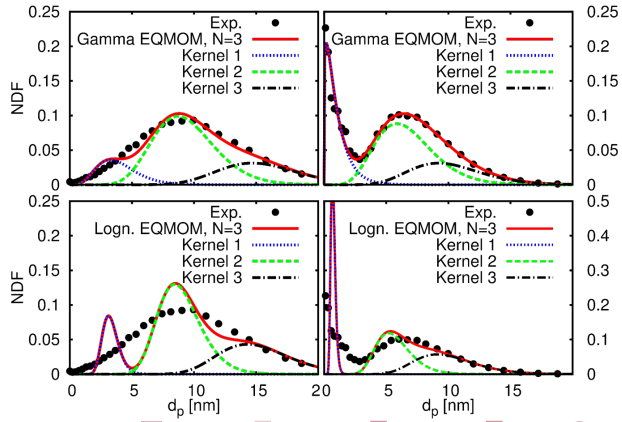
$$M_p^{(n)} \frac{d^2\mathbf{X}^{(n)}}{dt^2} = \text{Fluid force} + \text{collisions}$$

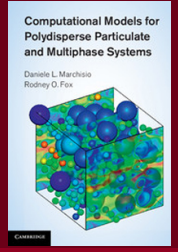
Very fine grid needed to predict forces



- **Mesoscale description** works with distributions:

Fluid velocity: Navier–Stokes eq. with coupling terms
Population balance equation for distribution

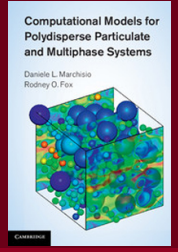




Numerical Methods for Mesoscale

A plethora of methods have been generated:

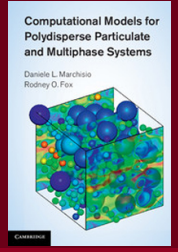
- In PBE, distribution is often discretized into classes (sections), generating **discretized population balance equation (DPBE)**
- Among many methods developed, one used in computational fluid dynamics (CFD) is **multiple-size-group (MUSIG) method**
- Same discretization is carried out for BE in **discrete velocity method (DVM)** and **lattice Boltzmann method (LBM)**
- Method of moments (MOM) is used for solution of both PBE and BE, but resulting closure problem is overcome by following different strategies (e.g. Grad method, **particle number density method**, method of moments with interpolative closure – MOMIC, etc.)



Numerical Methods for Mesoscale

A plethora of methods have been generated:

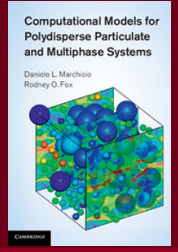
- In PBE, distribution is often discretized into classes (sections), generating **discretized population balance equation (DPBE)**
- Among many methods developed, one used in computational fluid dynamics (CFD) is **multiple-size-group (MUSIG) method**
- Same discretization is carried out for BE in **discrete velocity method (DVM)** and **lattice Boltzmann method (LBM)**
- Method of moments (MOM) is used for solution of both PBE and BE, but resulting closure problem is overcome by following different strategies (e.g. Grad method, **particle number density method**, method of moments with interpolative closure – MOMIC, etc.)



Numerical Methods for Mesoscale

A plethora of methods have been generated:

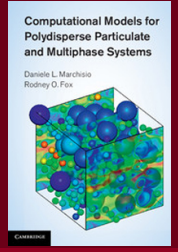
- In PBE, distribution is often discretized into classes (sections), generating **discretized population balance equation (DPBE)**
- Among many methods developed, one used in computational fluid dynamics (CFD) is **multiple-size-group (MUSIG) method**
- Same discretization is carried out for BE in **discrete velocity method (DVM)** and **lattice Boltzmann method (LBM)**
- Method of moments (MOM) is used for solution of both PBE and BE, but resulting closure problem is overcome by following different strategies (e.g. Grad method, **particle number density method**, method of moments with interpolative closure – MOMIC, etc.)



Numerical Methods for Mesoscale

A plethora of methods have been generated:

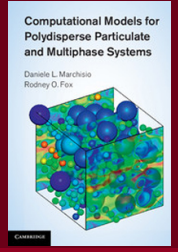
- In PBE, distribution is often discretized into classes (sections), generating **discretized population balance equation (DPBE)**
- Among many methods developed, one used in computational fluid dynamics (CFD) is **multiple-size-group (MUSIG) method**
- Same discretization is carried out for BE in **discrete velocity method (DVM)** and **lattice Boltzmann method (LBM)**
- Method of moments (MOM) is used for solution of both PBE and BE, but resulting closure problem is overcome by following different strategies (e.g. Grad method, **particle number density method**, method of moments with interpolative closure – MOMIC, etc.)



Numerical Methods for Mesoscale

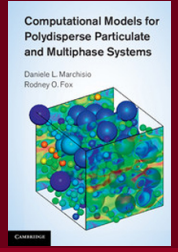
A plethora of methods have been generated:

- In PBE, distribution is often discretized into classes (sections), generating **discretized population balance equation (DPBE)**
- Among many methods developed, one used in computational fluid dynamics (CFD) is **multiple-size-group (MUSIG) method**
- Same discretization is carried out for BE in **discrete velocity method (DVM)** and **lattice Boltzmann method (LBM)**
- Method of moments (MOM) is used for solution of both PBE and BE, but resulting closure problem is overcome by following different strategies (e.g. Grad method, **particle number density method**, method of moments with interpolative closure – MOMIC, etc.)



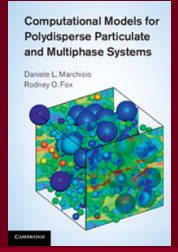
Number Density Functions

- Disperse phase is constituted by **discrete elements**
- Length scales of elements are smaller than length scale of spatial variation of properties of interest
- Each element has properties known as **coordinates** (i.e. two elements are identical if they have identical values for their coordinates)
- Coordinates are classified as **external (spatial coordinates)** and **internal** (properties such as mass, momentum, enthalpy, temperature, volume, surface area, size, or age)
- State of multiphase system is defined through number density function (NDF): $n_\xi(t, \mathbf{x}, \xi)$ where $\mathbf{x} = (x_1, x_2, x_3)$ and $\xi = (\xi_1, \ldots, \xi_M)$



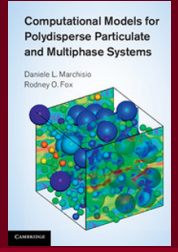
Number Density Functions

- Disperse phase is constituted by **discrete elements**
- Length scales of elements are smaller than length scale of spatial variation of properties of interest
- Each element has properties known as **coordinates** (i.e. two elements are identical if they have identical values for their coordinates)
- Coordinates are classified as **external (spatial coordinates)** and **internal** (properties such as mass, momentum, enthalpy, temperature, volume, surface area, size, or age)
- State of multiphase system is defined through number density function (NDF): $n_\xi(t, \mathbf{x}, \xi)$ where $\mathbf{x} = (x_1, x_2, x_3)$ and $\xi = (\xi_1, \ldots, \xi_M)$



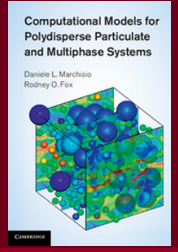
Number Density Functions

- Disperse phase is constituted by **discrete elements**
- Length scales of elements are smaller than length scale of spatial variation of properties of interest
- Each element has properties known as **coordinates** (i.e. two elements are identical if they have identical values for their coordinates)
- Coordinates are classified as **external (spatial coordinates)** and **internal** (properties such as mass, momentum, enthalpy, temperature, volume, surface area, size, or age)
- State of multiphase system is defined through number density function (NDF): $n_\xi(t, \mathbf{x}, \xi)$ where $\mathbf{x} = (x_1, x_2, x_3)$ and $\xi = (\xi_1, \ldots, \xi_M)$



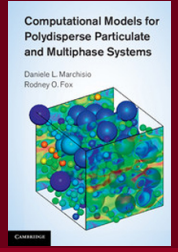
Number Density Functions

- Disperse phase is constituted by **discrete elements**
- Length scales of elements are smaller than length scale of spatial variation of properties of interest
- Each element has properties known as **coordinates** (i.e. two elements are identical if they have identical values for their coordinates)
- Coordinates are classified as **external (spatial coordinates)** and **internal** (properties such as mass, momentum, enthalpy, temperature, volume, surface area, size, or age)
- State of multiphase system is defined through number density function (NDF): $n_\xi(t, \mathbf{x}, \xi)$ where $\mathbf{x} = (x_1, x_2, x_3)$ and $\xi = (\xi_1, \ldots, \xi_M)$



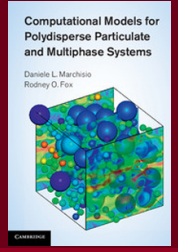
Number Density Functions

- Disperse phase is constituted by **discrete elements**
- Length scales of elements are smaller than length scale of spatial variation of properties of interest
- Each element has properties known as **coordinates** (i.e. two elements are identical if they have identical values for their coordinates)
- Coordinates are classified as **external (spatial coordinates)** and **internal** (properties such as mass, momentum, enthalpy, temperature, volume, surface area, size, or age)
- State of multiphase system is defined through number density function (NDF): $n_\xi(t, \mathbf{x}, \xi)$ where $\mathbf{x} = (x_1, x_2, x_3)$ and $\xi = (\xi_1, \ldots, \xi_M)$



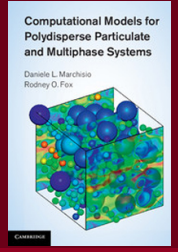
Number Density Functions

- Disperse phase is constituted by **discrete elements**
- Length scales of elements are smaller than length scale of spatial variation of properties of interest
- Each element has properties known as **coordinates** (i.e. two elements are identical if they have identical values for their coordinates)
- Coordinates are classified as **external (spatial coordinates)** and **internal** (properties such as mass, momentum, enthalpy, temperature, volume, surface area, size, or age)
- State of multiphase system is defined through number density function (NDF): $n_{\boldsymbol{\xi}}(t, \mathbf{x}, \boldsymbol{\xi})$ where $\mathbf{x} = (x_1, x_2, x_3)$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)$



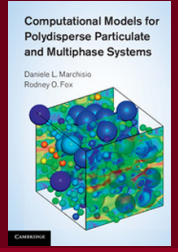
Number Density Functions

- NDF is expected number of entities in infinitesimal physical volume $\mathrm{d}\mathbf{x}$ and in infinitesimal phase-space volume $\mathrm{d}\xi$: $n_\xi \, \mathrm{d}\mathbf{x} \, \mathrm{d}\xi$
- **NDF is not a random quantity**, rather it is ensemble average of infinite realizations (corresponding to same identical macroscopic boundary and initial conditions) of multiphase system
- Being an average quantity, it is (usually) differentiable in space/time
- Underlying definition of NDF is a stochastic description (behavior of any single particle is regarded as random) but behavior of population of particles is treated as deterministic if number of particles in population is large!
- **NDF identifies entire population of particles at any instant and at any given point** in computational domain and considers probability associated with state of each of these particles



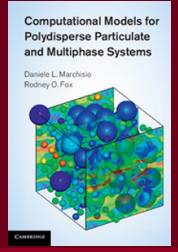
Number Density Functions

- NDF is expected number of entities in infinitesimal physical volume $\mathrm{d}\mathbf{x}$ and in infinitesimal phase-space volume $\mathrm{d}\boldsymbol{\xi}$: $n_{\xi}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\xi}$
- **NDF is not a random quantity**, rather it is ensemble average of infinite realizations (corresponding to same identical macroscopic boundary and initial conditions) of multiphase system
- Being an average quantity, it is (usually) differentiable in space/time
- Underlying definition of NDF is a stochastic description (behavior of any single particle is regarded as random) but behavior of population of particles is treated as deterministic if number of particles in population is large!
- **NDF identifies entire population of particles at any instant and at any given point** in computational domain and considers probability associated with state of each of these particles



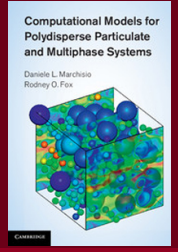
Number Density Functions

- NDF is expected number of entities in infinitesimal physical volume $\mathrm{d}\mathbf{x}$ and in infinitesimal phase-space volume $\mathrm{d}\boldsymbol{\xi}$: $n_{\xi}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\xi}$
- **NDF is not a random quantity**, rather it is ensemble average of infinite realizations (corresponding to same identical macroscopic boundary and initial conditions) of multiphase system
- Being an average quantity, it is (usually) differentiable in space/time
- Underlying definition of NDF is a stochastic description (behavior of any single particle is regarded as random) but behavior of population of particles is treated as deterministic if number of particles in population is large!
- **NDF identifies entire population of particles at any instant and at any given point** in computational domain and considers probability associated with state of each of these particles



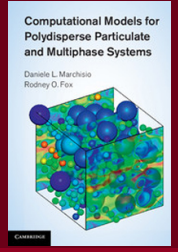
Number Density Functions

- NDF is expected number of entities in infinitesimal physical volume $\mathrm{d}\mathbf{x}$ and in infinitesimal phase-space volume $\mathrm{d}\boldsymbol{\xi}$: $n_{\xi}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\xi}$
- **NDF is not a random quantity**, rather it is ensemble average of infinite realizations (corresponding to same identical macroscopic boundary and initial conditions) of multiphase system
- Being an average quantity, it is (usually) differentiable in space/time
- Underlying definition of NDF is a stochastic description (behavior of any single particle is regarded as random) but behavior of population of particles is treated as deterministic if number of particles in population is large!
- **NDF identifies entire population of particles at any instant and at any given point** in computational domain and considers probability associated with state of each of these particles



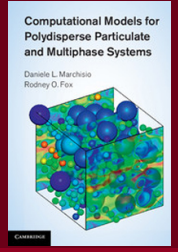
Number Density Functions

- NDF is expected number of entities in infinitesimal physical volume $\mathrm{d}\mathbf{x}$ and in infinitesimal phase-space volume $\mathrm{d}\boldsymbol{\xi}$: $n_{\xi}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\xi}$
- **NDF is not a random quantity**, rather it is ensemble average of infinite realizations (corresponding to same identical macroscopic boundary and initial conditions) of multiphase system
- Being an average quantity, it is (usually) differentiable in space/time
- Underlying definition of NDF is a stochastic description (behavior of any single particle is regarded as random) but behavior of population of particles is treated as deterministic if number of particles in population is large!
- **NDF identifies entire population of particles at any instant and at any given point** in computational domain and considers probability associated with state of each of these particles



Number Density Functions

- NDF is expected number of entities in infinitesimal physical volume $\mathrm{d}\mathbf{x}$ and in infinitesimal phase-space volume $\mathrm{d}\boldsymbol{\xi}$: $n_{\xi}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\xi}$
- **NDF is not a random quantity**, rather it is ensemble average of infinite realizations (corresponding to same identical macroscopic boundary and initial conditions) of multiphase system
- Being an average quantity, it is (usually) differentiable in space/time
- Underlying definition of NDF is a stochastic description (behavior of any single particle is regarded as random) but behavior of population of particles is treated as deterministic if number of particles in population is large!
- **NDF identifies entire population of particles at any instant and at any given point** in computational domain and considers probability associated with state of each of these particles



Number Density Functions

- This random behavior may reflect some chaotic properties (e.g. Brownian motion) or some other behavior (e.g. microscale turbulence → NDF can be treated as any other scalar field)
- Therefore $n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ represents number of disperse entities contained in phase-space volume $\mathrm{d}\boldsymbol{\xi}$ centered at $\boldsymbol{\xi}$ per unit of physical volume (i.e., number density)
- Integration of NDF over all possible values of internal-coordinate vector:

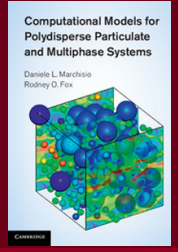
$$N \equiv m_{\xi,0} \equiv \int_{\Omega_\xi} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

corresponding to zero-order moment of NDF

- **Arbitrary integer moment of NDF** is defined by

$$m_{\xi,\mathbf{k}} \equiv \int_{\Omega_\xi} \xi_1^{k_1} \cdots \xi_M^{k_M} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

where $\mathbf{k} \equiv (k_1, \ldots, k_M)$



Number Density Functions

- This random behavior may reflect some chaotic properties (e.g. Brownian motion) or some other behavior (e.g. microscale turbulence → NDF can be treated as any other scalar field)
- Therefore $n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ represents number of disperse entities contained in phase-space volume $\mathrm{d}\boldsymbol{\xi}$ centered at $\boldsymbol{\xi}$ per unit of physical volume (i.e., number density)
- Integration of NDF over all possible values of internal-coordinate vector:

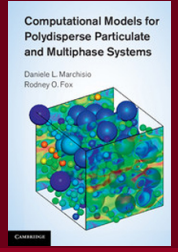
$$N \equiv m_{\xi,0} \equiv \int_{\Omega_\xi} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

corresponding to zero-order moment of NDF

- **Arbitrary integer moment of NDF** is defined by

$$m_{\xi,\mathbf{k}} \equiv \int_{\Omega_\xi} \xi_1^{k_1} \cdots \xi_M^{k_M} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

where $\mathbf{k} \equiv (k_1, \ldots, k_M)$



Number Density Functions

- This random behavior may reflect some chaotic properties (e.g. Brownian motion) or some other behavior (e.g. microscale turbulence → NDF can be treated as any other scalar field)
- Therefore $n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ represents number of disperse entities contained in phase-space volume $\mathrm{d}\boldsymbol{\xi}$ centered at $\boldsymbol{\xi}$ per unit of physical volume (i.e., number density)
- Integration of NDF over all possible values of internal-coordinate vector:

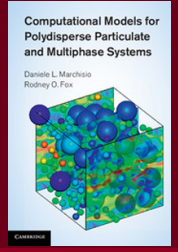
$$N \equiv m_{\xi,0} \equiv \int_{\Omega_\xi} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

corresponding to zero-order moment of NDF

- **Arbitrary integer moment of NDF** is defined by

$$m_{\xi,\mathbf{k}} \equiv \int_{\Omega_\xi} \xi_1^{k_1} \cdots \xi_M^{k_M} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

where $\mathbf{k} \equiv (k_1, \ldots, k_M)$



Number Density Functions

- This random behavior may reflect some chaotic properties (e.g. Brownian motion) or some other behavior (e.g. microscale turbulence $\rightarrow$ NDF can be treated as any other scalar field)
- Therefore $n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ represents number of disperse entities contained in phase-space volume $\mathrm{d}\boldsymbol{\xi}$ centered at $\boldsymbol{\xi}$ per unit of physical volume (i.e., number density)
- Integration of NDF over all possible values of internal-coordinate vector:

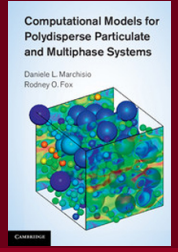
$$N \equiv m_{\xi,0} \equiv \int_{\Omega_\xi} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

corresponding to zero-order moment of NDF

- **Arbitrary integer moment of NDF** is defined by

$$m_{\xi,\mathbf{k}} \equiv \int_{\Omega_\xi} \xi_1^{k_1} \cdots \xi_M^{k_M} n_\xi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$$

where $\mathbf{k} \equiv (k_1, \ldots, k_M)$



Different Types of NDF

- Different definitions of univariate NDF are used in literature
- Choice is problem dependent (e.g. colloids, aerosols, droplets, etc.)
- A change of variables allows to go back and forth:

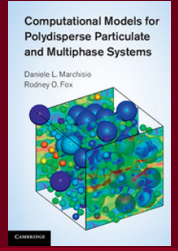
$$n_L(L)\,\mathrm{d}L = n_V(V)\,\mathrm{d}V = n_M(M)\,\mathrm{d}M$$

so it suffices to know relation between length L, volume V and mass M

- For example, if $M = \rho V = \rho L^3$:

$$n_M(M) = n_V(M/\rho)\frac{\mathrm{d}V}{\mathrm{d}M} = \frac{1}{\rho}n_V(M/\rho)$$

$$n_M(M) = n_L\left[(M/\rho)^{1/3}\right]\frac{\mathrm{d}L}{\mathrm{d}M} = \frac{1}{3\rho}\left(\rho/M\right)^{2/3} n_L\left[(M/\rho)^{1/3}\right]$$



Different Types of NDF

- Different definitions of univariate NDF are used in literature
- Choice is problem dependent (e.g. colloids, aerosols, droplets, etc.)
- A change of variables allows to go back and forth:

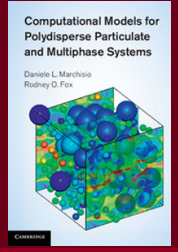
$$n_L(L)\,\mathrm{d}L = n_V(V)\,\mathrm{d}V = n_M(M)\,\mathrm{d}M$$

so it suffices to know relation between length L, volume V and mass M

- For example, if $M = \rho V = \rho L^3$:

$$n_M(M) = n_V(M/\rho)\frac{\mathrm{d}V}{\mathrm{d}M} = \frac{1}{\rho}n_V(M/\rho)$$

$$n_M(M) = n_L\left[(M/\rho)^{1/3}\right]\frac{\mathrm{d}L}{\mathrm{d}M} = \frac{1}{3\rho}\left(\rho/M\right)^{2/3} n_L\left[(M/\rho)^{1/3}\right]$$



Different Types of NDF

- Different definitions of univariate NDF are used in literature
- Choice is problem dependent (e.g. colloids, aerosols, droplets, etc.)
- A change of variables allows to go back and forth:

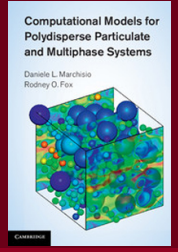
$$n_L(L)\,\mathrm{d}L = n_V(V)\,\mathrm{d}V = n_M(M)\,\mathrm{d}M$$

so it suffices to know relation between length L, volume V and mass M

- For example, if $M = \rho V = \rho L^3$:

$$n_M(M) = n_V(M/\rho)\frac{\mathrm{d}V}{\mathrm{d}M} = \frac{1}{\rho}n_V(M/\rho)$$

$$n_M(M) = n_L\left[(M/\rho)^{1/3}\right]\frac{\mathrm{d}L}{\mathrm{d}M} = \frac{1}{3\rho}(\rho/M)^{2/3}\,n_L\left[(M/\rho)^{1/3}\right]$$



Different Types of NDF

- Different definitions of univariate NDF are used in literature
- Choice is problem dependent (e.g. colloids, aerosols, droplets, etc.)
- A change of variables allows to go back and forth:

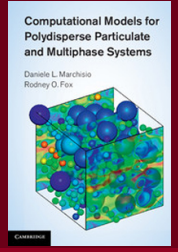
$$n_L(L)\,\mathrm{d}L = n_V(V)\,\mathrm{d}V = n_M(M)\,\mathrm{d}M$$

so it suffices to know relation between length L, volume V and mass M

- For example, if $M = \rho V = \rho L^3$:

$$n_M(M) = n_V(M/\rho)\frac{\mathrm{d}V}{\mathrm{d}M} = \frac{1}{\rho} n_V(M/\rho)$$

$$n_M(M) = n_L\left[(M/\rho)^{1/3}\right]\frac{\mathrm{d}L}{\mathrm{d}M} = \frac{1}{3\rho}\,(\rho/M)^{2/3}\,n_L\left[(M/\rho)^{1/3}\right]$$



Length-Based NDF

- Particles can be characterized by their size or length L
- **Length-based NDF**, $n_{\mathrm{L}}(t, \mathbf{x}, L)$, is defined so that: $n_{\mathrm{L}}\,\mathrm{d}L$ is expected number density of particles with length between L and $L + \mathrm{d}L$
- Total number concentration can be calculated as:

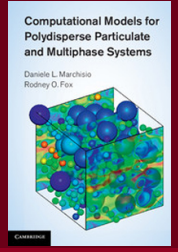
$$N \equiv m_{\mathrm{L},0} = \int_0^\infty n_{\mathrm{L}}(L)\,\mathrm{d}L$$

- Number-averaged particle size can be defined as

$$L_{10} \equiv \frac{1}{N}\int_0^\infty L n_{\mathrm{L}}(L)\,\mathrm{d}L = \frac{m_{\mathrm{L},1}}{N}$$

- We define k^{th} moment of length-based NDF as

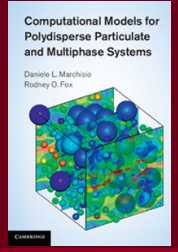
$$m_{\mathrm{L},k} \equiv \int_0^\infty L^k n_{\mathrm{L}}(L)\,\mathrm{d}L$$



Length-Based NDF

- Any mean particle size can be defined as ratio $m_{\mathrm{L},k+1}/m_{\mathrm{L},k}$ for any value of k ($k = 2$, **Sauter mean diameter**, $L_{32} \equiv m_{\mathrm{L},3}/m_{\mathrm{L},2}$)
- If volume scales with third power of length, $V = k_{\mathrm{V}} L^3$ (sphere: $k_{\mathrm{V}} = \pi/6$, cube: $k_{\mathrm{V}} = 1$, etc.), we can define *volume density function* $V_{\mathrm{L}}(L)$ representing volume of particles per unit spatial volume with lengths between L and $L + \mathrm{d}L$: $V_{\mathrm{L}} \equiv k_{\mathrm{V}} L^3 n_{\mathrm{L}}$
- After normalizing V_{L} to unity, *volume-fraction density function* $\alpha_{\mathrm{V}}(L)$ represents volume fraction of particles with a specific length over total particle volume: $\alpha_{\mathrm{V}}(L) := \dfrac{L^3 n_{\mathrm{L}}}{\int_0^\infty L^3 n_{\mathrm{L}}(L)\,\mathrm{d}L}$
- **Mean particle length calculated from volume-fraction density function** is

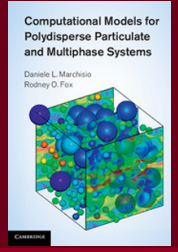
$$L_{43} = \int_0^\infty \alpha_{\mathrm{V}}(L) L \,\mathrm{d}L = \frac{m_{\mathrm{L},4}}{m_{\mathrm{L},3}}$$



Length-Based NDF

- Any mean particle size can be defined as ratio $m_{\mathrm{L},k+1}/m_{\mathrm{L},k}$ for any value of k ($k = 2$, **Sauter mean diameter**, $L_{32} \equiv m_{\mathrm{L},3}/m_{\mathrm{L},2}$)
- If volume scales with third power of length, $V = k_{\mathrm{V}} L^3$ (sphere: $k_{\mathrm{V}} = \pi/6$, cube: $k_{\mathrm{V}} = 1$, etc.), we can define *volume density function* $V_{\mathrm{L}}(L)$ representing volume of particles per unit spatial volume with lengths between L and $L + \mathrm{d}L$: $V_{\mathrm{L}} \equiv k_{\mathrm{V}} L^3 n_{\mathrm{L}}$
- After normalizing V_{L} to unity, *volume-fraction density function* $\alpha_{\mathrm{V}}(L)$ represents volume fraction of particles with a specific length over total particle volume: $\alpha_{\mathrm{V}}(L) := \dfrac{L^3 n_{\mathrm{L}}}{\int_0^\infty L^3 n_{\mathrm{L}}(L)\,\mathrm{d}L}$
- **Mean particle length calculated from volume-fraction density function** is

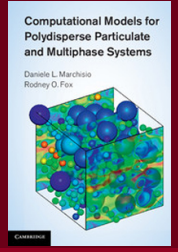
$$L_{43} = \int_0^\infty \alpha_{\mathrm{V}}(L) L \,\mathrm{d}L = \frac{m_{\mathrm{L},4}}{m_{\mathrm{L},3}}$$



Length-Based NDF

- Any mean particle size can be defined as ratio $m_{\mathrm{L},k+1}/m_{\mathrm{L},k}$ for any value of k ($k = 2$, **Sauter mean diameter**, $L_{32} \equiv m_{\mathrm{L},3}/m_{\mathrm{L},2}$)
- If volume scales with third power of length, $V = k_{\mathrm{V}} L^3$ (sphere: $k_{\mathrm{V}} = \pi/6$, cube: $k_{\mathrm{V}} = 1$, etc.), we can define *volume density function* $V_{\mathrm{L}}(L)$ representing volume of particles per unit spatial volume with lengths between L and $L + \mathrm{d}L$: $V_{\mathrm{L}} \equiv k_{\mathrm{V}} L^3 n_{\mathrm{L}}$
- After normalizing V_{L} to unity, *volume-fraction density function* $\alpha_{\mathrm{V}}(L)$ represents volume fraction of particles with a specific length over total particle volume: $\alpha_{\mathrm{V}}(L) := \dfrac{L^3 n_{\mathrm{L}}}{\int_0^\infty L^3 n_{\mathrm{L}}(L)\,\mathrm{d}L}$
- **Mean particle length calculated from volume-fraction density function** is

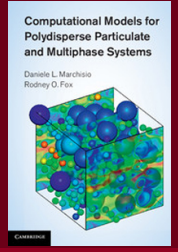
$$L_{43} = \int_0^\infty \alpha_{\mathrm{V}}(L) L \,\mathrm{d}L = \frac{m_{\mathrm{L},4}}{m_{\mathrm{L},3}}$$



Length-Based NDF

- Any mean particle size can be defined as ratio $m_{\mathrm{L},k+1}/m_{\mathrm{L},k}$ for any value of k ($k = 2$, **Sauter mean diameter**, $L_{32} \equiv m_{\mathrm{L},3}/m_{\mathrm{L},2}$)
- If volume scales with third power of length, $V = k_{\mathrm{V}} L^3$ (sphere: $k_{\mathrm{V}} = \pi/6$, cube: $k_{\mathrm{V}} = 1$, etc.), we can define *volume density function* $V_{\mathrm{L}}(L)$ representing volume of particles per unit spatial volume with lengths between L and $L + \mathrm{d}L$: $V_{\mathrm{L}} \equiv k_{\mathrm{V}} L^3 n_{\mathrm{L}}$
- After normalizing V_{L} to unity, *volume-fraction density function* $\alpha_{\mathrm{V}}(L)$ represents volume fraction of particles with a specific length over total particle volume: $\alpha_{\mathrm{V}}(L) := \dfrac{L^3 n_{\mathrm{L}}}{\int_0^\infty L^3 n_{\mathrm{L}}(L)\,\mathrm{d}L}$
- **Mean particle length calculated from volume-fraction density function** is

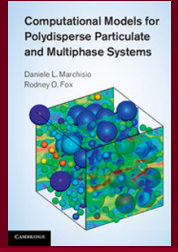
$$L_{43} = \int_0^\infty \alpha_{\mathrm{V}}(L) L \,\mathrm{d}L = \frac{m_{\mathrm{L},4}}{m_{\mathrm{L},3}}$$



Length-Based NDF

- Any mean particle size can be defined as ratio $m_{\mathrm{L},k+1}/m_{\mathrm{L},k}$ for any value of k ($k = 2$, **Sauter mean diameter**, $L_{32} \equiv m_{\mathrm{L},3}/m_{\mathrm{L},2}$)
- If volume scales with third power of length, $V = k_{\mathrm{V}} L^3$ (sphere: $k_{\mathrm{V}} = \pi/6$, cube: $k_{\mathrm{V}} = 1$, etc.), we can define *volume density function* $V_{\mathrm{L}}(L)$ representing volume of particles per unit spatial volume with lengths between L and $L + \mathrm{d}L$: $V_{\mathrm{L}} \equiv k_{\mathrm{V}} L^3 n_{\mathrm{L}}$
- After normalizing V_{L} to unity, *volume-fraction density function* $\alpha_{\mathrm{V}}(L)$ represents volume fraction of particles with a specific length over total particle volume: $\alpha_{\mathrm{V}}(L) := \dfrac{L^3 n_{\mathrm{L}}}{\int_0^\infty L^3 n_{\mathrm{L}}(L)\,\mathrm{d}L}$
- **Mean particle length calculated from volume-fraction density function** is

$$L_{43} = \int_0^\infty \alpha_{\mathrm{V}}(L) L \,\mathrm{d}L = \frac{m_{\mathrm{L},4}}{m_{\mathrm{L},3}}$$



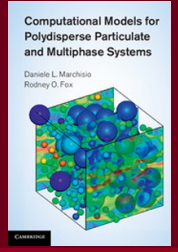
Volume-Based NDF

- Another common internal coordinate is particle volume $V \rightarrow$ **volume-based NDF** as expected number of particles with volume between V and $V + \mathrm{d}V$: $n_{\mathrm{V}}\,\mathrm{d}\mathbf{x}\,\mathrm{d}V$
- If volume and length scale with third power as $V = k_{\mathrm{V}}L^3$, relationship between length-based and volume-based NDFs is straightforward: $n_{\mathrm{L}}(L) = 3k_{\mathrm{V}}L^2 n_{\mathrm{V}}(k_{\mathrm{V}}L^3)$
- Moments of volume-based number NDF are defined as

$$m_{\mathrm{V},k} \equiv \int_0^\infty V^k n_{\mathrm{V}}(V)\,\mathrm{d}V$$

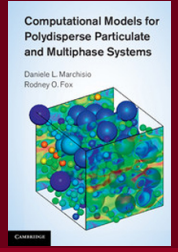
and can be easily related to moments of length-based NDF:

$$m_{\mathrm{V},k} = \int_0^\infty (k_{\mathrm{V}}L^3)^k n_{\mathrm{L}}(L)\,\mathrm{d}L = k_{\mathrm{V}}^k m_{\mathrm{L},3k}$$



Mass-Based NDF

- **Mass-based NDF** $n_{\mathrm{M}}(M)$ is often preferred because mass is conserved quantity for complex shapes
- In case of fractal aggregates: $M = k_{\mathrm{M}} L^{D_{\mathrm{f}}}$ where $1 < D_{\mathrm{f}} \leq 3$ is *fractal dimension* (sphere: $D_{\mathrm{f}} = 3$, diffusion-limited aggregate: $D_{\mathrm{f}} = 1.8$, etc.)
- Length-based and mass-based NDFs are related by $n_{\mathrm{L}}(L) = D_{\mathrm{f}} k_{\mathrm{M}} L^{D_{\mathrm{f}}-1} n_{\mathrm{M}}(k_{\mathrm{M}} L^{D_{\mathrm{f}}})$
- Their moments are related by: $m_{\mathrm{M},k} = k_{\mathrm{M}}^{k} m_{\mathrm{L},D_{\mathrm{f}}k}$
- It is often the case that in closed systems total particle mass is conserved, i.e., $m_{\mathrm{M},1}$ is constant
- In practical applications, choice of NDF with which to work often depends on which moments can be measured experimentally (e.g. light scattering $\Longrightarrow L$, etc.)



Velocity-Based NDF

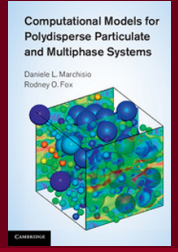
- Special case of considerable interest occurs when internal-coordinate vector is particle velocity vector: $\mathbf{v}$
- **Velocity-based NDF** $n_{\mathrm{U}}(t, \mathbf{x}, \mathbf{v})$ is parameterized by 3 velocity components $\mathbf{v} = (v_1, v_2, v_3)$
- Total number concentration is defined by integrating over all possible values of particle velocity:

$$N = m_{\mathrm{U},0} \equiv \int_{\Omega_{\mathrm{v}}} n_{\mathrm{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

- Because particles may have different velocities, number-weighted particle velocity $\mathbf{U}_{\mathrm{p}}$ can be calculated as:

$$\mathbf{U}_{\mathrm{p}} \equiv \frac{1}{N} \int_{\Omega_{\mathrm{v}}} \mathbf{v} n_{\mathrm{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v} = \frac{m_{\mathrm{U},1}}{m_{\mathrm{U},0}}$$

- Number-weighted particle velocity is only used when all particles have same mass!



Velocity-Based NDF

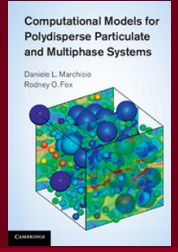
- Special case of considerable interest occurs when internal-coordinate vector is particle velocity vector: $\mathbf{v}$
- **Velocity-based NDF** $n_{\mathbf{U}}(t, \mathbf{x}, \mathbf{v})$ is parameterized by 3 velocity components $\mathbf{v} = (v_1, v_2, v_3)$
- Total number concentration is defined by integrating over all possible values of particle velocity:

$$N = m_{\mathbf{U},0} \equiv \int_{\Omega_\mathbf{v}} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

- Because particles may have different velocities, number-weighted particle velocity $\mathbf{U}_\mathrm{p}$ can be calculated as:

$$\mathbf{U}_\mathrm{p} \equiv \frac{1}{N} \int_{\Omega_\mathbf{v}} \mathbf{v} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v} = \frac{m_{\mathbf{U},1}}{m_{\mathbf{U},0}}$$

- Number-weighted particle velocity is only used when all particles have same mass!



Velocity-Based NDF

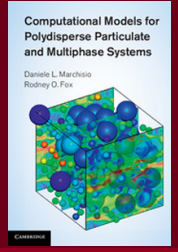
- Special case of considerable interest occurs when internal-coordinate vector is particle velocity vector: $\mathbf{v}$
- **Velocity-based NDF** $n_{\mathbf{U}}(t, \mathbf{x}, \mathbf{v})$ is parameterized by 3 velocity components $\mathbf{v} = (v_1, v_2, v_3)$
- Total number concentration is defined by integrating over all possible values of particle velocity:

$$N = m_{\mathbf{U},0} \equiv \int_{\Omega_{\mathbf{v}}} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

- Because particles may have different velocities, number-weighted particle velocity $\mathbf{U}_{\mathrm{p}}$ can be calculated as:

$$\mathbf{U}_{\mathrm{p}} \equiv \frac{1}{N} \int_{\Omega_{\mathbf{v}}} \mathbf{v} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v} = \frac{m_{\mathbf{U},1}}{m_{\mathbf{U},0}}$$

- Number-weighted particle velocity is only used when all particles have same mass!



Velocity-Based NDF

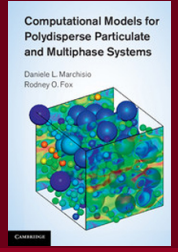
- Special case of considerable interest occurs when internal-coordinate vector is particle velocity vector: $\mathbf{v}$
- **Velocity-based NDF** $n_{\mathbf{U}}(t, \mathbf{x}, \mathbf{v})$ is parameterized by 3 velocity components $\mathbf{v} = (v_1, v_2, v_3)$
- Total number concentration is defined by integrating over all possible values of particle velocity:

$$N = m_{\mathbf{U},0} \equiv \int_{\Omega_{\mathbf{v}}} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

- Because particles may have different velocities, number-weighted particle velocity $\mathbf{U}_{\mathrm{p}}$ can be calculated as:

$$\mathbf{U}_{\mathrm{p}} \equiv \frac{1}{N} \int_{\Omega_{\mathbf{v}}} \mathbf{v} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v} = \frac{m_{\mathbf{U},1}}{m_{\mathbf{U},0}}$$

- Number-weighted particle velocity is only used when all particles have same mass!



Velocity-Based NDF

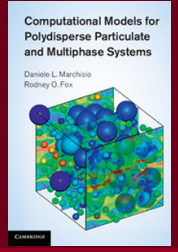
- Special case of considerable interest occurs when internal-coordinate vector is particle velocity vector: $\mathbf{v}$
- **Velocity-based NDF** $n_{\mathbf{U}}(t, \mathbf{x}, \mathbf{v})$ is parameterized by 3 velocity components $\mathbf{v} = (v_1, v_2, v_3)$
- Total number concentration is defined by integrating over all possible values of particle velocity:

$$N = m_{\mathbf{U},0} \equiv \int_{\Omega_{\mathbf{v}}} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

- Because particles may have different velocities, number-weighted particle velocity $\mathbf{U}_{\mathrm{p}}$ can be calculated as:

$$\mathbf{U}_{\mathrm{p}} \equiv \frac{1}{N} \int_{\Omega_{\mathbf{v}}} \mathbf{v} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v} = \frac{m_{\mathbf{U},1}}{m_{\mathbf{U},0}}$$

- Number-weighted particle velocity is only used when all particles have same mass!



Velocity-Based NDF

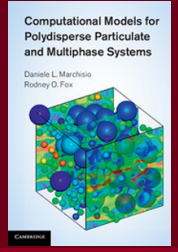
- Special case of considerable interest occurs when internal-coordinate vector is particle velocity vector: $\mathbf{v}$
- **Velocity-based NDF** $n_{\mathbf{U}}(t, \mathbf{x}, \mathbf{v})$ is parameterized by 3 velocity components $\mathbf{v} = (v_1, v_2, v_3)$
- Total number concentration is defined by integrating over all possible values of particle velocity:

$$N = m_{\mathbf{U},0} \equiv \int_{\Omega_{\mathbf{v}}} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

- Because particles may have different velocities, number-weighted particle velocity $\mathbf{U}_{\mathrm{p}}$ can be calculated as:

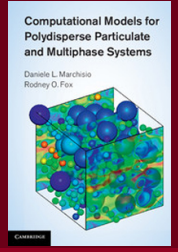
$$\mathbf{U}_{\mathrm{p}} \equiv \frac{1}{N} \int_{\Omega_{\mathbf{v}}} \mathbf{v} n_{\mathbf{U}}(\mathbf{v})\, \mathrm{d}\mathbf{v} = \frac{m_{\mathbf{U},1}}{m_{\mathbf{U},0}}$$

- Number-weighted particle velocity is only used when all particles have same mass!



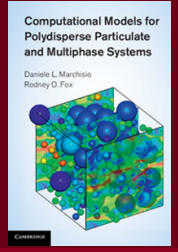
Velocity-Based NDF

- NDF n_ξ and velocity-based NDF $n_{\mathbf{U}}$ have same mathematical meaning but particle velocity is a particular internal coordinate
- We distinguish between ***passive* internal coordinates** such as particle size, volume, area, or temperature, from ***active* internal coordinates** such as particle velocity
- In first case we refer to **population balance equation (PBE)** and in second to **generalized population balance equation (GPBE)**
- In PBE, velocity of disperse phase is assumed to be known and NDF can be treated as advected scalar field
- Scalar fields appearing in GPBE are usually active (i.e., momentum exchange between particles depends on internal coordinates such as length) and thus velocity of disperse phase must be computed from seperate momentum balance



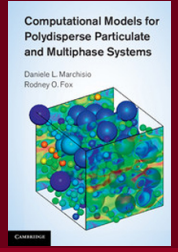
Velocity-Based NDF

- NDF n_ξ and velocity-based NDF $n_{\mathbf{U}}$ have same mathematical meaning but particle velocity is a particular internal coordinate
- We distinguish between ***passive* internal coordinates** such as particle size, volume, area, or temperature, from ***active* internal coordinates** such as particle velocity
- In first case we refer to **population balance equation (PBE)** and in second to **generalized population balance equation (GPBE)**
- In PBE, velocity of disperse phase is assumed to be known and NDF can be treated as advected scalar field
- Scalar fields appearing in GPBE are usually active (i.e., momentum exchange between particles depends on internal coordinates such as length) and thus velocity of disperse phase must be computed from seperate momentum balance



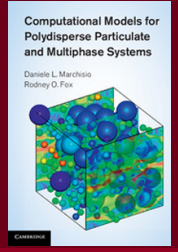
Velocity-Based NDF

- NDF n_ξ and velocity-based NDF $n_{\mathbf{U}}$ have same mathematical meaning but particle velocity is a particular internal coordinate
- We distinguish between ***passive* internal coordinates** such as particle size, volume, area, or temperature, from ***active* internal coordinates** such as particle velocity
- In first case we refer to **population balance equation (PBE)** and in second to **generalized population balance equation (GPBE)**
- In PBE, velocity of disperse phase is assumed to be known and NDF can be treated as advected scalar field
- Scalar fields appearing in GPBE are usually active (i.e., momentum exchange between particles depends on internal coordinates such as length) and thus velocity of disperse phase must be computed from seperate momentum balance



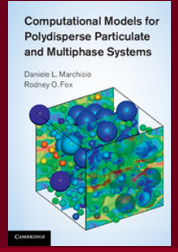
Velocity-Based NDF

- NDF n_ξ and velocity-based NDF $n_{\mathbf{U}}$ have same mathematical meaning but particle velocity is a particular internal coordinate
- We distinguish between ***passive* internal coordinates** such as particle size, volume, area, or temperature, from ***active* internal coordinates** such as particle velocity
- In first case we refer to **population balance equation (PBE)** and in second to **generalized population balance equation (GPBE)**
- In PBE, velocity of disperse phase is assumed to be known and NDF can be treated as advected scalar field
- Scalar fields appearing in GPBE are usually active (i.e., momentum exchange between particles depends on internal coordinates such as length) and thus velocity of disperse phase must be computed from seperate momentum balance



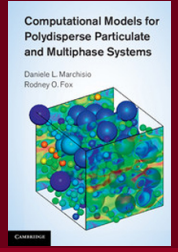
Velocity-Based NDF

- NDF n_ξ and velocity-based NDF $n_{\mathbf{U}}$ have same mathematical meaning but particle velocity is a particular internal coordinate
- We distinguish between ***passive* internal coordinates** such as particle size, volume, area, or temperature, from ***active* internal coordinates** such as particle velocity
- In first case we refer to **population balance equation (PBE)** and in second to **generalized population balance equation (GPBE)**
- In PBE, velocity of disperse phase is assumed to be known and NDF can be treated as advected scalar field
- Scalar fields appearing in GPBE are usually active (i.e., momentum exchange between particles depends on internal coordinates such as length) and thus velocity of disperse phase must be computed from seperate momentum balance



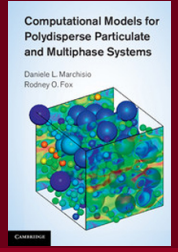
Velocity-Based NDF

- NDF n_ξ and velocity-based NDF $n_{\mathbf{U}}$ have same mathematical meaning but particle velocity is a particular internal coordinate
- We distinguish between ***passive* internal coordinates** such as particle size, volume, area, or temperature, from ***active* internal coordinates** such as particle velocity
- In first case we refer to **population balance equation (PBE)** and in second to **generalized population balance equation (GPBE)**
- In PBE, velocity of disperse phase is assumed to be known and NDF can be treated as advected scalar field
- Scalar fields appearing in GPBE are usually active (i.e., momentum exchange between particles depends on internal coordinates such as length) and thus velocity of disperse phase must be computed from seperate momentum balance



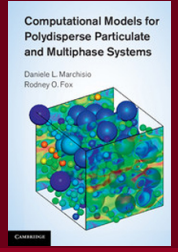
Summary

- Disperse multiphase systems can be modeled at different levels
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- NDF can be univariate or multivariate, and definition is problem dependent
- Thus, first step is to formulate a mesoscale model for polydisperse flows



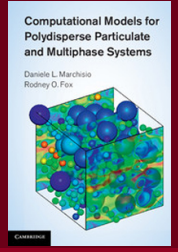
Summary

- Disperse multiphase systems can be modeled at different levels
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- NDF can be univariate or multivariate, and definition is problem dependent
- Thus, first step is to formulate a mesoscale model for polydisperse flows



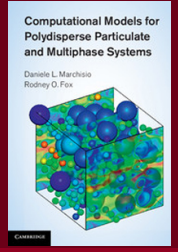
Summary

- Disperse multiphase systems can be modeled at different levels
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- NDF can be univariate or multivariate, and definition is problem dependent
- Thus, first step is to formulate a mesoscale model for polydisperse flows



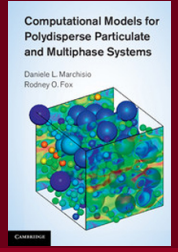
Summary

- Disperse multiphase systems can be modeled at different levels
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- NDF can be univariate or multivariate, and definition is problem dependent
- Thus, first step is to formulate a mesoscale model for polydisperse flows



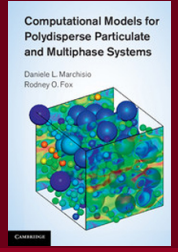
Summary

- Disperse multiphase systems can be modeled at different levels
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- NDF can be univariate or multivariate, and definition is problem dependent
- Thus, first step is to formulate a mesoscale model for polydisperse flows

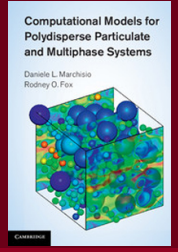


Summary

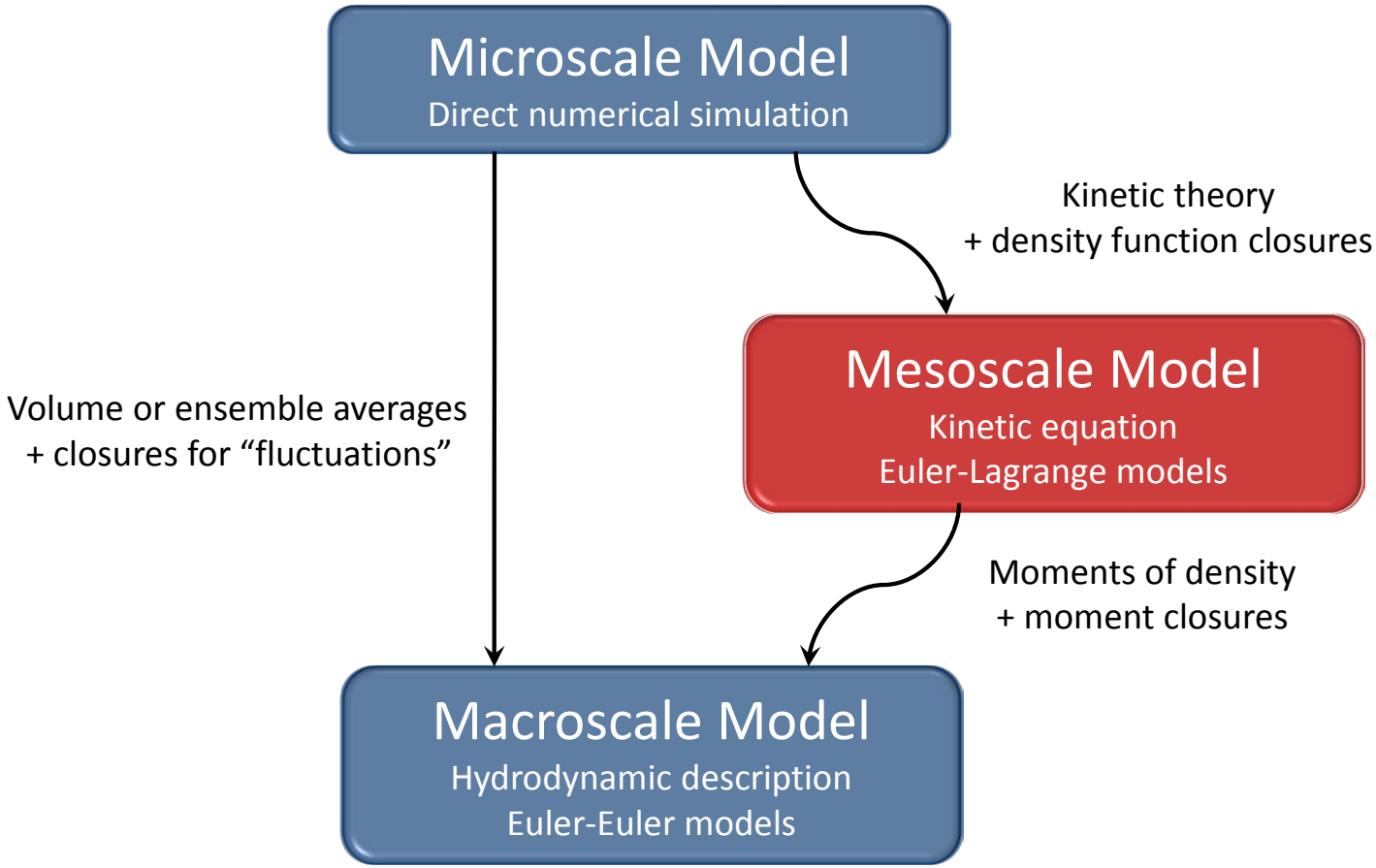
- Disperse multiphase systems can be modeled at different levels
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- NDF can be univariate or multivariate, and definition is problem dependent
- Thus, first step is to formulate a mesoscale model for polydisperse flows



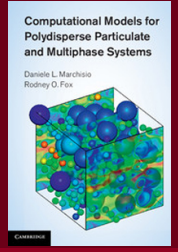
MESOSCALE DESCRIPTION OF POLYDISPERSE FLOWS



Microscale versus Mesoscale versus Macroscale



Mesoscale model incorporates more microscale physics in closures!

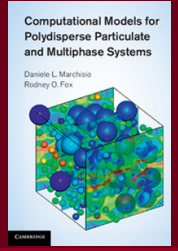


Population Balance Equation

- PBE is simple continuity statement written in terms of NDF
- It can be derived as balance for particles in fixed finite control volume in physical space $\Omega_{\mathbf{x}}$ and in phase space $\Omega_{\boldsymbol{\xi}}$ with boundaries defined as $\partial\Omega_{\mathbf{x}}$ and $\partial\Omega_{\boldsymbol{\xi}}$:

$$\frac{\partial}{\partial t}\left(\int_{\Omega_{\mathbf{x}}} \mathrm{d}\mathbf{x} \int_{\Omega_{\boldsymbol{\xi}}} \mathrm{d}\boldsymbol{\xi}\, n_{\boldsymbol{\xi}}\right) + \int_{\Omega_{\boldsymbol{\xi}}} \mathrm{d}\boldsymbol{\xi} \int_{\partial\Omega_{\mathbf{x}}} n_{\boldsymbol{\xi}}\mathbf{v}\cdot\mathrm{d}\mathbf{A}_{\mathbf{x}} + \int_{\Omega_{\mathbf{x}}} \mathrm{d}\mathbf{x} \int_{\partial\Omega_{\boldsymbol{\xi}}} n_{\boldsymbol{\xi}}\dot{\boldsymbol{\xi}}\cdot\mathrm{d}\mathbf{A}_{\boldsymbol{\xi}} = \int_{\Omega_{\mathbf{x}}} \mathrm{d}\mathbf{x} \int_{\Omega_{\boldsymbol{\xi}}} \mathrm{d}\boldsymbol{\xi}\, h_{\boldsymbol{\xi}}$$

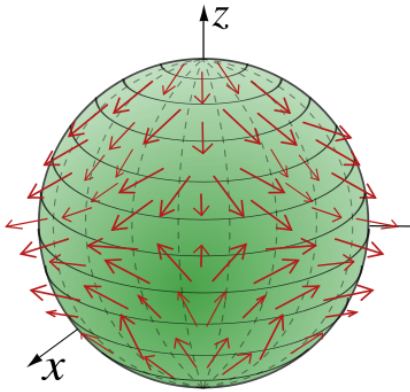
- where $\mathbf{v}$ is (known) velocity vector for particulate system, $\dot{\boldsymbol{\xi}}$ is continuous rate of change in phase space, and $h_{\boldsymbol{\xi}}$ is discontinuous jump function representing discrete events



Population Balance Equation

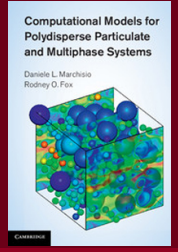
Reynolds-Gauss theorem

Given control volume $\Omega_{\mathbf{x}}$ and its boundary $\partial\Omega_{\mathbf{x}}$ oriented by outward-pointing normals, flux of vector $\mathbf{v}$ across boundary is equal to volume integral of divergence of $\mathbf{v}$ (i.e., $\nabla_{\mathbf{x}} \cdot \mathbf{v}$) inside control volume



$$\int_{\partial\Omega_{\mathbf{x}}} n_\xi \mathbf{v} \cdot \mathrm{d}\mathbf{A}_{\mathbf{x}} = \int_{\Omega_{\mathbf{x}}} (\nabla_{\mathbf{x}} \cdot \mathbf{v} n_\xi)\, \mathrm{d}\mathbf{x}$$

$$\int_{\partial\Omega_\xi} n_\xi \dot{\boldsymbol{\xi}} \cdot \mathrm{d}\mathbf{A}_\xi = \int_{\Omega_\xi} \left(\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n_\xi \right) \mathrm{d}\boldsymbol{\xi}$$

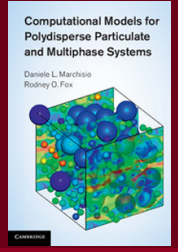


Population Balance Equation

- If Reynolds-Gauss theorem is applied, it is straightforward to obtain

$$\frac{\partial}{\partial t}\left(\int_{\Omega_{\mathbf{x}}} \mathrm{d}\mathbf{x} \int_{\Omega_{\boldsymbol{\xi}}} \mathrm{d}\boldsymbol{\xi}\, n_{\boldsymbol{\xi}}\right) + \int_{\Omega_{\mathbf{x}}} \mathrm{d}\mathbf{x} \int_{\Omega_{\boldsymbol{\xi}}} \mathrm{d}\boldsymbol{\xi} \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}\, n_{\boldsymbol{\xi}} + \int_{\Omega_{\mathbf{x}}} \mathrm{d}\mathbf{x} \int_{\Omega_{\boldsymbol{\xi}}} \mathrm{d}\boldsymbol{\xi} \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}\, n_{\boldsymbol{\xi}} = \int_{\Omega_{\mathbf{x}}} \mathrm{d}\mathbf{x} \int_{\Omega_{\boldsymbol{\xi}}} \mathrm{d}\boldsymbol{\xi}\, h_{\boldsymbol{\xi}}$$

- $\frac{\partial}{\partial \mathbf{x}} = \boldsymbol{\nabla}_{\mathbf{x}} = (\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3)$ is gradient operator in physical space
- $\frac{\partial}{\partial \boldsymbol{\xi}} = \boldsymbol{\nabla}_{\boldsymbol{\xi}} = (\partial/\partial \xi_1, \ldots, \partial/\partial \xi_M)$ is gradient operator in phase space



Population Balance Equation

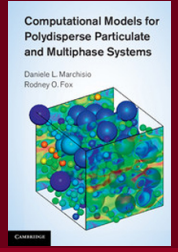
- For any arbitrary control volumes $\Omega_{\mathbf{x}}$ and Ω_{ξ}, integrand must satisfy relation

$$\frac{\partial n_{\xi}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}\, n_{\xi} + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}\, n_{\xi} = h_{\xi}$$

- Einstein notation (i.e., repeated Roman indices imply summation)

$$\frac{\partial n_{\xi}}{\partial t} + \frac{\partial}{\partial x_i} v_i\, n_{\xi} + \frac{\partial}{\partial \xi_i} \dot{\xi}_i\, n_{\xi} = h_{\xi}$$

- Very important is distinction between **continuous processes (drift terms)** and **discontinuous processes (instantaneous jumps)**

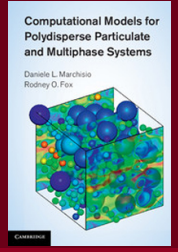


Generalized Population Balance Equation

- By using very similar approach it is possible to derive GPBE for NDF that includes particle velocity: $n(t, \mathbf{x}, \mathbf{v}, \boldsymbol{\xi})$

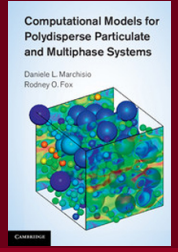
$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}\, n + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}_{\mathrm{p}} n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}\, n = h$$

- $\mathbf{A}_{\mathrm{p}}$ is continuous rate of change of particle velocity (i.e., acceleration) or force per unit mass acting on particles (e.g., gravity, fluid drag, etc.)
- Right-hand side h is discontinuous jump term, but now including discontinuous changes in particle momentum (e.g., collisions)



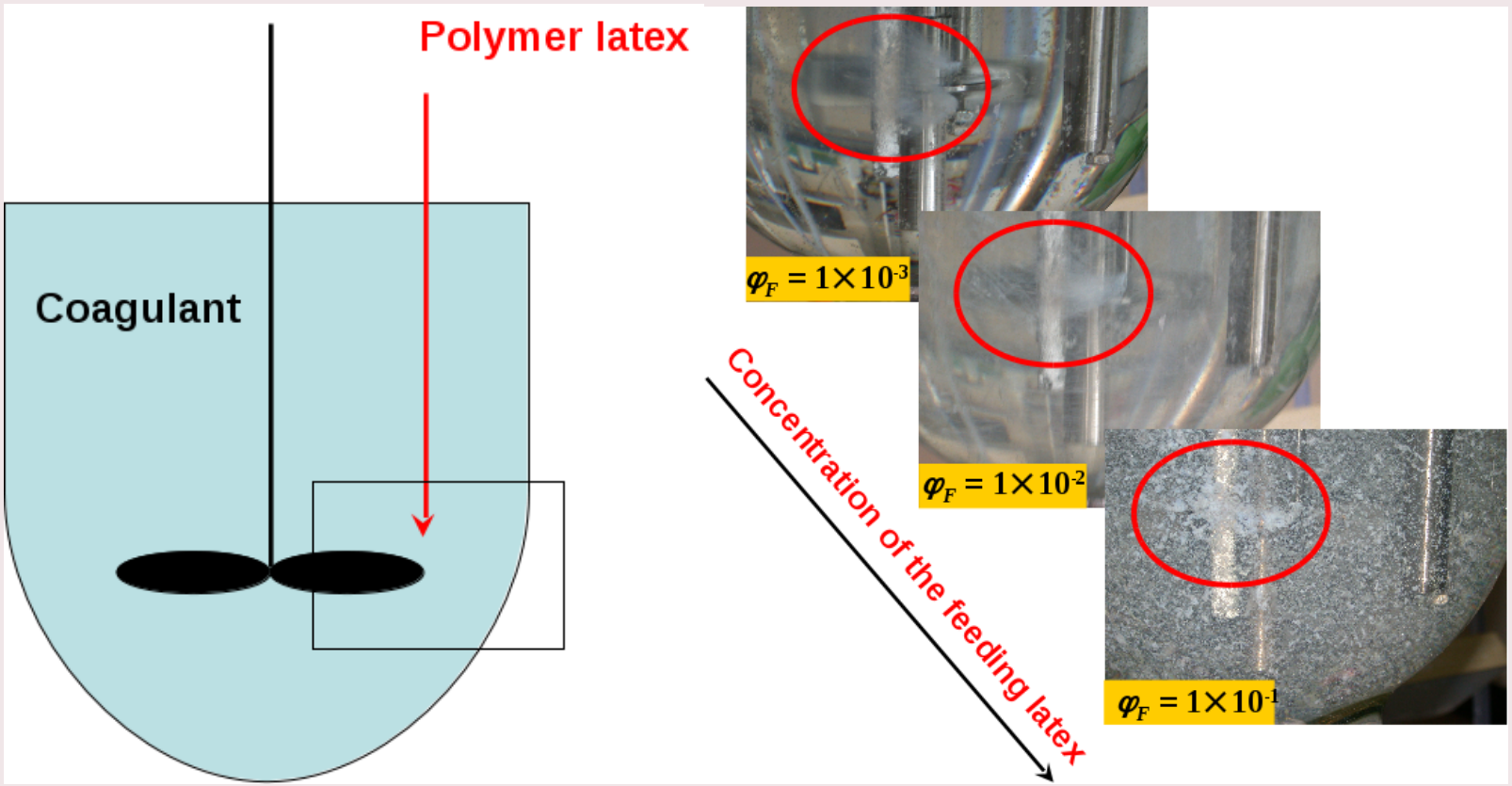
Generalized Population Balance Equation

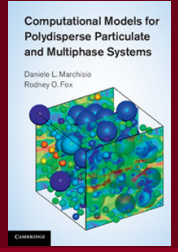
- Collision, aggregation, agglomeration, coalescence, breakup events can be described as **discontinuous** only when they occur on time and length scales that are much shorter than those characterizing changes in NDF
- This is indeed the case of particle–particle interactions via the hard-sphere potential (pure collisions) but with other potentials (i.e. soft-sphere potential) this may not be a good approximation
- In GPBE there are never terms for diffusion in physical space, however, there are often terms for diffusion in velocity phase space due to random (e.g., Brownian) forces, or to turbulent fluctuations in continuous phase due to drag



A Simple Example: Particle Aggregation and Breakage in CSTR

Stirred tank / Polystyrene latex primary particles - 66 nm / Coagulant - $Al(NO_3)_3$ / Re (capillary) = 500 / D (capillary) = 3 mm

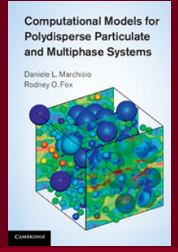




A Simple Example: Particle Aggregation and Breakage in CSTR

When is CFD needed vs. well-stirred model?

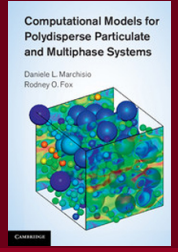
- Latex particles aggregation due to coagulant
- Aggregation rate depends on shear rate G seen by particles
- Aggregates breakup in high-shear regions
- Faster stirring results in faster mixing, but higher shear
- Latex concentration in feed affects aggregation rate
- Laser light scattering determines radius of gyration $\langle R_g \rangle \propto \langle L \rangle$
- Three characeristic time scales: τ_M mixing, τ_A aggregation, τ_B breakage



A Simple Example: Particle Aggregation and Breakage in CSTR

When is CFD needed vs. well-stirred model?

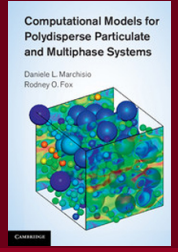
- Latex particles aggregation due to coagulant
- Aggregation rate depends on shear rate G seen by particles
- Aggregates breakup in high-shear regions
- Faster stirring results in faster mixing, but higher shear
- Latex concentration in feed affects aggregation rate
- Laser light scattering determines radius of gyration $\langle R_g \rangle \propto \langle L \rangle$
- Three characeristic time scales: τ_M mixing, τ_A aggregation, τ_B breakage



A Simple Example: Particle Aggregation and Breakage in CSTR

When is CFD needed vs. well-stirred model?

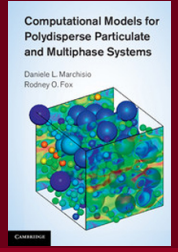
- Latex particles aggregation due to coagulant
- Aggregation rate depends on shear rate G seen by particles
- Aggregates breakup in high-shear regions
- Faster stirring results in faster mixing, but higher shear
- Latex concentration in feed affects aggregation rate
- Laser light scattering determines radius of gyration $\langle R_g \rangle \propto \langle L \rangle$
- Three characeristic time scales: τ_M mixing, τ_A aggregation, τ_B breakage



A Simple Example: Particle Aggregation and Breakage in CSTR

When is CFD needed vs. well-stirred model?

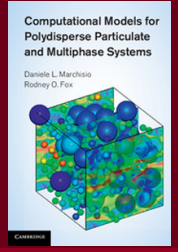
- Latex particles aggregation due to coagulant
- Aggregation rate depends on shear rate G seen by particles
- Aggregates breakup in high-shear regions
- Faster stirring results in faster mixing, but higher shear
- Latex concentration in feed affects aggregation rate
- Laser light scattering determines radius of gyration $\langle R_g \rangle \propto \langle L \rangle$
- Three characeristic time scales: τ_M mixing, τ_A aggregation, τ_B breakage



A Simple Example: Particle Aggregation and Breakage in CSTR

When is CFD needed vs. well-stirred model?

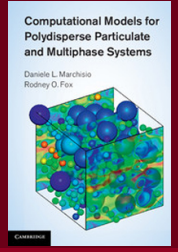
- Latex particles aggregation due to coagulant
- Aggregation rate depends on shear rate G seen by particles
- Aggregates breakup in high-shear regions
- Faster stirring results in faster mixing, but higher shear
- Latex concentration in feed affects aggregation rate
- Laser light scattering determines radius of gyration $\langle R_g \rangle \propto \langle L \rangle$
- Three characeristic time scales: τ_M mixing, τ_A aggregation, τ_B breakage



A Simple Example: Particle Aggregation and Breakage in CSTR

When is CFD needed vs. well-stirred model?

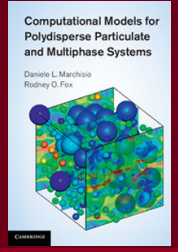
- Latex particles aggregation due to coagulant
- Aggregation rate depends on shear rate G seen by particles
- Aggregates breakup in high-shear regions
- Faster stirring results in faster mixing, but higher shear
- Latex concentration in feed affects aggregation rate
- Laser light scattering determines radius of gyration $\langle R_g \rangle \propto \langle L \rangle$
- Three characeristic time scales: τ_M mixing, τ_A aggregation, τ_B breakage



A Simple Example: Particle Aggregation and Breakage in CSTR

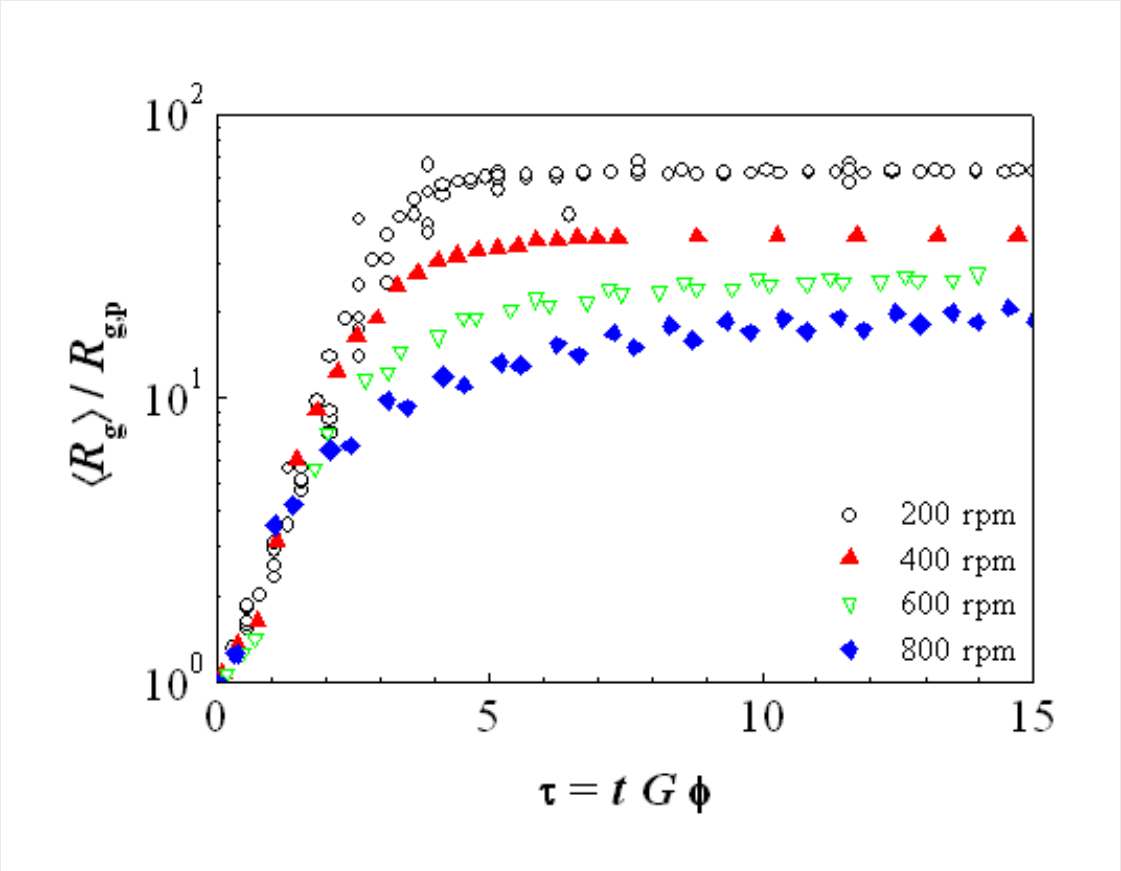
When is CFD needed vs. well-stirred model?

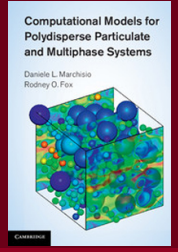
- Latex particles aggregation due to coagulant
- Aggregation rate depends on shear rate G seen by particles
- Aggregates breakup in high-shear regions
- Faster stirring results in faster mixing, but higher shear
- Latex concentration in feed affects aggregation rate
- Laser light scattering determines radius of gyration $\langle R_g \rangle \propto \langle L \rangle$
- Three characeristic time scales: τ_M mixing, τ_A aggregation, τ_B breakage



A Simple Example: Particle Aggregation and Breakage in CSTR

Eventually aggregation and breakage balance each other and steady state is reached (dynamic equilibrium)

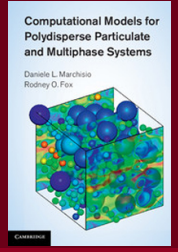




A Simple Example: Particle Aggregation and Breakage in CSTR

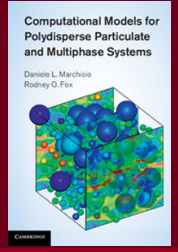
- Here only one internal coordinate is considered, ξ, dimensionless mass of aggregate (i.e., actual mass divided by mass of primary particles)
- $n(t, \mathbf{x}, \xi)$ is Reynolds-averaged NDF often called cluster mass distribution (CMD)
- Reynolds-averaged PBE (neglecting fluctuations) is:

$$\frac{\partial n(t,\mathbf{x},\xi)}{\partial t} + \frac{\partial}{\partial x_i}\langle u_i\rangle n(t,\mathbf{x},\xi) - \frac{\partial}{\partial x_i} D_t \frac{\partial n(t,\mathbf{x},\xi)}{\partial x_i} =$$
$$+ \frac{1}{2}\int_0^{\xi} k^A(G;\xi-\xi',\xi') n(t,\mathbf{x},\xi-\xi') n(t,\mathbf{x},\xi')\,\mathrm{d}\xi'$$
$$- n(t,\mathbf{x},\xi)\int_0^{\infty} k^A(G;\xi,\xi') n(t,\mathbf{x},\xi')\,\mathrm{d}\xi'$$
$$+ \int_{\xi}^{\infty} k^B(G;\xi') b(\xi|\xi') n(t,\mathbf{x},\xi')\,\mathrm{d}\xi' - k^B(G;\xi) n(t,\mathbf{x},\xi)$$



A Simple Example: Particle Aggregation and Breakage in CSTR

- $\langle u_i \rangle$ is Reynolds-averaged fluid velocity in i^{th} spatial direction and D_t is turbulent diffusivity
- $k^A(G;\xi,\xi')$ is aggregation kernel of two aggregates with masses ξ and ξ'
- $k^B(G;\xi')$ is breakage kernel of aggregate of mass ξ'
- $b(\xi|\xi')$ is daughter distribution function
- G is (turbulent) shear rate, a spatial dependent property [i.e., $G = G(\mathbf{x})$], that is driving force for turbulent aggregation and breakage defined as $G = \left(\frac{\varepsilon}{\nu}\right)^{1/2}$
- It is usually quite easy to estimate volume-averaged shear rate in a vessel: $\overline{G} = \left(\frac{P}{\rho V \nu}\right)^{1/2}$



A Simple Example: Particle Aggregation and Breakage in CSTR

When steady state is reached PBE can be easily made dimensionless by using volume-averaged total number density of aggregates, N_t, characteristic length scale (e.g. tank diameter), characteristic mixing time τ_M and average mass of aggregates at steady state, $\langle \xi \rangle$:

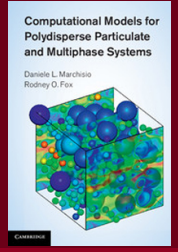
$$\frac{\partial}{\partial X_i} U_i N(\theta, \mathbf{X}, \xi) - \frac{\partial}{\partial X_i} \Gamma_t \frac{\partial N(\theta, \mathbf{X}, \xi)}{\partial X_i} =$$

$$+ \frac{\tau_M}{2\tau_A} \int_0^{\xi} K^A(G; \xi - \xi', \xi') N(\theta, \mathbf{X}, \xi - \xi') N(\theta, \mathbf{X}, \xi') \, \mathrm{d}\xi'$$

$$- \frac{\tau_M}{\tau_A} N(\theta, \mathbf{X}, \xi) \int_0^{\infty} K^A(G; \xi, \xi') N(\theta, \mathbf{X}, \xi') \, \mathrm{d}\xi'$$

$$+ \frac{\tau_M}{\tau_B} \int_{\xi}^{\infty} K^B(G; \xi') b(\xi | \xi') N(\theta, \mathbf{X}, \xi') \, \mathrm{d}\xi'$$

$$- \frac{\tau_M}{\tau_B} K^B(G; \xi) N(\theta, \mathbf{X}, \xi)$$



A Simple Example: Particle Aggregation and Breakage in CSTR

$$N := \frac{n(\xi;\mathbf{x},t)}{N_t}$$

$$X_i := \frac{x_i}{d}$$

$$U_i := \frac{\langle u_i \rangle}{d/\tau_M}$$

$$\Gamma_t := \frac{D_t}{d^2/\tau_M}$$

$$K^A := \frac{k^A(G;\xi,\xi')}{k^A(\overline{G};\langle\xi\rangle,\langle\xi\rangle)}$$

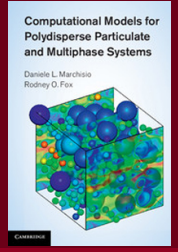
$$K^B := \frac{k^B(G;\xi)}{k^B(\overline{G};\langle\xi\rangle)}$$

Characteristic time for aggregation

$$\tau_A := \frac{1}{k^A(\overline{G};\langle\xi\rangle,\langle\xi\rangle)N_t}$$

Characteristic time for breakage

$$\tau_B := \frac{1}{k^B(\overline{G};\langle\xi\rangle)}$$



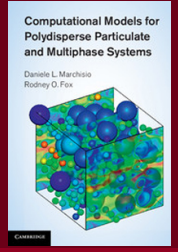
A Simple Example: Particle Aggregation and Breakage in CSTR

- When mixing is much faster than aggregation $\tau_M \ll \tau_A$ and breakage $\tau_M \ll \tau_B$ spatial gradients of CMD are very small and problem can be solved in terms of volume-averaged CMD:

$$\overline{n}(t,\xi) = \frac{\int_V n(t,\mathbf{x},\xi)\,\mathrm{d}\mathbf{x}}{V}$$

- Its evolution is dictated by volume-averaged PBE:

$$\frac{\partial \overline{n}(t,\xi)}{\partial t} = \frac{1}{2}\int_0^{\xi} \overline{k}^A(G;\xi-\xi',\xi')\overline{n}(t,\xi-\xi')\overline{n}(t,\xi')\,\mathrm{d}\xi' - \overline{n}(t,\xi)\int_0^{\infty} \overline{k}^A(G;\xi,\xi')\overline{n}(t,\xi')\,\mathrm{d}\xi' + \int_{\xi}^{\infty} \overline{k}^B(G;\xi')b(\xi|\xi')\overline{n}(t,\xi')\,\mathrm{d}\xi' - \overline{k}^B(G;\xi)\overline{n}(t,\xi)$$



A Simple Example: Particle Aggregation and Breakage in CSTR

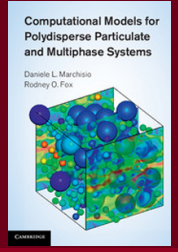
- Volume-averaged aggregation kernel is

$$\overline{k}^A(G;\xi,\xi') = \int_0^\infty k^A(G^*;\xi,\xi') f(G^*)\,\mathrm{d}G^* \neq k^A(\overline{G};\xi,\xi')$$

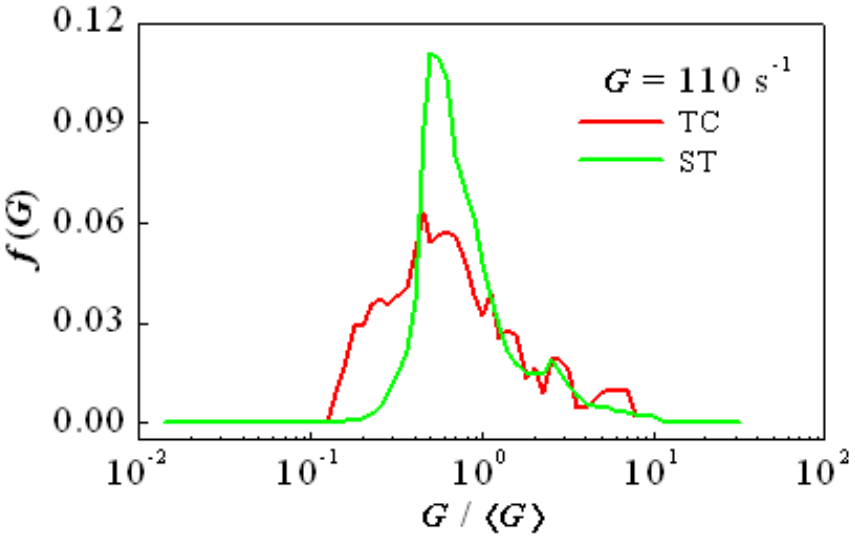
- Volume-averaged breakage kernel is

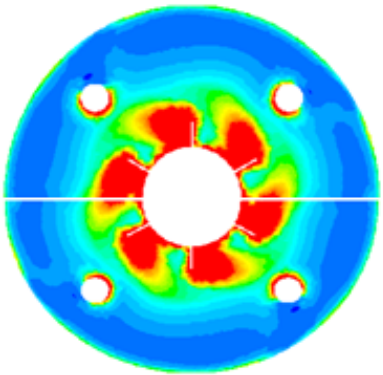
$$\overline{k}^B(G;\xi) = \int_0^\infty k^B(G^*;\xi) f(G^*)\,\mathrm{d}G^* \neq k^B(\overline{G};\xi)$$

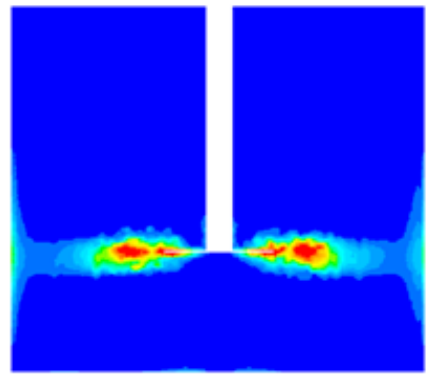
- Distribution function $f(G^*)$ accounts for fact that shear rate is not homogeneous in vessel

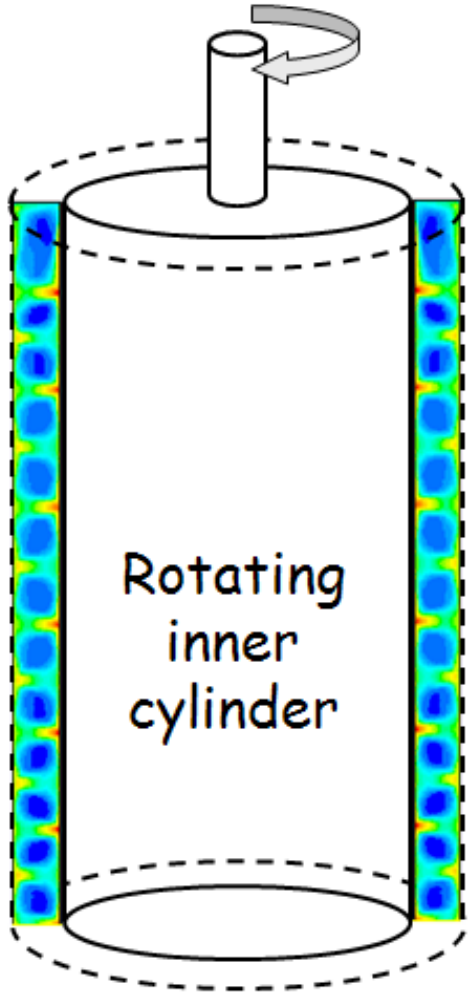


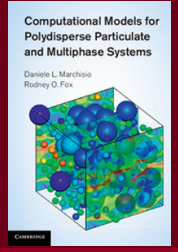
A Simple Example: Particle Aggregation and Breakage in CSTR





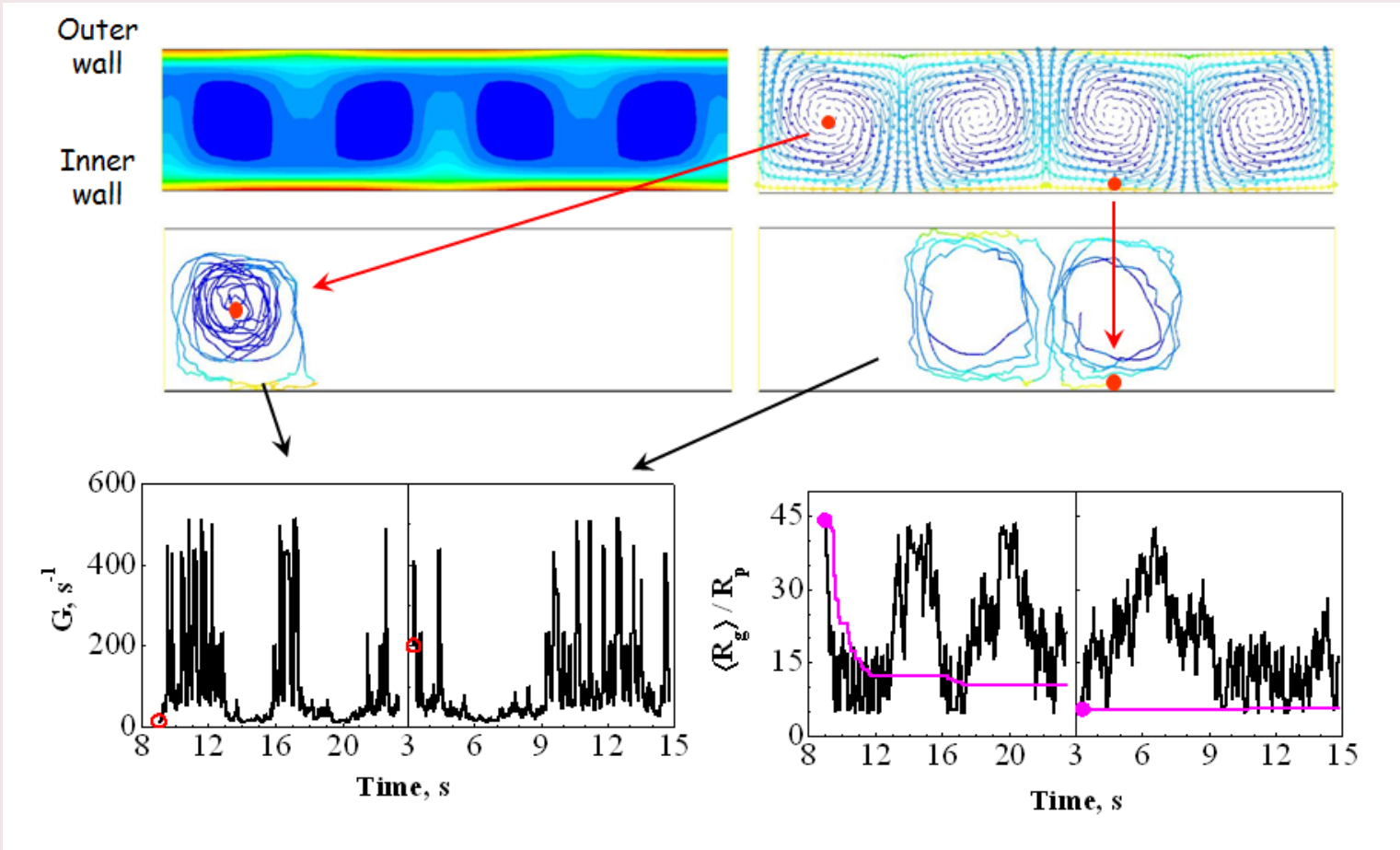


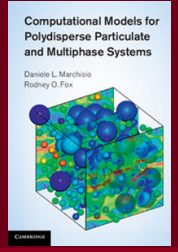




A Simple Example: Particle Aggregation and Breakage in CSTR

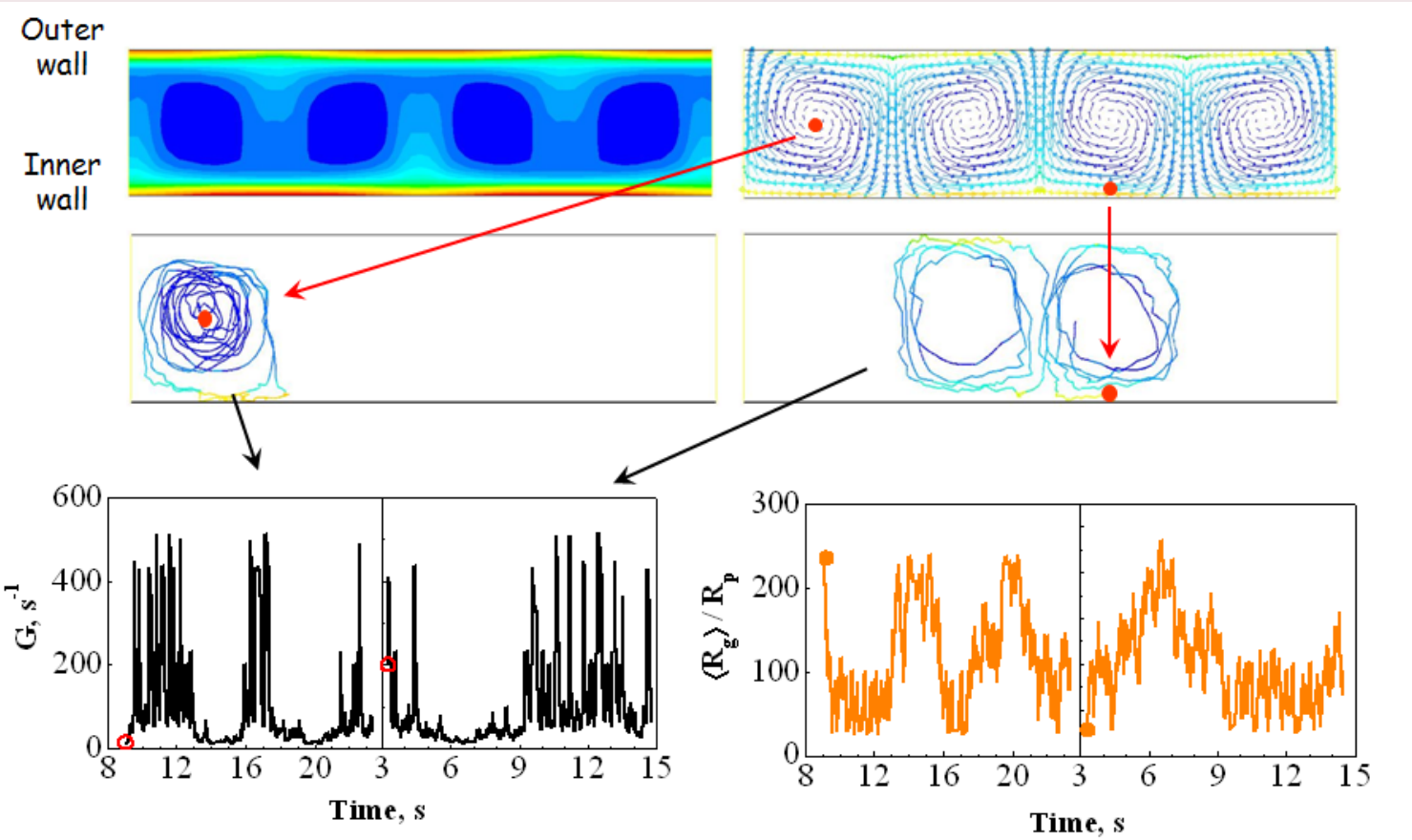
Mixing FASTER than aggregation and breakage $\tau_M \ll \tau_A, \tau_B$

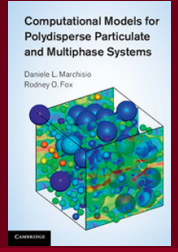




A Simple Example: Particle Aggregation and Breakage in CSTR

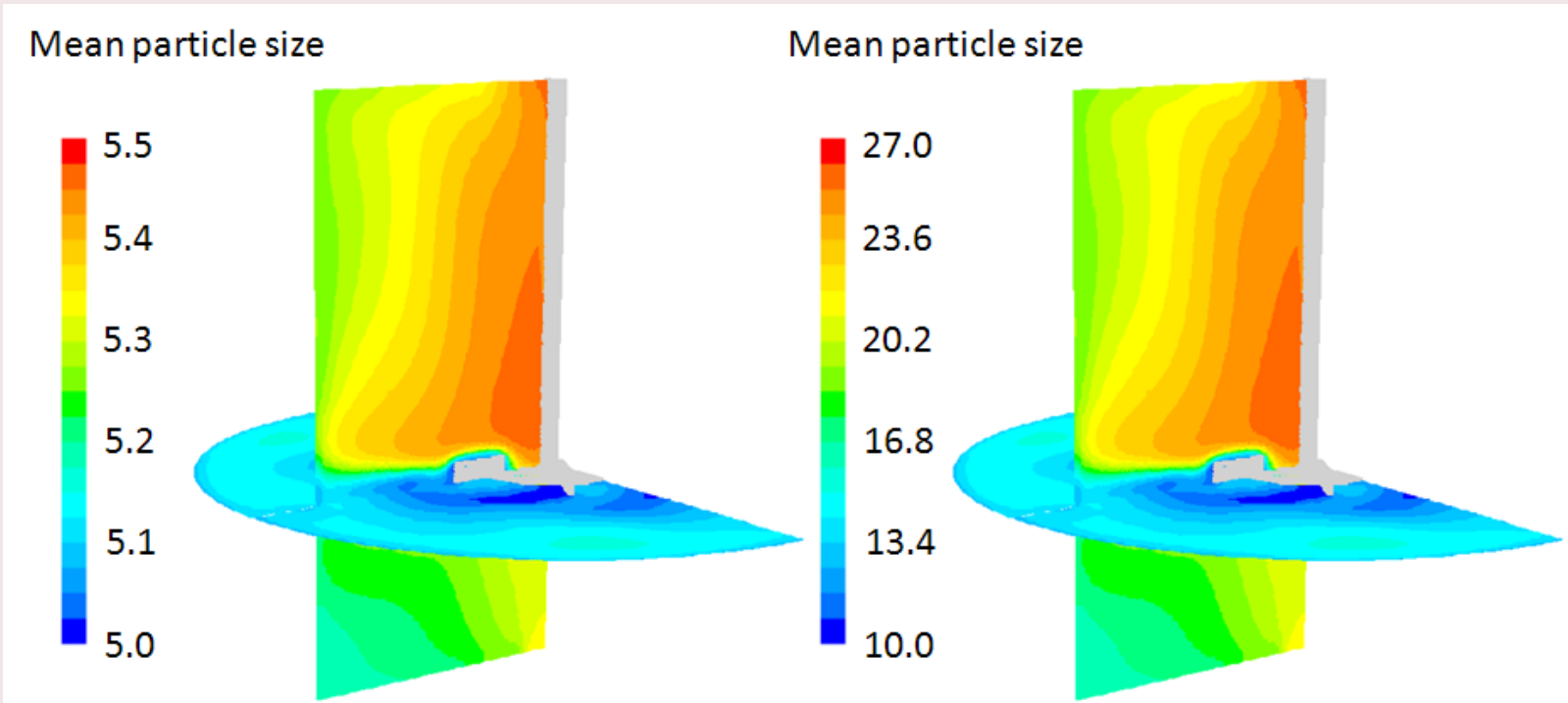
Mixing SLOWER than aggregation and breakage $\tau_M \gg \tau_A, \tau_B$

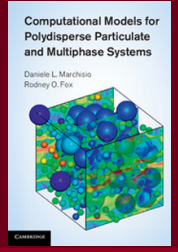




A Simple Example: Particle Aggregation and Breakage in CSTR

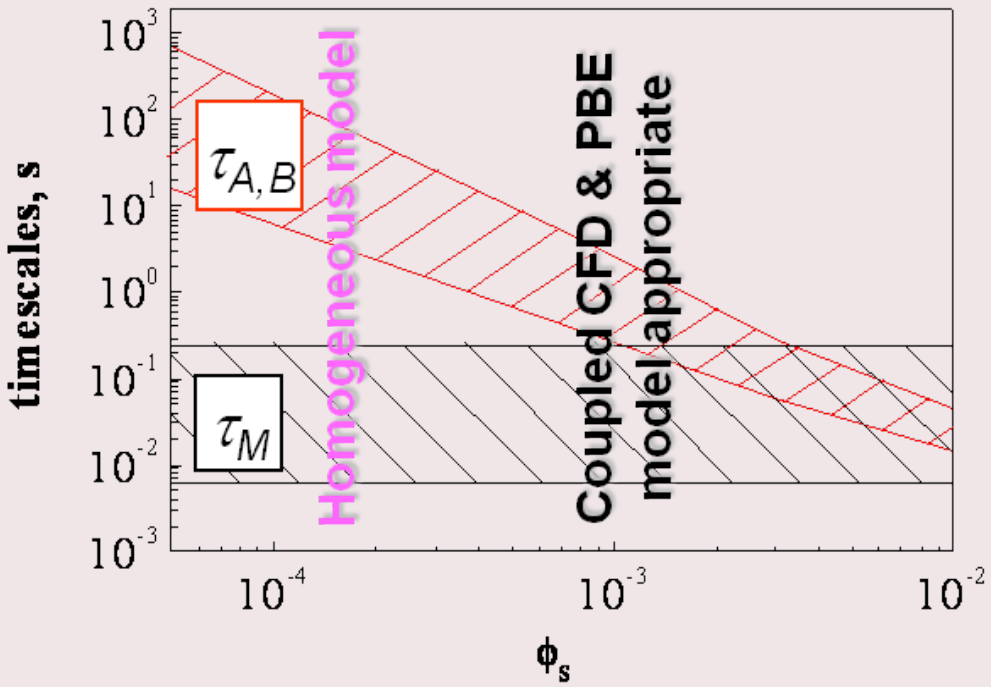
$\tau_M \ll \tau_A, \tau_B$ $\qquad$ $\tau_M \gg \tau_A, \tau_B$



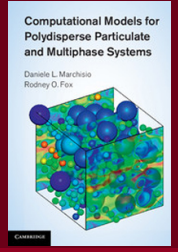


A Simple Example: Particle Aggregation and Breakage in CSTR

Need for CFD depends on ratio between different time scales!

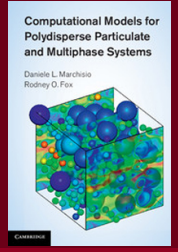


Marchisio et al. (2006) AIChE J. **52**, 158



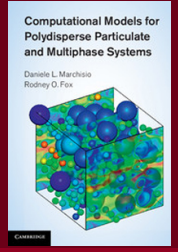
Summary

- Disperse multiphase systems can be modeled at different levels of detail
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- **Macroscale** simulations use mathematical approximations to approximate unclosed moments of NDF
- Thus, first step is to formulate a mesoscale model for polydisperse flows



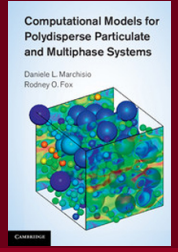
Summary

- Disperse multiphase systems can be modeled at different levels of detail
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- **Macroscale** simulations use mathematical approximations to approximate unclosed moments of NDF
- Thus, first step is to formulate a mesoscale model for polydisperse flows



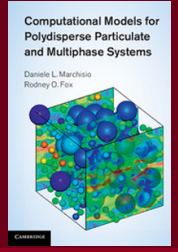
Summary

- Disperse multiphase systems can be modeled at different levels of detail
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- **Macroscale** simulations use mathematical approximations to approximate unclosed moments of NDF
- Thus, first step is to formulate a mesoscale model for polydisperse flows



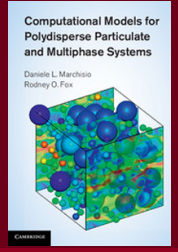
Summary

- Disperse multiphase systems can be modeled at different levels of detail
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- **Macroscale** simulations use mathematical approximations to approximate unclosed moments of NDF
- Thus, first step is to formulate a mesoscale model for polydisperse flows



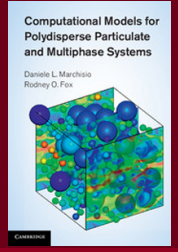
Summary

- Disperse multiphase systems can be modeled at different levels of detail
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- **Macroscale** simulations use mathematical approximations to approximate unclosed moments of NDF
- Thus, first step is to formulate a mesoscale model for polydisperse flows



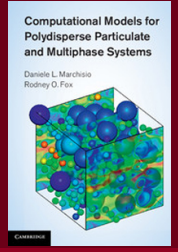
Summary

- Disperse multiphase systems can be modeled at different levels of detail
- **Microscale** simulations contain most detail, but are too expensive for most applications
- **Mesoscale** simulations use physical approximations to formulate a PBE or GPBE for NDF
- **Macroscale** simulations use mathematical approximations to approximate unclosed moments of NDF
- Thus, first step is to formulate a mesoscale model for polydisperse flows



Part 2

QUADRATURE-BASED MOMENT METHODS



Closure Problem

Closure problem appears always in following form:

$$I = \int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi$$

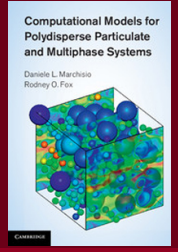
where $n(\xi)$ is unknown univariate NDF and Ω_ξ is integration interval

QBMM use a Gaussian quadrature

In Gaussian quadrature theory NDF is weight function or measure for which integer moments

$$m_k = m(k) = \langle \xi^k \rangle := \int_{\Omega_\xi} n(\xi)\, \xi^k\, \mathrm{d}\xi \quad k = 0, 1, 2, \ldots$$

must exist



Quadrature Theory

A set of polynomials $\{P_0(\xi), P_1(\xi), \ldots, P_\alpha(\xi), \ldots\}$ with

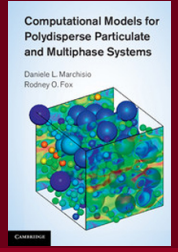
$$P_\alpha(\xi) = k_{\alpha,0}\xi^\alpha + k_{\alpha,1}\xi^{\alpha-1} + \cdots + k_{\alpha,\alpha}$$

is orthogonal in integration interval Ω_ξ, with respect to weight function, if

$$\int_{\Omega_\xi} n(\xi) P_\alpha(\xi) P_\beta(\xi)\, \mathrm{d}\xi \begin{cases} = 0 & \text{for } \alpha \neq \beta \\ > 0 & \text{for } \alpha = \beta \end{cases}$$

and, of course, is said to be orthonormal if

$$\int_{\Omega_\xi} n(\xi) P_\alpha(\xi) P_\beta(\xi)\, \mathrm{d}\xi = \begin{cases} 0 & \text{for } \alpha \neq \beta \\ 1 & \text{for } \alpha = \beta \end{cases}$$



Quadrature Theory

Any set of orthogonal polynomials $\{P_\alpha(\xi)\}$ has recurrence formula relating three consecutive polynomials in a sequence:

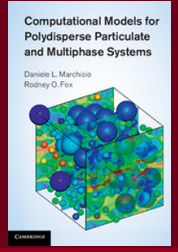
$$P_{\alpha+1}(\xi) = (\xi - a_\alpha)P_\alpha(\xi) - b_\alpha P_{\alpha-1}(\xi) \quad \alpha = 0, 1, 2, \ldots$$

with $P_{-1}(\xi) \equiv 0$ and $P_0(\xi) \equiv 1$ and

$$a_\alpha = \frac{\int_{\Omega_\xi} n(\xi)\xi P_\alpha(\xi)P_\alpha(\xi)\,\mathrm{d}\xi}{\int_{\Omega_\xi} n(\xi)P_\alpha(\xi)P_\alpha(\xi)\,\mathrm{d}\xi} \quad \text{for } \alpha = 0, 1, 2, \ldots$$

$$b_\alpha = \frac{\int_{\Omega_\xi} n(\xi)P_\alpha(\xi)P_\alpha(\xi)\,\mathrm{d}\xi}{\int_{\Omega_\xi} n(\xi)P_{\alpha-1}(\xi)P_{\alpha-1}(\xi)\,\mathrm{d}\xi} > 0 \quad \text{for } \alpha = 1, 2, \ldots$$

One can calculate a_0, then $P_1(\xi)$, then a_1 and b_1 and so on ...



Quadrature Theory

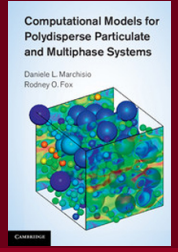
- Coefficients a_α and b_α can be written in terms of moments
- Coefficients necessary for construction of polynomial of order N can be calculated from first $2N-1$ moments of NDF
- For example with m_0, m_1, m_2 and m_3, it is possible to calculate the following coefficients:

$$a_0 = \frac{m_1}{m_0}$$

$$a_1 = \frac{m_3 m_0^2 + m_1^3 - 2m_2 m_1 m_0}{m_2 m_0 + m_1^2 - 2m_1^2 m_0}$$

$$b_1 = \frac{m_2 m_0 + m_1^2 - 2m_1^2 m_0}{m_0^2}$$

which suffice for calculation of polynomial $P_2(\xi)$



Quadrature Theory

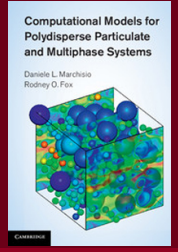
- Coefficients a_α and b_α can be written in terms of moments
- Coefficients necessary for construction of polynomial of order N can be calculated from first $2N-1$ moments of NDF
- For example with m_0, m_1, m_2 and m_3, it is possible to calculate the following coefficients:

$$a_0 = \frac{m_1}{m_0}$$

$$a_1 = \frac{m_3 m_0^2 + m_1^3 - 2m_2 m_1 m_0}{m_2 m_0 + m_1^2 - 2m_1^2 m_0}$$

$$b_1 = \frac{m_2 m_0 + m_1^2 - 2m_1^2 m_0}{m_0^2}$$

which suffice for calculation of polynomial $P_2(\xi)$



Quadrature Theory

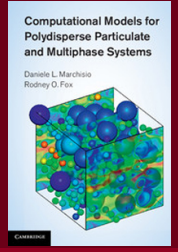
- Coefficients a_α and b_α can be written in terms of moments
- Coefficients necessary for construction of polynomial of order N can be calculated from first $2N-1$ moments of NDF
- For example with m_0, m_1, m_2 and m_3, it is possible to calculate the following coefficients:

$$a_0 = \frac{m_1}{m_0}$$

$$a_1 = \frac{m_3 m_0^2 + m_1^3 - 2m_2 m_1 m_0}{m_2 m_0 + m_1^2 - 2m_1^2 m_0}$$

$$b_1 = \frac{m_2 m_0 + m_1^2 - 2m_1^2 m_0}{m_0^2}$$

which suffice for calculation of polynomial $P_2(\xi)$



Quadrature Theory

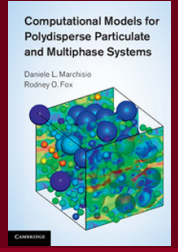
- Coefficients a_α and b_α can be written in terms of moments
- Coefficients necessary for construction of polynomial of order N can be calculated from first $2N-1$ moments of NDF
- For example with m_0, m_1, m_2 and m_3, it is possible to calculate the following coefficients:

$$a_0 = \frac{m_1}{m_0}$$

$$a_1 = \frac{m_3 m_0^2 + m_1^3 - 2 m_2 m_1 m_0}{m_2 m_0 + m_1^2 - 2 m_1^2 m_0}$$

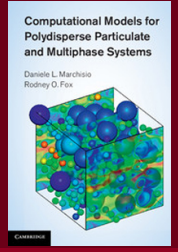
$$b_1 = \frac{m_2 m_0 + m_1^2 - 2 m_1^2 m_0}{m_0^2}$$

which suffice for calculation of polynomial $P_2(\xi)$



Quadrature-Based Moment Methods

Why are we interested in orthogonal polynomials?

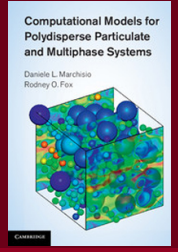


Quadrature-Based Moment Methods

- Closure problem can be overcome by using a quadrature formula:

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

  where w_α and ξ_α are weights and nodes/abscissas of quadrature formula, and N is number of nodes
- Accuracy of quadrature formula is quantified by its degree of accuracy
- Degree of accuracy is equal to d if quadrature formula is exact when the integrand is a polynomial of order less than or equal to d and there exists at least one polynomial of order $d+1$ that makes quadrature formula inexact



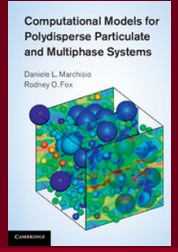
Quadrature-Based Moment Methods

- Closure problem can be overcome by using a quadrature formula:

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

  where w_α and ξ_α are weights and nodes/abscissas of quadrature formula, and N is number of nodes

- Accuracy of quadrature formula is quantified by its degree of accuracy
- Degree of accuracy is equal to d if quadrature formula is exact when the integrand is a polynomial of order less than or equal to d and there exists at least one polynomial of order $d+1$ that makes quadrature formula inexact

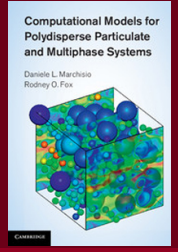


Quadrature-Based Moment Methods

- Closure problem can be overcome by using a quadrature formula:

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

  where w_α and ξ_α are weights and nodes/abscissas of quadrature formula, and N is number of nodes
- Accuracy of quadrature formula is quantified by its degree of accuracy
- Degree of accuracy is equal to d if quadrature formula is exact when the integrand is a polynomial of order less than or equal to d and there exists at least one polynomial of order $d+1$ that makes quadrature formula inexact

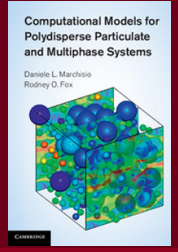


Quadrature-Based Moment Methods

- Closure problem can be overcome by using a quadrature formula:

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

  where w_α and ξ_α are weights and nodes/abscissas of quadrature formula, and N is number of nodes
- Accuracy of quadrature formula is quantified by its degree of accuracy
- Degree of accuracy is equal to d if quadrature formula is exact when the integrand is a polynomial of order less than or equal to d and there exists at least one polynomial of order $d+1$ that makes quadrature formula inexact



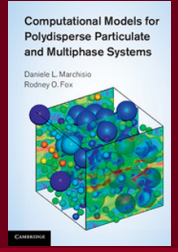
Quadrature-Based Moment Methods

GAUSSIAN QUADRATURE

Necessary and sufficient condition for following formula:

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi = \sum_{\alpha=1}^{N} g(\xi_\alpha) w_\alpha + R_N(g)$$

to be Gaussian quadrature approximation or, equivalently, that it has degree of accuracy of $2N - 1$, is that its nodes $\{\xi_\alpha\}$ coincide with the N roots of polynomial $P_N(\xi)$ of order N orthogonal in Ω_ξ with respect to weight function $n(\xi)$



Computing Quadrature Approximation

HOW DO WE CALCULATE THE QUADRATURE APPROX?

N weights and N abscissas can be determined by solving a non-linear system:

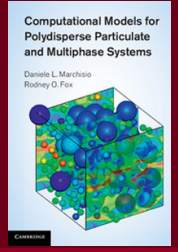
$$m_0 = \sum_{\alpha=1}^{N} w_\alpha$$

$$m_1 = \sum_{\alpha=1}^{N} w_\alpha \xi_\alpha$$

$$\vdots$$

$$m_{2N-1} = \sum_{\alpha=1}^{N} w_\alpha \xi_\alpha^{2N-1}$$

using Newton-Raphson method, or any other non-linear equation solver (very good initial guess needed to ensure convergence!)



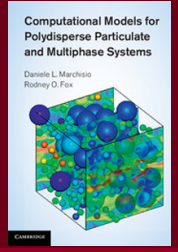
Computing Quadrature Approximation

Better way is to employ recursive relationship for orthogonal polynomials:

$$\xi \begin{bmatrix} P_0(\xi) \\ P_1(\xi) \\ \vdots \\ P_{N-2}(\xi) \\ P_{N-1}(\xi) \end{bmatrix} = \begin{bmatrix} a_0 & 1 & & & \\ b_1 & a_1 & 1 & & \\ & & \ddots & & \\ & & \ddots & 1 & \\ & & & a_{N-1} & \end{bmatrix} \begin{bmatrix} P_0(\xi) \\ P_1(\xi) \\ \vdots \\ P_{N-2}(\xi) \\ P_{N-1}(\xi) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ P_N(\xi) \end{bmatrix}$$

Nodes of quadrature approximation $\{\xi_\alpha\}$ (i.e., roots of $P_N(\xi)$), are eigenvalues of tridiagonal matrix appearing above

Matrix is re-written in terms of equivalent tridiagonal symmetric (Jacobi) matrix

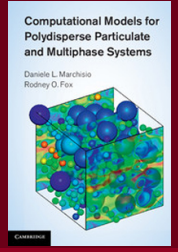


Computing Quadrature Approximation

Better way is to employ recursive relationship for orthogonal polynomials:

$$\xi \begin{bmatrix} P_0(\xi) \\ P_1(\xi) \\ \vdots \\ P_{N-2}(\xi) \\ P_{N-1}(\xi) \end{bmatrix} = \begin{bmatrix} a_0 & 1 & & & \\ b_1 & a_1 & 1 & & \\ & & \ddots & & \\ & & \ddots & 1 & \\ & & & a_{N-1} & \end{bmatrix} \begin{bmatrix} P_0(\xi) \\ P_1(\xi) \\ \vdots \\ P_{N-2}(\xi) \\ P_{N-1}(\xi) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ P_N(\xi) \end{bmatrix}$$

Nodes of quadrature approximation $\{\xi_\alpha\}$ (i.e., roots of $P_N(\xi)$), are eigenvalues of tridiagonal matrix appearing above
Matrix is re-written in terms of equivalent tridiagonal symmetric (Jacobi) matrix



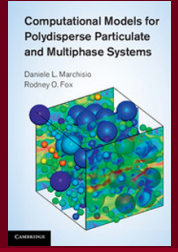
Computing Quadrature Approximation

Matrix can be made symmetric (preserving eigenvalues) by a diagonal similarity transformation to give a Jacobi matrix:

$$\mathbf{J} = \begin{bmatrix} a_0 & \sqrt{b_1} & & & & & \\ \sqrt{b_1} & a_1 & \sqrt{b_2} & & & & \\ & \sqrt{b_2} & a_2 & \sqrt{b_3} & & & \\ & & \sqrt{b_3} & a_3 & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & a_{N-2} & \sqrt{b_{N-1}} \\ & & & & & \sqrt{b_{N-1}} & a_{N-1} \end{bmatrix}$$

Procedure transforms ill-conditioned problem of finding roots of polynomial into well-conditioned problem of finding eigenvalues and eigenvectors of tridiagonal symmetric matrix

N weights are calculated as $w_\alpha = m_0 \varphi_{\alpha 1}^2$ where $\varphi_{\alpha 1}$ is first component of α^{th} eigenvector $\boldsymbol{\varphi}_\alpha$ of Jacobi matrix



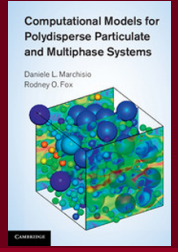
Computing Quadrature Approximation

Matrix can be made symmetric (preserving eigenvalues) by a diagonal similarity transformation to give a Jacobi matrix:

$$\mathbf{J} = \begin{bmatrix} a_0 & \sqrt{b_1} & & & & & \\ \sqrt{b_1} & a_1 & \sqrt{b_2} & & & & \\ & \sqrt{b_2} & a_2 & \sqrt{b_3} & & & \\ & & \sqrt{b_3} & a_3 & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & a_{N-2} & \sqrt{b_{N-1}} \\ & & & & & \sqrt{b_{N-1}} & a_{N-1} \end{bmatrix}$$

Procedure transforms ill-conditioned problem of finding roots of polynomial into well-conditioned problem of finding eigenvalues and eigenvectors of tridiagonal symmetric matrix

N weights are calculated as $w_\alpha = m_0 \varphi_{\alpha 1}^2$ where $\varphi_{\alpha 1}$ is first component of α^{th} eigenvector $\boldsymbol{\varphi}_\alpha$ of Jacobi matrix



Product-Difference Algorithm

1. Construct matrix $\mathbf{P}$ with components $P_{\alpha,\beta}$:

$$P_{\alpha,\beta} = P_{1,\beta-1}P_{\alpha+1,\beta-2} - P_{1,\beta-2}P_{\alpha+1,\beta-1}$$

$$\beta \in 3,\ldots,2N+1 \text{ and } \alpha \in 1,\ldots,2N+2-\beta$$

2. first row of matrix is

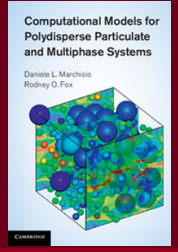
$$P_{\alpha,1} = \delta_{\alpha 1} \quad \alpha \in 1,\ldots,2N+1$$

3. $\delta_{\alpha 1}$ is Kronecker delta and components in second column of $\mathbf{P}$ are

$$P_{\alpha,2} = (-1)^{\alpha-1} m_{\alpha-1} \quad \alpha \in 1,\ldots,2N$$

4. Calculate coefficients of continued fraction $\{\zeta_\alpha\}$:

$$\zeta_\alpha = \frac{P_{1,\alpha+1}}{P_{1,\alpha}P_{1,\alpha-1}} \quad \alpha \in 2,\ldots,2N$$



Product-Difference Algorithm

1. Construct matrix $\mathbf{P}$ with components $P_{\alpha,\beta}$:

$$P_{\alpha,\beta} = P_{1,\beta-1}P_{\alpha+1,\beta-2} - P_{1,\beta-2}P_{\alpha+1,\beta-1}$$

$$\beta \in 3,\ldots,2N+1 \text{ and } \alpha \in 1,\ldots,2N+2-\beta$$

2. first row of matrix is

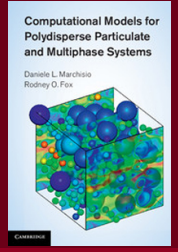
$$P_{\alpha,1} = \delta_{\alpha 1} \quad \alpha \in 1,\ldots,2N+1$$

3. $\delta_{\alpha 1}$ is Kronecker delta and components in second column of $\mathbf{P}$ are

$$P_{\alpha,2} = (-1)^{\alpha-1} m_{\alpha-1} \quad \alpha \in 1,\ldots,2N$$

4. Calculate coefficients of continued fraction $\{\zeta_\alpha\}$:

$$\zeta_\alpha = \frac{P_{1,\alpha+1}}{P_{1,\alpha}P_{1,\alpha-1}} \quad \alpha \in 2,\ldots,2N$$



Product-Difference Algorithm

1. Construct matrix $\mathbf{P}$ with components $P_{\alpha,\beta}$:

$$P_{\alpha,\beta} = P_{1,\beta-1}P_{\alpha+1,\beta-2} - P_{1,\beta-2}P_{\alpha+1,\beta-1}$$
$$\beta \in 3,\ldots,2N+1 \text{ and } \alpha \in 1,\ldots,2N+2-\beta$$

2. first row of matrix is

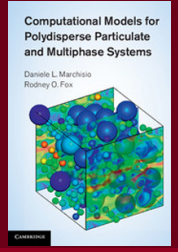
$$P_{\alpha,1} = \delta_{\alpha 1} \quad \alpha \in 1,\ldots,2N+1$$

3. $\delta_{\alpha 1}$ is Kronecker delta and components in second column of $\mathbf{P}$ are

$$P_{\alpha,2} = (-1)^{\alpha-1} m_{\alpha-1} \quad \alpha \in 1,\ldots,2N$$

4. Calculate coefficients of continued fraction $\{\zeta_\alpha\}$:

$$\zeta_\alpha = \frac{P_{1,\alpha+1}}{P_{1,\alpha}P_{1,\alpha-1}} \quad \alpha \in 2,\ldots,2N$$



Product-Difference Algorithm

1. Construct matrix $\mathbf{P}$ with components $P_{\alpha,\beta}$:

$$P_{\alpha,\beta} = P_{1,\beta-1}P_{\alpha+1,\beta-2} - P_{1,\beta-2}P_{\alpha+1,\beta-1}$$
$$\beta \in 3,\ldots,2N+1 \text{ and } \alpha \in 1,\ldots,2N+2-\beta$$

2. first row of matrix is

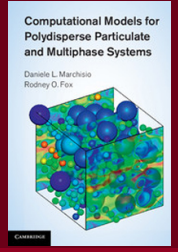
$$P_{\alpha,1} = \delta_{\alpha 1} \quad \alpha \in 1,\ldots,2N+1$$

3. $\delta_{\alpha 1}$ is Kronecker delta and components in second column of $\mathbf{P}$ are

$$P_{\alpha,2} = (-1)^{\alpha-1} m_{\alpha-1} \quad \alpha \in 1,\ldots,2N$$

4. Calculate coefficients of continued fraction $\{\zeta_\alpha\}$:

$$\zeta_\alpha = \frac{P_{1,\alpha+1}}{P_{1,\alpha}P_{1,\alpha-1}} \quad \alpha \in 2,\ldots,2N$$



Product-Difference Algorithm

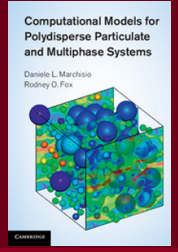
1. Coefficients of Jacobi matrix are obtained from sums and products of ζ_α:

$$a_\alpha = \zeta_{2\alpha} + \zeta_{2\alpha-1} \quad \alpha \in 1, \ldots, N$$

$$b_\alpha = \sqrt{\zeta_{2\alpha+1}\zeta_{2\alpha}} \quad \alpha \in 1, \ldots, N-1$$

2. For example with $N = 2$, $\mathbf{P}$ matrix is

$$\begin{bmatrix} 1 & m_0 & m_1 & m_0 m_2 - m_1^2 & m_0\left(m_3 m_1 - m_2^2\right) \\ 0 & -m_1 & -m_2 & -\left(m_0 m_3 - m_2 m_1\right) & \\ 0 & m_2 & m_3 & & \\ 0 & -m_3 & & & \\ 0 & & & & \end{bmatrix}$$



Product-Difference Algorithm

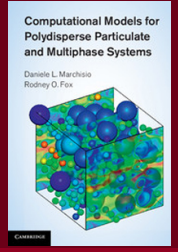
1. Coefficients of Jacobi matrix are obtained from sums and products of ζ_α:

$$a_\alpha = \zeta_{2\alpha} + \zeta_{2\alpha-1} \quad \alpha \in 1, \ldots, N$$

$$b_\alpha = \sqrt{\zeta_{2\alpha+1}\zeta_{2\alpha}} \quad \alpha \in 1, \ldots, N-1$$

2. For example with $N = 2$, $\mathbf{P}$ matrix is

$$\begin{bmatrix} 1 & m_0 & m_1 & m_0 m_2 - m_1^2 & m_0\left(m_3 m_1 - m_2^2\right) \\ 0 & -m_1 & -m_2 & -\left(m_0 m_3 - m_2 m_1\right) & \\ 0 & m_2 & m_3 & & \\ 0 & -m_3 & & & \\ 0 & & & & \end{bmatrix}$$



Product-Difference Algorithm

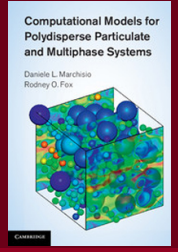
Consider a normal (or Gaussian) distribution:

$$n(\xi) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\xi-\mu)^2}{2\sigma^2}\right)$$

Moments of distribution:

$$m_0 = 1 \quad m_1 = \mu \quad m_2 = \mu^2 + \sigma^2$$
$$m_3 = \mu^3 + 3\mu\sigma^2$$
$$m_4 = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4$$
$$m_5 = \mu^5 + 10\mu^3\sigma^2 + 15\mu\sigma^4$$
$$m_6 = \mu^6 + 15\mu^4\sigma^2 + 45\mu^2\sigma^4 + 15\sigma^6$$
$$m_7 = \mu^7 + 21\mu^5\sigma^2 + 105\mu^3\sigma^4 + 105\mu\sigma^6$$
$$m_8 = \mu^8 + 28\mu^6\sigma^2 + 210\mu^4\sigma^4 + 420\mu^2\sigma^6 + 105\sigma^8$$

Use PD algorithm to calculate quadrature approximation of order four (i.e., $N = 4$) for $\mu = 5$ and $\sigma = 1$



Product-Difference Algorithm

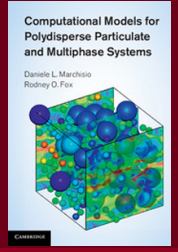
First eight moments of distribution are needed:

$$m_0 = 1 \quad m_1 = 5 \quad m_2 = 26 \quad m_3 = 140$$
$$m_4 = 778$$
$$m_5 = 4450$$
$$m_6 = 26140$$
$$m_7 = 157400$$

After applying PD algorithm Jacobi matrix is obtained:

$$\mathbf{J} = \begin{bmatrix} 5 & 1 & 0 & 0 \\ 1 & 5 & \sqrt{2} & 0 \\ 0 & \sqrt{2} & 5 & \sqrt{3} \\ 0 & 0 & \sqrt{3} & 5 \end{bmatrix}$$

resulting in: $w_1 = 0.0459$, $w_2 = 0.4541$, $w_3 = 0.4541$, $w_4 = 0.0459$, and $\xi_1 = 2.6656$, $\xi_2 = 4.2580$, $\xi_3 = 5.7420$, $\xi_4 = 7.3344$

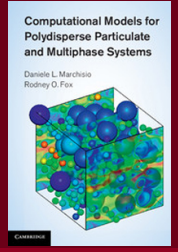


Product-Difference Algorithm

PD algorithm works well when ξ is positive, but not when distribution has zero mean (moment of order one null)
Example: Gaussian distribution with $\mu = 0$ and $\sigma = 1$

$$m_0 = 1$$
$$m_1 = 0$$
$$m_2 = 1$$
$$m_3 = 0$$
$$m_4 = 3$$
$$m_5 = 0$$
$$m_6 = 15$$
$$m_7 = 0$$

In general, Wheeler algorithm should be used (see Numerical Recipes for details)



Realizable Moments

Moment inversion algorithm only works if moments are realizable

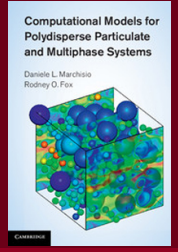
Definition: Realizable moments correspond to a non-negative NDF

Hankel matrices are used to check if moments $\{m_0, m_1, \ldots, m_{2N}\}$ are realizable:

$$\mathbf{H}_N := \begin{bmatrix} m_0 & m_1 & m_2 & \ldots & m_N \\ m_1 & m_2 & m_3 & \ldots & m_{N+1} \\ m_2 & m_3 & \ddots & \ldots & m_{N+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_N & m_{N+1} & m_{N+2} & \ldots & m_{2N} \end{bmatrix}$$

Moment set $\{m_0, m_1, \ldots, m_{2N}\}$ is realizable iff $|\mathbf{H}_N| \geq 0$

With N-node QMOM, $|\mathbf{H}_N| = 0$, but $|\mathbf{H}_n| > 0$ for $n = 0, 1, \ldots, N-1$



Realizable Moments

Example for $N = 2$:

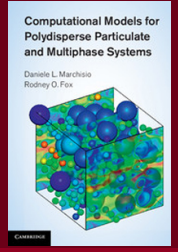
$$|\mathbf{H}_0| = m_0$$
$$|\mathbf{H}_1| = m_0 m_2 - m_1^2$$
$$|\mathbf{H}_2| = m_4(m_0 m_2 - m_1^2) - m_3(m_0 m_3 - m_1 m_2) + m_2(m_1 m_3 - m_2^2)$$

Or, using central moments, $C_0 = 1$, $C_1 = 0$, $C_2 =$ variance:

$$|\mathbf{H}_0| = 1$$
$$|\mathbf{H}_1| = C_2$$
$$|\mathbf{H}_2| = C_4 C_2 - C_3^2 - C_2^3$$

With 2-node QMOM, $|\mathbf{H}_2| = 0$, and $n^*(\xi) = w_1\delta(\xi - \xi_1) + w_2\delta(\xi - \xi_2)$

When $|\mathbf{H}_N| = 0$, but $|\mathbf{H}_n| > 0$ for $n = 0, 1, \ldots, N-1$; we say that **moments are on boundary of moment space**

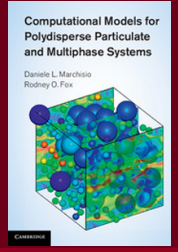


Moment Space

Moment vectors $\mathbf{M} = \{m_0, m_1, \ldots, m_{2N}\}$ live in finite moment space:

- Moment space is convex: let $\mathbf{M}_1$ and $\mathbf{M}_2$ be two sets of realizable moments, then $\mathbf{M}^* = a_1\mathbf{M}_1 + a_2\mathbf{M}_2$ is realizable when a_1 and a_2 are non-negative
- If $|\mathbf{H}_N| = 0$, then $\mathbf{M}$ is on boundary of moment space
- If $|\mathbf{H}_N| = 0$, then m_{2N} has smallest possible value
- If $|\mathbf{H}_N| = 0$, then n^* is sum of N weighted delta functions
- If $|\mathbf{H}_N| > 0$, then $\mathbf{M}$ is in interior of moment space
- If $|\mathbf{H}_N| > 0$, then $n^*(\xi)$ is continuous function of ξ

Quadrature-Based Moment Methods reconstruct $n^*(\xi)$ for the boundary of moment space by satisfying all moments $\{m_0, m_1, \ldots, m_{2N-1}\}$



Variation: Gauss–Radau Quadrature

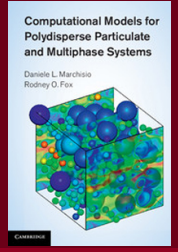
- In many cases, it is advantageous to include even-numbered moment m_{2N}: evaporation, nucleation, spatial fluxes, reconstruct continuous NDF (EQMOM), etc.
- Moment set $\mathbf{M} = \{m_0, m_1, \ldots, m_{2N}\}$ has $2N + 1$ degrees of freedom
- Gauss–Radau quadrature fixes one node ξ_0, but weight w_0 is free:

$$\int g(\xi) n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i g(\xi_i)$$

  e.g., ξ_0 is smallest particle mass in system
- There are $N + 1$ weights w_i and N nodes ξ_i that must be found from moments for $k = 0, 1, \ldots, 2N$:

$$m_k := \int \xi^k n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i \xi_i^k$$

- Small change in Wheeler algorithm (see Numerical Recipes)



Variation: Gauss–Radau Quadrature

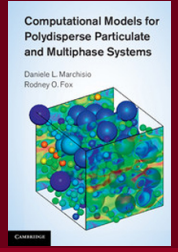
- In many cases, it is advantageous to include even-numbered moment m_{2N}: evaporation, nucleation, spatial fluxes, reconstruct continuous NDF (EQMOM), etc.
- Moment set $\mathbf{M} = \{m_0, m_1, \ldots, m_{2N}\}$ has $2N + 1$ degrees of freedom
- Gauss–Radau quadrature fixes one node ξ_0, but weight w_0 is free:

$$\int g(\xi) n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i g(\xi_i)$$

  e.g., ξ_0 is smallest particle mass in system
- There are $N + 1$ weights w_i and N nodes ξ_i that must be found from moments for $k = 0, 1, \ldots, 2N$:

$$m_k := \int \xi^k n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i \xi_i^k$$

- Small change in Wheeler algorithm (see Numerical Recipes)



Variation: Gauss–Radau Quadrature

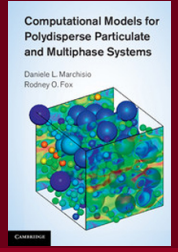
- In many cases, it is advantageous to include even-numbered moment m_{2N}: evaporation, nucleation, spatial fluxes, reconstruct continuous NDF (EQMOM), etc.
- Moment set $\mathbf{M} = \{m_0, m_1, \ldots, m_{2N}\}$ has $2N + 1$ degrees of freedom
- Gauss–Radau quadrature fixes one node ξ_0, but weight w_0 is free:

$$\int g(\xi) n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i g(\xi_i)$$

  e.g., ξ_0 is smallest particle mass in system
- There are $N + 1$ weights w_i and N nodes ξ_i that must be found from moments for $k = 0, 1, \ldots, 2N$:

$$m_k := \int \xi^k n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i \xi_i^k$$

- Small change in Wheeler algorithm (see Numerical Recipes)



Variation: Gauss–Radau Quadrature

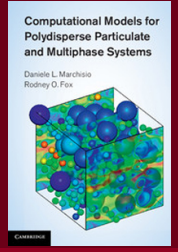
- In many cases, it is advantageous to include even-numbered moment m_{2N}: evaporation, nucleation, spatial fluxes, reconstruct continuous NDF (EQMOM), etc.
- Moment set $\mathbf{M} = \{m_0, m_1, \ldots, m_{2N}\}$ has $2N + 1$ degrees of freedom
- Gauss–Radau quadrature fixes one node ξ_0, but weight w_0 is free:

$$\int g(\xi) n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i g(\xi_i)$$

  e.g., ξ_0 is smallest particle mass in system
- There are $N + 1$ weights w_i and N nodes ξ_i that must be found from moments for $k = 0, 1, \ldots, 2N$:

$$m_k := \int \xi^k n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i \xi_i^k$$

- Small change in Wheeler algorithm (see Numerical Recipes)



Variation: Gauss–Radau Quadrature

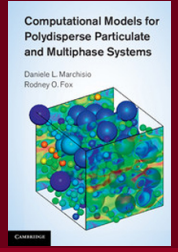
- In many cases, it is advantageous to include even-numbered moment m_{2N}: evaporation, nucleation, spatial fluxes, reconstruct continuous NDF (EQMOM), etc.
- Moment set $\mathbf{M} = \{m_0, m_1, \ldots, m_{2N}\}$ has $2N + 1$ degrees of freedom
- Gauss–Radau quadrature fixes one node ξ_0, but weight w_0 is free:

$$\int g(\xi) n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i g(\xi_i)$$

  e.g., ξ_0 is smallest particle mass in system
- There are $N + 1$ weights w_i and N nodes ξ_i that must be found from moments for $k = 0, 1, \ldots, 2N$:

$$m_k := \int \xi^k n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i \xi_i^k$$

- Small change in Wheeler algorithm (see Numerical Recipes)



Variation: Gauss–Radau Quadrature

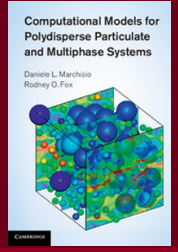
- In many cases, it is advantageous to include even-numbered moment m_{2N}: evaporation, nucleation, spatial fluxes, reconstruct continuous NDF (EQMOM), etc.
- Moment set $\mathbf{M} = \{m_0, m_1, \ldots, m_{2N}\}$ has $2N + 1$ degrees of freedom
- Gauss–Radau quadrature fixes one node ξ_0, but weight w_0 is free:

$$\int g(\xi) n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i g(\xi_i)$$

  e.g., ξ_0 is smallest particle mass in system
- There are $N + 1$ weights w_i and N nodes ξ_i that must be found from moments for $k = 0, 1, \ldots, 2N$:

$$m_k := \int \xi^k n(\xi)\, \mathrm{d}\xi = \sum_{i=0}^{N} w_i \xi_i^k$$

- Small change in Wheeler algorithm (see Numerical Recipes)



Summary of QBMM for Solving PBE

- Starting from PBE for $n(t,\xi)$, formally derived unclosed moment equations $\mathbf{M}(t) = \{m_0(t), m_1(t), \ldots, m_k(t)\} \,|\, k = 0, 1, \ldots, 2N$:

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = S_k(\mathbf{M}) \quad \Longrightarrow \quad \frac{\mathrm{d}\mathbf{M}}{\mathrm{d}t} = \mathbf{S}(\mathbf{M})$$

- Close moment equations with quadrature:

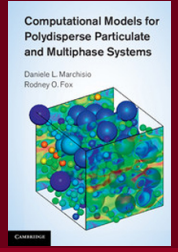
$$n(t,\xi) \approx n^*(t,\xi) = \sum_{i=0}^{N} w_i \delta(\xi - \xi_i)$$

where w_i and ξ_i are found from $\mathbf{M}(t)$ with Wheeler algorithm

- Use numerical ODE solver to advance in time:

$$\mathbf{M}(t + \Delta t) = \mathbf{M}(t) + \mathbf{S}(\mathbf{M}(t))\Delta t$$

given initial conditions $\mathbf{M}(0)$



Summary of QBMM for Solving PBE

- Starting from PBE for $n(t,\xi)$, formally derived unclosed moment equations $\mathbf{M}(t) = \{m_0(t), m_1(t), \ldots, m_k(t)\} \,|\, k = 0, 1, \ldots, 2N$:

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = S_k(\mathbf{M}) \quad \Longrightarrow \quad \frac{\mathrm{d}\mathbf{M}}{\mathrm{d}t} = \mathbf{S}(\mathbf{M})$$

- Close moment equations with quadrature:

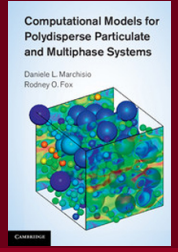
$$n(t,\xi) \approx n^*(t,\xi) = \sum_{i=0}^{N} w_i \delta(\xi - \xi_i)$$

where w_i and ξ_i are found from $\mathbf{M}(t)$ with Wheeler algorithm

- Use numerical ODE solver to advance in time:

$$\mathbf{M}(t + \Delta t) = \mathbf{M}(t) + \mathbf{S}(\mathbf{M}(t))\Delta t$$

given initial conditions $\mathbf{M}(0)$



Summary of QBMM for Solving PBE

- Starting from PBE for $n(t,\xi)$, formally derived unclosed moment equations $\mathbf{M}(t) = \{m_0(t), m_1(t), \ldots, m_k(t)\} \,|\, k = 0, 1, \ldots, 2N$:

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = S_k(\mathbf{M}) \quad \Longrightarrow \quad \frac{\mathrm{d}\mathbf{M}}{\mathrm{d}t} = \mathbf{S}(\mathbf{M})$$

- Close moment equations with quadrature:

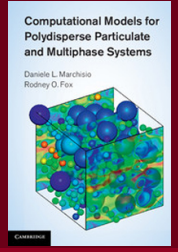
$$n(t,\xi) \approx n^*(t,\xi) = \sum_{i=0}^{N} w_i \delta(\xi - \xi_i)$$

where w_i and ξ_i are found from $\mathbf{M}(t)$ with Wheeler algorithm

- Use numerical ODE solver to advance in time:

$$\mathbf{M}(t + \Delta t) = \mathbf{M}(t) + \mathbf{S}(\mathbf{M}(t))\Delta t$$

given initial conditions $\mathbf{M}(0)$



Summary of QBMM for Solving PBE

- Starting from PBE for $n(t,\xi)$, formally derived unclosed moment equations $\mathbf{M}(t) = \{m_0(t), m_1(t), \ldots, m_k(t)\} \,|\, k = 0, 1, \ldots, 2N$:

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = S_k(\mathbf{M}) \quad \Longrightarrow \quad \frac{\mathrm{d}\mathbf{M}}{\mathrm{d}t} = \mathbf{S}(\mathbf{M})$$

- Close moment equations with quadrature:

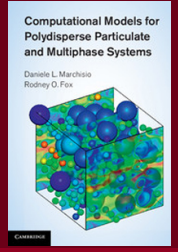
$$n(t,\xi) \approx n^*(t,\xi) = \sum_{i=0}^{N} w_i \delta(\xi - \xi_i)$$

where w_i and ξ_i are found from $\mathbf{M}(t)$ with Wheeler algorithm

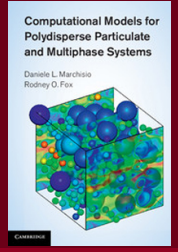
- Use numerical ODE solver to advance in time:

$$\mathbf{M}(t + \Delta t) = \mathbf{M}(t) + \mathbf{S}(\mathbf{M}(t))\Delta t$$

given initial conditions $\mathbf{M}(0)$



GENERALIZED POPULATION BALANCE EQUATION



From Microscale to Mesoscale Model

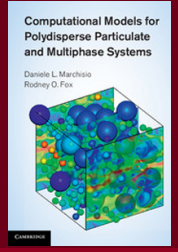
Microscale Description

Define phase space of *mesoscale* variables needed to describe a "particle" (velocity, volume, etc.)

Model changes to *one* particle's mesoscale variables due to all other particles, fluid, body forces, etc. (one-particle density function)

Mesoscale Model

Closure occurs at the level of the one-particle density function



NDF for Fluid–Particle Systems

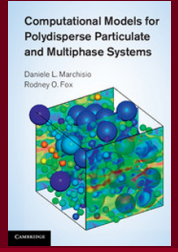
- Let us consider population of N_p particles characterized by variables: position $\{\mathbf{X}^{(n)}\}$, velocity $\{\mathbf{U}_\mathrm{p}^{(n)}\}$ and composition $\{\boldsymbol{\xi}_\mathrm{p}^{(n)}\}$
- State of n^{th} particle will change according to following Lagrangian (following particle) equations:

$$\frac{\mathrm{d}\mathbf{X}^{(n)}}{\mathrm{d}t} = \mathbf{U}_\mathrm{p}^{(n)}$$

$$\frac{\mathrm{d}\mathbf{U}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{A}_\mathrm{fp}^{(n)} + \mathbf{A}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}U}^{(n)}$$

$$\frac{\mathrm{d}\boldsymbol{\xi}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{G}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}\xi}^{(n)}$$

- Operators on right-hand sides (**mesoscale models**) depend on complete set of variables for all particles
- In principle, they can be found from **microscale simulations**



NDF for Fluid–Particle Systems

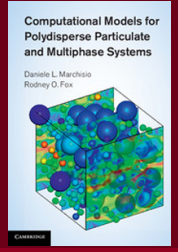
- Let us consider population of N_p particles characterized by variables: position $\{\mathbf{X}^{(n)}\}$, velocity $\{\mathbf{U}_\mathrm{p}^{(n)}\}$ and composition $\{\boldsymbol{\xi}_\mathrm{p}^{(n)}\}$
- State of n^{th} particle will change according to following Lagrangian (following particle) equations:

$$\frac{\mathrm{d}\mathbf{X}^{(n)}}{\mathrm{d}t} = \mathbf{U}_\mathrm{p}^{(n)}$$

$$\frac{\mathrm{d}\mathbf{U}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{A}_\mathrm{fp}^{(n)} + \mathbf{A}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}U}^{(n)}$$

$$\frac{\mathrm{d}\boldsymbol{\xi}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{G}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}\xi}^{(n)}$$

- Operators on right-hand sides (**mesoscale models**) depend on complete set of variables for all particles
- In principle, they can be found from **microscale simulations**



NDF for Fluid–Particle Systems

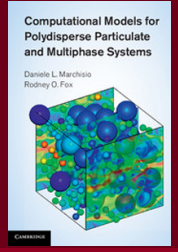
- Let us consider population of N_p particles characterized by variables: position $\{\mathbf{X}^{(n)}\}$, velocity $\{\mathbf{U}_\mathrm{p}^{(n)}\}$ and composition $\{\boldsymbol{\xi}_\mathrm{p}^{(n)}\}$
- State of n^{th} particle will change according to following Lagrangian (following particle) equations:

$$\frac{\mathrm{d}\mathbf{X}^{(n)}}{\mathrm{d}t} = \mathbf{U}_\mathrm{p}^{(n)}$$

$$\frac{\mathrm{d}\mathbf{U}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{A}_\mathrm{fp}^{(n)} + \mathbf{A}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}U}^{(n)}$$

$$\frac{\mathrm{d}\boldsymbol{\xi}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{G}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}\xi}^{(n)}$$

- Operators on right-hand sides (**mesoscale models**) depend on complete set of variables for all particles
- In principle, they can be found from **microscale simulations**



NDF for Fluid–Particle Systems

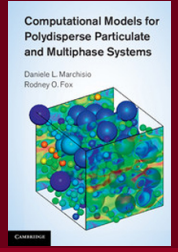
- Let us consider population of N_p particles characterized by variables: position $\{\mathbf{X}^{(n)}\}$, velocity $\{\mathbf{U}_\mathrm{p}^{(n)}\}$ and composition $\{\boldsymbol{\xi}_\mathrm{p}^{(n)}\}$
- State of n^{th} particle will change according to following Lagrangian (following particle) equations:

$$\frac{\mathrm{d}\mathbf{X}^{(n)}}{\mathrm{d}t} = \mathbf{U}_\mathrm{p}^{(n)}$$

$$\frac{\mathrm{d}\mathbf{U}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{A}_\mathrm{fp}^{(n)} + \mathbf{A}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}U}^{(n)}$$

$$\frac{\mathrm{d}\boldsymbol{\xi}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{G}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}\xi}^{(n)}$$

- Operators on right-hand sides (**mesoscale models**) depend on complete set of variables for all particles
- In principle, they can be found from **microscale simulations**



NDF for Fluid–Particle Systems

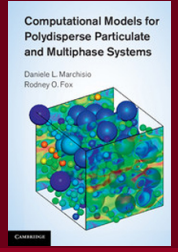
- Let us consider population of N_p particles characterized by variables: position $\{\mathbf{X}^{(n)}\}$, velocity $\{\mathbf{U}_\mathrm{p}^{(n)}\}$ and composition $\{\boldsymbol{\xi}_\mathrm{p}^{(n)}\}$
- State of n^{th} particle will change according to following Lagrangian (following particle) equations:

$$\frac{\mathrm{d}\mathbf{X}^{(n)}}{\mathrm{d}t} = \mathbf{U}_\mathrm{p}^{(n)}$$

$$\frac{\mathrm{d}\mathbf{U}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{A}_\mathrm{fp}^{(n)} + \mathbf{A}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}U}^{(n)}$$

$$\frac{\mathrm{d}\boldsymbol{\xi}_\mathrm{p}^{(n)}}{\mathrm{d}t} = \mathbf{G}_\mathrm{p}^{(n)} + \mathbf{C}_{\mathrm{p}\xi}^{(n)}$$

- Operators on right-hand sides (**mesoscale models**) depend on complete set of variables for all particles
- In principle, they can be found from **microscale simulations**

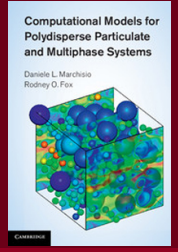


NDF for Fluid–Particle Systems

- In most descriptions of fluid–particle flows, mesoscale fluid–particle acceleration $\mathbf{A}_{\text{fp}}^{(n)}$ includes a drag term of form

$$\mathbf{A}_{\text{fp}}^{(n)}(t) \propto C_{\text{D}}^{(n)} \left|\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right| \left[\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right]$$

  where $C_{\text{D}}^{(n)}$ is drag coefficient, and $\mathbf{U}_{\text{f}}^{(n)}(t)$ is characteristic fluid velocity in neighborhood of $\mathbf{X}^{(n)}(t)$
- $\mathbf{U}_{\text{f}}^{(n)}(t) = \mathbf{U}_{\text{f}}(t, \mathbf{X}^{(n)}(t)) + \mathbf{u}_{\text{f}}^{\prime(n)}(t)$ is called "fluid velocity seen by n^{th} solid particle", which is difficult to model *a priori* but is included in set of particle properties to be tracked
- In general it will be necessary to introduce some additional internal coordinates representing other fluid properties seen by n^{th} particle: $\boldsymbol{\xi}_{\text{f}}^{(n)}$



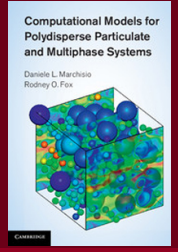
NDF for Fluid–Particle Systems

- In most descriptions of fluid–particle flows, mesoscale fluid–particle acceleration $\mathbf{A}_{\text{fp}}^{(n)}$ includes a drag term of form

$$\mathbf{A}_{\text{fp}}^{(n)}(t) \propto C_{\text{D}}^{(n)} \left|\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right| \left[\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right]$$

  where $C_{\text{D}}^{(n)}$ is drag coefficient, and $\mathbf{U}_{\text{f}}^{(n)}(t)$ is characteristic fluid velocity in neighborhood of $\mathbf{X}^{(n)}(t)$

- $\mathbf{U}_{\text{f}}^{(n)}(t) = \mathbf{U}_{\text{f}}(t, \mathbf{X}^{(n)}(t)) + \mathbf{u}_{\text{f}}^{\prime(n)}(t)$ is called "fluid velocity seen by n^{th} solid particle", which is difficult to model *a priori* but is included in set of particle properties to be tracked
- In general it will be necessary to introduce some additional internal coordinates representing other fluid properties seen by n^{th} particle: $\boldsymbol{\xi}_{\text{f}}^{(n)}$

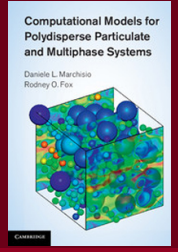


NDF for Fluid–Particle Systems

- In most descriptions of fluid–particle flows, mesoscale fluid–particle acceleration $\mathbf{A}_{\text{fp}}^{(n)}$ includes a drag term of form

$$\mathbf{A}_{\text{fp}}^{(n)}(t) \propto C_{\text{D}}^{(n)} \left|\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right| \left[\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right]$$

  where $C_{\text{D}}^{(n)}$ is drag coefficient, and $\mathbf{U}_{\text{f}}^{(n)}(t)$ is characteristic fluid velocity in neighborhood of $\mathbf{X}^{(n)}(t)$
- $\mathbf{U}_{\text{f}}^{(n)}(t) = \mathbf{U}_{\text{f}}(t, \mathbf{X}^{(n)}(t)) + \mathbf{u}_{\text{f}}'^{(n)}(t)$ is called "fluid velocity seen by n^{th} solid particle", which is difficult to model *a priori* but is included in set of particle properties to be tracked
- In general it will be necessary to introduce some additional internal coordinates representing other fluid properties seen by n^{th} particle: $\boldsymbol{\xi}_{\text{f}}^{(n)}$

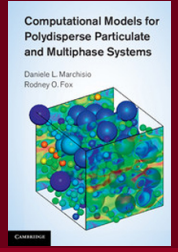


NDF for Fluid–Particle Systems

- In most descriptions of fluid–particle flows, mesoscale fluid–particle acceleration $\mathbf{A}_{\text{fp}}^{(n)}$ includes a drag term of form

$$\mathbf{A}_{\text{fp}}^{(n)}(t) \propto C_{\text{D}}^{(n)} \left|\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right| \left[\mathbf{U}_{\text{f}}^{(n)}(t) - \mathbf{U}_{\text{p}}^{(n)}(t)\right]$$

  where $C_{\text{D}}^{(n)}$ is drag coefficient, and $\mathbf{U}_{\text{f}}^{(n)}(t)$ is characteristic fluid velocity in neighborhood of $\mathbf{X}^{(n)}(t)$
- $\mathbf{U}_{\text{f}}^{(n)}(t) = \mathbf{U}_{\text{f}}(t, \mathbf{X}^{(n)}(t)) + \mathbf{u}_{\text{f}}^{\prime(n)}(t)$ is called "fluid velocity seen by n^{th} solid particle", which is difficult to model *a priori* but is included in set of particle properties to be tracked
- In general it will be necessary to introduce some additional internal coordinates representing other fluid properties seen by n^{th} particle: $\boldsymbol{\xi}_{\text{f}}^{(n)}$



NDF for Fluid–Particle Systems

- State of N_p particles immersed in continuous phase is completely defined by ***multi-particle–fluid joint PDF***:

$$f_{N_\mathrm{p}}\left(t, \{\mathbf{x}^{(n)}\}, \{\mathbf{V}_\mathrm{p}^{(n)}\}, \{\boldsymbol{\eta}_\mathrm{p}^{(n)}\}, \{\mathbf{V}_\mathrm{f}^{(n)}\}, \{\boldsymbol{\eta}_\mathrm{f}^{(n)}\}\right) \mathrm{d}\{\mathbf{x}^{(n)}\}\,\mathrm{d}\{\mathbf{V}_\mathrm{p}^{(n)}\}\,\mathrm{d}\{\boldsymbol{\eta}_\mathrm{p}^{(n)}\}\,\mathrm{d}\{\mathbf{V}_\mathrm{f}^{(n)}\}\,\mathrm{d}\{\boldsymbol{\eta}_\mathrm{f}^{(n)}\}$$

$$:=$$

$$P\Big[\cap_{n=1}^{N_\mathrm{p}}\Big\{\Big(\mathbf{x}^{(n)} < \mathbf{X}^{(n)}(t) \le \mathbf{x}^{(n)} + \mathrm{d}\mathbf{x}^{(n)}\Big) \qquad \text{particle positions}$$

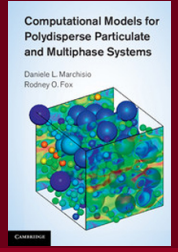
$$\cap\Big(\mathbf{V}_\mathrm{p}^{(n)} < \mathbf{U}_\mathrm{p}^{(n)}(t) \le \mathbf{V}_\mathrm{p}^{(n)} + \mathrm{d}\mathbf{V}_\mathrm{p}^{(n)}\Big) \qquad \text{particle velocities}$$

$$\cap\Big(\boldsymbol{\eta}_\mathrm{p}^{(n)} < \boldsymbol{\xi}_\mathrm{p}^{(n)}(t) \le \boldsymbol{\eta}_\mathrm{p}^{(n)} + \mathrm{d}\boldsymbol{\eta}_\mathrm{p}^{(n)}\Big) \qquad \text{particle compositions}$$

$$\cap\Big(\mathbf{V}_\mathrm{f}^{(n)} < \mathbf{U}_\mathrm{f}^{(n)}(t) \le \mathbf{V}_\mathrm{f}^{(n)} + \mathrm{d}\mathbf{V}_\mathrm{f}^{(n)}\Big) \qquad \text{seen fluid velocities}$$

$$\cap\Big(\boldsymbol{\eta}_\mathrm{f}^{(n)} < \boldsymbol{\xi}_\mathrm{f}^{(n)}(t) \le \boldsymbol{\eta}_\mathrm{f}^{(n)} + \mathrm{d}\boldsymbol{\eta}_\mathrm{f}^{(n)}\Big)\Big\}\Big] \qquad \text{seen fluid compositions}$$

In principle, it is known from **microscale simulations**



NDF for Fluid–Particle Systems

- Multi-particle joint PDF can be reduced to ***single-particle joint PDF*** by integrating out state variables for all particles except n^{th}:

$$f_1^{(n)}\left(t, \mathbf{x}^{(n)}, \mathbf{V}_{\text{p}}^{(n)}, \boldsymbol{\eta}_{\text{p}}^{(n)}, \mathbf{V}_{\text{f}}^{(n)}, \boldsymbol{\eta}_{\text{f}}^{(n)}\right) :=$$

$$\int_{m \neq n} f_{N_{\text{p}}}\left(t, \{\mathbf{x}^{(m)}\}, \{\mathbf{V}_{\text{p}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{p}}^{(m)}\}, \{\mathbf{V}_{\text{f}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{f}}^{(m)}\}\right) \mathrm{d}\mathbf{x}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{p}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{p}}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{f}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{f}}^{(m)}$$

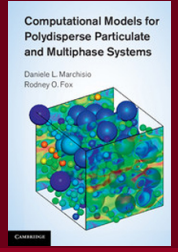
- **Fluid–particle NDF** is therefore defined as

$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) := \sum_{n=1}^{N_{\text{p}}} f_1^{(n)}(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

- In limit of identically distributed particles NDF becomes:

$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) = N_{\text{p}} f_1(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

i.e. numbering is arbitrary



NDF for Fluid–Particle Systems

- Multi-particle joint PDF can be reduced to ***single-particle joint PDF*** by integrating out state variables for all particles except n^{th}:

$$f_1^{(n)}\left(t, \mathbf{x}^{(n)}, \mathbf{V}_{\text{p}}^{(n)}, \boldsymbol{\eta}_{\text{p}}^{(n)}, \mathbf{V}_{\text{f}}^{(n)}, \boldsymbol{\eta}_{\text{f}}^{(n)}\right) :=$$

$$\int_{m \neq n} f_{N_{\text{p}}}\left(t, \{\mathbf{x}^{(m)}\}, \{\mathbf{V}_{\text{p}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{p}}^{(m)}\}, \{\mathbf{V}_{\text{f}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{f}}^{(m)}\}\right) \mathrm{d}\mathbf{x}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{p}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{p}}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{f}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{f}}^{(m)}$$

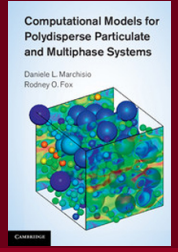
- **Fluid–particle NDF** is therefore defined as

$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) := \sum_{n=1}^{N_{\text{p}}} f_1^{(n)}(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

- In limit of identically distributed particles NDF becomes:

$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) = N_{\text{p}} f_1(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

i.e. numbering is arbitrary



NDF for Fluid–Particle Systems

- Multi-particle joint PDF can be reduced to ***single-particle joint PDF*** by integrating out state variables for all particles except n^{th}:

$$f_1^{(n)}\left(t, \mathbf{x}^{(n)}, \mathbf{V}_{\text{p}}^{(n)}, \boldsymbol{\eta}_{\text{p}}^{(n)}, \mathbf{V}_{\text{f}}^{(n)}, \boldsymbol{\eta}_{\text{f}}^{(n)}\right) :=$$

$$\int_{m \neq n} f_{N_{\text{p}}}\left(t, \{\mathbf{x}^{(m)}\}, \{\mathbf{V}_{\text{p}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{p}}^{(m)}\}, \{\mathbf{V}_{\text{f}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{f}}^{(m)}\}\right) \mathrm{d}\mathbf{x}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{p}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{p}}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{f}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{f}}^{(m)}$$

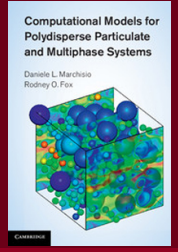
- **Fluid–particle NDF** is therefore defined as

$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) := \sum_{n=1}^{N_{\text{p}}} f_1^{(n)}(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

- In limit of identically distributed particles NDF becomes:

$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) = N_{\text{p}} f_1(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

i.e. numbering is arbitrary



NDF for Fluid–Particle Systems

- Multi-particle joint PDF can be reduced to ***single-particle joint PDF*** by integrating out state variables for all particles except n^{th}:

$$f_1^{(n)}\left(t, \mathbf{x}^{(n)}, \mathbf{V}_{\text{p}}^{(n)}, \boldsymbol{\eta}_{\text{p}}^{(n)}, \mathbf{V}_{\text{f}}^{(n)}, \boldsymbol{\eta}_{\text{f}}^{(n)}\right) :=$$

$$\int_{m\neq n} f_{N_{\text{p}}}\left(t, \{\mathbf{x}^{(m)}\}, \{\mathbf{V}_{\text{p}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{p}}^{(m)}\}, \{\mathbf{V}_{\text{f}}^{(m)}\}, \{\boldsymbol{\eta}_{\text{f}}^{(m)}\}\right) \mathrm{d}\mathbf{x}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{p}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{p}}^{(m)}\, \mathrm{d}\mathbf{V}_{\text{f}}^{(m)}\, \mathrm{d}\boldsymbol{\eta}_{\text{f}}^{(m)}$$

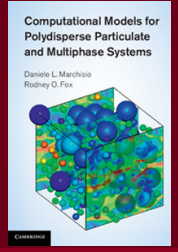
- **Fluid–particle NDF** is therefore defined as

$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) := \sum_{n=1}^{N_{\text{p}}} f_1^{(n)}(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

- In limit of identically distributed particles NDF becomes:

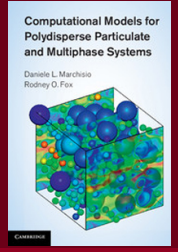
$$n(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}}) = N_{\text{p}} f_1(t, \mathbf{x}, \mathbf{v}_{\text{p}}, \boldsymbol{\xi}_{\text{p}}, \mathbf{v}_{\text{f}}, \boldsymbol{\xi}_{\text{f}})$$

  i.e. numbering is arbitrary



NDF for Fluid–Particle Systems

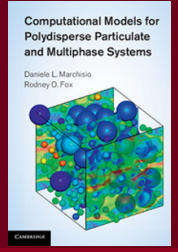
Based on these definitions, how can we derive transport equations for fluid–particle systems?



GPBE for Fluid–Particle Systems

- Starting from mesoscale models for particle properties (validated with DNS of fluid–particle system when possible), we first derive generalized transport equation for f_{N_p} (generalized Liouville equation)
- From this equation we find transport equation for one-particle PDF $f_1^{(n)}$ by integrating out all degrees of freedom except those associated with n^{th} particle (loss of information and generation of unclosed terms)
- Finally, using definition of fluid–particle NDF we derive GPBE:

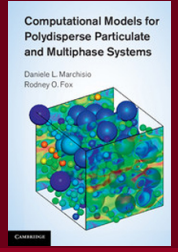
$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$



GPBE for Fluid–Particle Systems

- Starting from mesoscale models for particle properties (validated with DNS of fluid–particle system when possible), we first derive generalized transport equation for f_{N_p} (generalized Liouville equation)
- From this equation we find transport equation for one-particle PDF $f_1^{(n)}$ by integrating out all degrees of freedom except those associated with n^{th} particle (loss of information and generation of unclosed terms)
- Finally, using definition of fluid–particle NDF we derive GPBE:

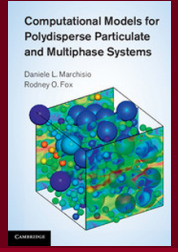
$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$



GPBE for Fluid–Particle Systems

- Starting from mesoscale models for particle properties (validated with DNS of fluid–particle system when possible), we first derive generalized transport equation for f_{N_p} (generalized Liouville equation)
- From this equation we find transport equation for one-particle PDF $f_1^{(n)}$ by integrating out all degrees of freedom except those associated with n^{th} particle (loss of information and generation of unclosed terms)
- Finally, using definition of fluid–particle NDF we derive GPBE:

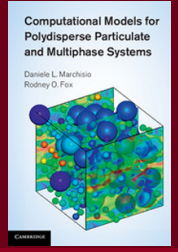
$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$



GPBE for Fluid–Particle Systems

- Starting from mesoscale models for particle properties (validated with DNS of fluid–particle system when possible), we first derive generalized transport equation for f_{N_p} (generalized Liouville equation)
- From this equation we find transport equation for one-particle PDF $f_1^{(n)}$ by integrating out all degrees of freedom except those associated with n^{th} particle (loss of information and generation of unclosed terms)
- Finally, using definition of fluid–particle NDF we derive GPBE:

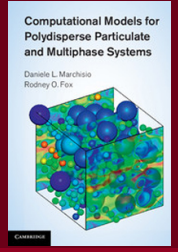
$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$



GPBE for Fluid–Particle Systems

Summary

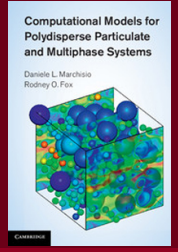
- We consider only mesoscale models where phase-space fluxes and source terms in GPBE are closed, in other words, depend only on independent variables $(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$ and on NDF $n(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$
- Finding accurate mesoscale closures is highly non-trivial task (e.g. drag coefficient for dense particulate systems, aggregation integrals and kernels, etc.)
- In order to reduce dimensionality of GPBE it is sometimes possible to work with selected **moments of NDF** by solving their transport equations
- Let us see how to derive moment equations and what additional problems are generated in their derivation (**moment closure problem**)



GPBE for Fluid–Particle Systems

Summary

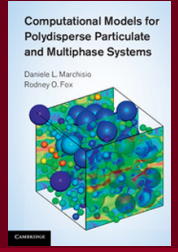
- We consider only mesoscale models where phase-space fluxes and source terms in GPBE are closed, in other words, depend only on independent variables $(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$ and on NDF $n(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$
- Finding accurate mesoscale closures is highly non-trivial task (e.g. drag coefficient for dense particulate systems, aggregation integrals and kernels, etc.)
- In order to reduce dimensionality of GPBE it is sometimes possible to work with selected **moments of NDF** by solving their transport equations
- Let us see how to derive moment equations and what additional problems are generated in their derivation (**moment closure problem**)



GPBE for Fluid–Particle Systems

Summary

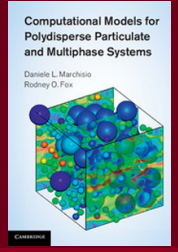
- We consider only mesoscale models where phase-space fluxes and source terms in GPBE are closed, in other words, depend only on independent variables $(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$ and on NDF $n(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$
- Finding accurate mesoscale closures is highly non-trivial task (e.g. drag coefficient for dense particulate systems, aggregation integrals and kernels, etc.)
- In order to reduce dimensionality of GPBE it is sometimes possible to work with selected **moments of NDF** by solving their transport equations
- Let us see how to derive moment equations and what additional problems are generated in their derivation (**moment closure problem**)



GPBE for Fluid–Particle Systems

Summary

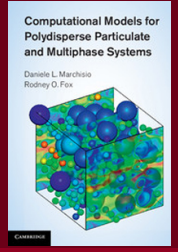
- We consider only mesoscale models where phase-space fluxes and source terms in GPBE are closed, in other words, depend only on independent variables $(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$ and on NDF $n(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$
- Finding accurate mesoscale closures is highly non-trivial task (e.g. drag coefficient for dense particulate systems, aggregation integrals and kernels, etc.)
- In order to reduce dimensionality of GPBE it is sometimes possible to work with selected **moments of NDF** by solving their transport equations
- Let us see how to derive moment equations and what additional problems are generated in their derivation (**moment closure problem**)



GPBE for Fluid–Particle Systems

Summary

- We consider only mesoscale models where phase-space fluxes and source terms in GPBE are closed, in other words, depend only on independent variables $(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$ and on NDF $n(t, \mathbf{x}, \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$
- Finding accurate mesoscale closures is highly non-trivial task (e.g. drag coefficient for dense particulate systems, aggregation integrals and kernels, etc.)
- In order to reduce dimensionality of GPBE it is sometimes possible to work with selected **moments of NDF** by solving their transport equations
- Let us see how to derive moment equations and what additional problems are generated in their derivation (**moment closure problem**)

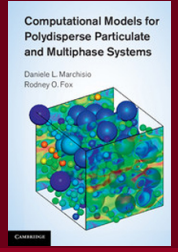


Moment Transport Equations

- At this point we will not enter into details of mesoscale models but it is worth mentioning that a set of consistency constraints has to be satisfied

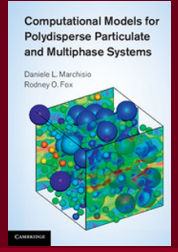
- Mesoscale models are designed to satisfy consistency constraints (e.g., conservation of mass, species, momentum, energy, etc.)

- Operation of deriving moment transport equations requires manipulation of multivariate integrals, it is therefore useful to review some integration rules



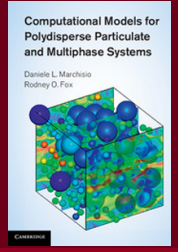
Moment Transport Equations

- At this point we will not enter into details of mesoscale models but it is worth mentioning that a set of consistency constraints has to be satisfied
- Mesoscale models are designed to satisfy consistency constraints (e.g., conservation of mass, species, momentum, energy, etc.)
- Operation of deriving moment transport equations requires manipulation of multivariate integrals, it is therefore useful to review some integration rules



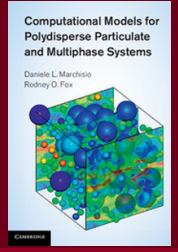
Moment Transport Equations

- At this point we will not enter into details of mesoscale models but it is worth mentioning that a set of consistency constraints has to be satisfied
- Mesoscale models are designed to satisfy consistency constraints (e.g., conservation of mass, species, momentum, energy, etc.)
- Operation of deriving moment transport equations requires manipulation of multivariate integrals, it is therefore useful to review some integration rules



Moment Transport Equations

- At this point we will not enter into details of mesoscale models but it is worth mentioning that a set of consistency constraints has to be satisfied
- Mesoscale models are designed to satisfy consistency constraints (e.g., conservation of mass, species, momentum, energy, etc.)
- Operation of deriving moment transport equations requires manipulation of multivariate integrals, it is therefore useful to review some integration rules



Moment Transport Equations

Treatment of time and space derivatives

Let $g(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$ be arbitrary function of its variables
Then

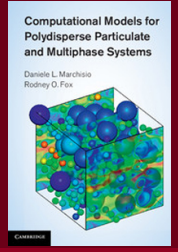
$$\int \left(\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n \right) g \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

$$= \int \left(\frac{\partial gn}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} gn \right) \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

$$= \frac{\partial}{\partial t} \left(\int gn \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} \right) + \frac{\partial}{\partial \mathbf{x}} \cdot \left(\int \mathbf{v}_\mathrm{p} gn \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} \right)$$

$$= \frac{\partial}{\partial t} \langle g \rangle + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{v}_\mathrm{p} g \rangle$$

where $\langle \bullet \rangle$ denotes integral w.r.t. n



Moment Transport Equations

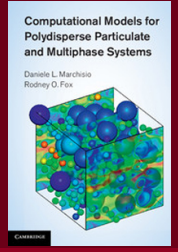
Treatment of velocity derivatives

With velocity derivatives, it is necessary to use integration by parts:

$$\int g \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

$$= \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) gn \Big|_{\mathbf{v}_\mathrm{p} = \pm\infty} - \int \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) \cdot \frac{\partial g}{\partial \mathbf{v}_\mathrm{p}} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

First term corresponds to flux normal to surface bounding velocity phase space and since velocity space extends to infinity, this flux must be null



Moment Transport Equations

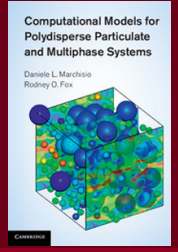
Treatment of internal-coordinate derivatives

Also for internal-coordinate derivatives it is necessary to use integration by parts:

$$\int g \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

$$= \langle \mathbf{G}_\mathrm{p} \rangle_1 g n \big|_{\boldsymbol{\xi}_\mathrm{p}=\max} - \langle \mathbf{G}_\mathrm{p} \rangle_1 g n \big|_{\boldsymbol{\xi}_\mathrm{p}=\min} - \int \langle \mathbf{G}_\mathrm{p} \rangle_1 \cdot \frac{\partial g}{\partial \boldsymbol{\xi}_\mathrm{p}} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

Since for particle internal coordinates phase space does not often extend to infinity, flux normal to phase-space boundaries might not be null (e.g. particle dissolution, droplet evaporation)



Disperse-Phase Number Transport

- Total number concentration $N(t, \mathbf{x})$ corresponds to zero-order moment of NDF (i.e., $g = 1$) and is defined by

$$N := \int n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Its transport equation can be found from GPBE

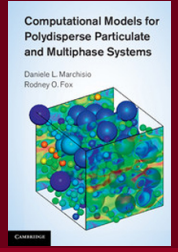
$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_\mathrm{N} N = \mathcal{S}_\mathrm{N}$$

- Number-average disperse-phase velocity is defined by

$$\mathbf{U}_\mathrm{N} := \frac{1}{N} \int \mathbf{v}_\mathrm{p} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Particle-number source term by

$$\mathcal{S}_\mathrm{N} := \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$



Disperse-Phase Number Transport

- Total number concentration $N(t, \mathbf{x})$ corresponds to zero-order moment of NDF (i.e., $g = 1$) and is defined by

$$N := \int n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Its transport equation can be found from GPBE

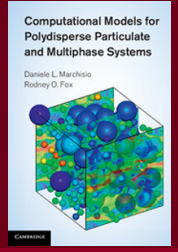
$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_\mathrm{N} N = \mathcal{S}_\mathrm{N}$$

- Number-average disperse-phase velocity is defined by

$$\mathbf{U}_\mathrm{N} := \frac{1}{N} \int \mathbf{v}_\mathrm{p} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Particle-number source term by

$$\mathcal{S}_\mathrm{N} := \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$



Disperse-Phase Number Transport

- Total number concentration $N(t, \mathbf{x})$ corresponds to zero-order moment of NDF (i.e., $g = 1$) and is defined by

$$N := \int n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Its transport equation can be found from GPBE

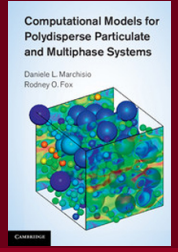
$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_\mathrm{N} N = \mathcal{S}_\mathrm{N}$$

- Number-average disperse-phase velocity is defined by

$$\mathbf{U}_\mathrm{N} := \frac{1}{N} \int \mathbf{v}_\mathrm{p} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Particle-number source term by

$$\mathcal{S}_\mathrm{N} := \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$



Disperse-Phase Number Transport

- Total number concentration $N(t, \mathbf{x})$ corresponds to zero-order moment of NDF (i.e., $g = 1$) and is defined by

$$N := \int n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Its transport equation can be found from GPBE

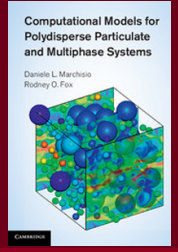
$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_\mathrm{N} N = \mathcal{S}_\mathrm{N}$$

- Number-average disperse-phase velocity is defined by

$$\mathbf{U}_\mathrm{N} := \frac{1}{N} \int \mathbf{v}_\mathrm{p} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Particle-number source term by

$$\mathcal{S}_\mathrm{N} := \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$



Disperse-Phase Number Transport

- Total number concentration $N(t, \mathbf{x})$ corresponds to zero-order moment of NDF (i.e., $g = 1$) and is defined by

$$N := \int n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Its transport equation can be found from GPBE

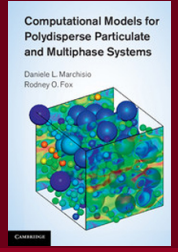
$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_\mathrm{N} N = \mathcal{S}_\mathrm{N}$$

- Number-average disperse-phase velocity is defined by

$$\mathbf{U}_\mathrm{N} := \frac{1}{N} \int \mathbf{v}_\mathrm{p} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

- Particle-number source term by

$$\mathcal{S}_\mathrm{N} := \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$



Disperse-Phase Volume Transport

- If first internal coordinate ξ_{p1} is particle volume (V_p) disperse-phase volume fraction is defined as ($g = \xi_{p1} = V_p$):

$$\alpha_p := \int \xi_{p1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Its transport equation can be derived from GPBE:

$$\frac{\partial \alpha_p}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_p \mathbf{U}_V = [\![G_p]\!]_V + \mathcal{S}_V$$

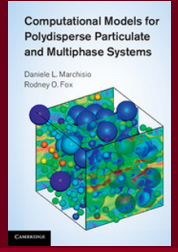
- Volume-average disperse-phase velocity is defined by

$$\mathbf{U}_V := \frac{1}{\alpha_p} \int \xi_{p1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Particle-volume source terms are defined by

$$[\![G_p]\!]_V := \int \langle G_{p1} \rangle_1 n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

$$\mathcal{S}_V := \int \xi_{p1} \mathcal{S}_1 \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Disperse-Phase Volume Transport

- If first internal coordinate ξ_{p1} is particle volume (V_p) disperse-phase volume fraction is defined as ($g = \xi_{p1} = V_p$):

$$\alpha_p := \int \xi_{p1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Its transport equation can be derived from GPBE:

$$\frac{\partial \alpha_p}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_p \mathbf{U}_V = [\![G_p]\!]_V + \mathcal{S}_V$$

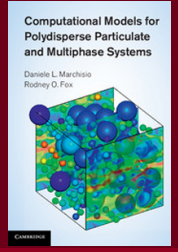
- Volume-average disperse-phase velocity is defined by

$$\mathbf{U}_V := \frac{1}{\alpha_p} \int \xi_{p1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Particle-volume source terms are defined by

$$[\![G_p]\!]_V := \int \langle G_{p1} \rangle_1 n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

$$\mathcal{S}_V := \int \xi_{p1} \mathcal{S}_1 \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Disperse-Phase Volume Transport

- If first internal coordinate ξ_{p1} is particle volume (V_p) disperse-phase volume fraction is defined as ($g = \xi_{p1} = V_p$):

$$\alpha_p := \int \xi_{p1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Its transport equation can be derived from GPBE:

$$\frac{\partial \alpha_p}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_p \mathbf{U}_V = [\![G_p]\!]_V + \mathcal{S}_V$$

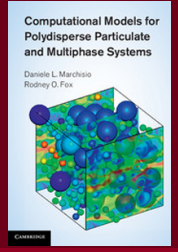
- Volume-average disperse-phase velocity is defined by

$$\mathbf{U}_V := \frac{1}{\alpha_p} \int \xi_{p1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Particle-volume source terms are defined by

$$[\![G_p]\!]_V := \int \langle G_{p1} \rangle_1 n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

$$\mathcal{S}_V := \int \xi_{p1} \mathcal{S}_1 \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Disperse-Phase Volume Transport

- If first internal coordinate ξ_{p1} is particle volume (V_p) disperse-phase volume fraction is defined as ($g = \xi_{p1} = V_p$):

$$\alpha_p := \int \xi_{p1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Its transport equation can be derived from GPBE:

$$\frac{\partial \alpha_p}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_p \mathbf{U}_V = [\![G_p]\!]_V + \mathcal{S}_V$$

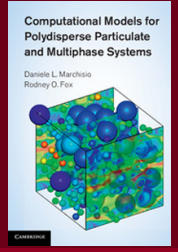
- Volume-average disperse-phase velocity is defined by

$$\mathbf{U}_V := \frac{1}{\alpha_p} \int \xi_{p1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Particle-volume source terms are defined by

$$[\![G_p]\!]_V := \int \langle G_{p1} \rangle_1 n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

$$\mathcal{S}_V := \int \xi_{p1} \mathcal{S}_1 \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Disperse-Phase Volume Transport

- If first internal coordinate ξ_{p1} is particle volume (V_p) disperse-phase volume fraction is defined as ($g = \xi_{p1} = V_p$):

$$\alpha_p := \int \xi_{p1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Its transport equation can be derived from GPBE:

$$\frac{\partial \alpha_p}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_p \mathbf{U}_V = [\![G_p]\!]_V + \mathcal{S}_V$$

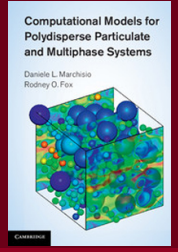
- Volume-average disperse-phase velocity is defined by

$$\mathbf{U}_V := \frac{1}{\alpha_p} \int \xi_{p1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Particle-volume source terms are defined by

$$[\![G_p]\!]_V := \int \langle G_{p1} \rangle_1 n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

$$\mathcal{S}_V := \int \xi_{p1} \mathcal{S}_1 \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Fluid-Phase Volume Transport

- If we let internal coordinate ξ_{f1} be equal to fluid volume *seen* by particle, fluid-phase volume fraction is

$$\alpha_f := \int \xi_{f1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Volume-average fluid velocity is

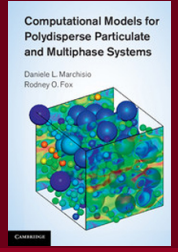
$$\mathbf{U}_{fV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_f n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Transport equation for α_f is

$$\frac{\partial \alpha_f}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_f \mathbf{U}_{pV} = [\![G_f]\!]_V + S_{fV}$$

- where

$$\mathbf{U}_{pV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Fluid-Phase Volume Transport

- If we let internal coordinate ξ_{f1} be equal to fluid volume *seen* by particle, fluid-phase volume fraction is

$$\alpha_f := \int \xi_{f1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Volume-average fluid velocity is

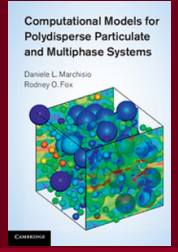
$$\mathbf{U}_{fV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_f n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Transport equation for α_f is

$$\frac{\partial \alpha_f}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_f \mathbf{U}_{pV} = [\![G_f]\!]_V + \mathcal{S}_{fV}$$

- where

$$\mathbf{U}_{pV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Fluid-Phase Volume Transport

- If we let internal coordinate ξ_{f1} be equal to fluid volume *seen* by particle, fluid-phase volume fraction is

$$\alpha_f := \int \xi_{f1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Volume-average fluid velocity is

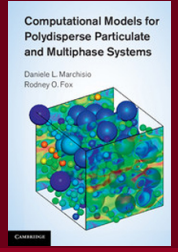
$$\mathbf{U}_{fV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_f n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Transport equation for α_f is

$$\frac{\partial \alpha_f}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_f \mathbf{U}_{pV} = [\![G_f]\!]_V + \mathcal{S}_{fV}$$

- where

$$\mathbf{U}_{pV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Fluid-Phase Volume Transport

- If we let internal coordinate ξ_{f1} be equal to fluid volume *seen* by particle, fluid-phase volume fraction is

$$\alpha_f := \int \xi_{f1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Volume-average fluid velocity is

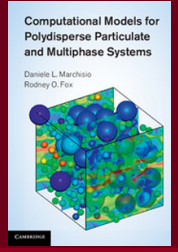
$$\mathbf{U}_{fV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_f n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Transport equation for α_f is

$$\frac{\partial \alpha_f}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_f \mathbf{U}_{pV} = [\![G_f]\!]_V + \mathcal{S}_{fV}$$

- where

$$\mathbf{U}_{pV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Fluid-Phase Volume Transport

- If we let internal coordinate ξ_{f1} be equal to fluid volume *seen* by particle, fluid-phase volume fraction is

$$\alpha_f := \int \xi_{f1} n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Volume-average fluid velocity is

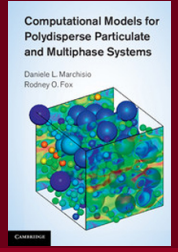
$$\mathbf{U}_{fV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_f n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$

- Transport equation for α_f is

$$\frac{\partial \alpha_f}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_f \mathbf{U}_{pV} = [\![G_f]\!]_V + \mathcal{S}_{fV}$$

- where

$$\mathbf{U}_{pV} := \frac{1}{\alpha_f} \int \xi_{f1} \mathbf{v}_p n \, d\mathbf{v}_p \, d\boldsymbol{\xi}_p \, d\mathbf{v}_f \, d\boldsymbol{\xi}_f$$



Fluid-Phase Volume Transport

- By definition, $\alpha_{\mathrm{p}} + \alpha_{\mathrm{f}} = 1$ and, in the absence of mass transfer between phases:

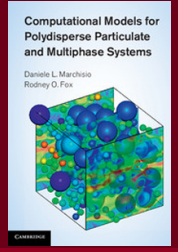
$$\frac{\partial \alpha_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{fV}} = 0$$

- This simply states that fluid-phase volume fraction is conserved quantity and therefore we must have:

$$[\![G_{\mathrm{f}}]\!]_{\mathrm{V}} = \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \left(\mathbf{U}_{\mathrm{pV}} - \mathbf{U}_{\mathrm{fV}} \right)$$

- In case of no mass transfer it will be necessary for $\langle G_{\mathrm{f1}} \rangle_1$ to be nonzero: volume of fluid seen by a particle must change along a particle trajectory due to presence of other particles!

$$\langle G_{\mathrm{f1}} \rangle_1 = \frac{1}{n} \frac{\partial}{\partial \mathbf{x}} \cdot \left(\mathbf{v}_{\mathrm{p}} - \mathbf{v}_{\mathrm{f}} \right) \xi_{\mathrm{f1}} n$$



Disperse-Phase Mass Transport

- If disperse phase has particles with different volumes and different masses (or material densities), at least two internal coordinates are necessary
- First internal coordinate particle volume and second particle mass ($\xi_{\mathrm{p}2} := M_{\mathrm{p}} = \rho_{\mathrm{p}} V_{\mathrm{p}}$; ρ_{p} is material density)
- Disperse-phase mass density is defined by

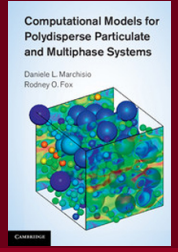
$$\varrho_{\mathrm{p}} := \int \xi_{\mathrm{p}2} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2} = M_{\mathrm{p}}$)

$$\frac{\partial \varrho_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{M}} = [\![G_{\mathrm{p}}]\!] + \mathcal{S}_{\mathrm{M}}$$

- Mass-average disperse-phase velocity

$$\mathbf{U}_{\mathrm{M}} := \frac{1}{\varrho_{\mathrm{p}}} \int \xi_{\mathrm{p}2} \mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Mass Transport

- If disperse phase has particles with different volumes and different masses (or material densities), at least two internal coordinates are necessary
- First internal coordinate particle volume and second particle mass ($\xi_{\mathrm{p}2} := M_{\mathrm{p}} = \rho_{\mathrm{p}} V_{\mathrm{p}}$; ρ_{p} is material density)
- Disperse-phase mass density is defined by

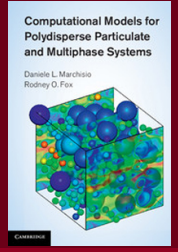
$$\varrho_{\mathrm{p}} := \int \xi_{\mathrm{p}2} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2} = M_{\mathrm{p}}$)

$$\frac{\partial \varrho_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{M}} = [\![G_{\mathrm{p}}]\!] + \mathcal{S}_{\mathrm{M}}$$

- Mass-average disperse-phase velocity

$$\mathbf{U}_{\mathrm{M}} := \frac{1}{\varrho_{\mathrm{p}}} \int \xi_{\mathrm{p}2} \mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Mass Transport

- If disperse phase has particles with different volumes and different masses (or material densities), at least two internal coordinates are necessary
- First internal coordinate particle volume and second particle mass ($\xi_{\mathrm{p}2} := M_{\mathrm{p}} = \rho_{\mathrm{p}} V_{\mathrm{p}}$; ρ_{p} is material density)
- Disperse-phase mass density is defined by

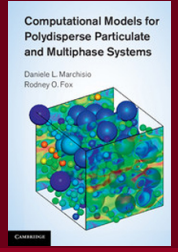
$$\varrho_{\mathrm{p}} := \int \xi_{\mathrm{p}2} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2} = M_{\mathrm{p}}$)

$$\frac{\partial \varrho_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{M}} = [\![G_{\mathrm{p}}]\!] + \mathcal{S}_{\mathrm{M}}$$

- Mass-average disperse-phase velocity

$$\mathbf{U}_{\mathrm{M}} := \frac{1}{\varrho_{\mathrm{p}}} \int \xi_{\mathrm{p}2} \mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Mass Transport

- If disperse phase has particles with different volumes and different masses (or material densities), at least two internal coordinates are necessary
- First internal coordinate particle volume and second particle mass ($\xi_{\mathrm{p}2} := M_{\mathrm{p}} = \rho_{\mathrm{p}} V_{\mathrm{p}}$; ρ_{p} is material density)
- Disperse-phase mass density is defined by

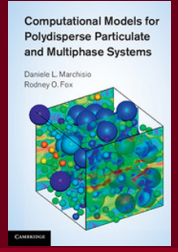
$$\varrho_{\mathrm{p}} := \int \xi_{\mathrm{p}2} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2} = M_{\mathrm{p}}$)

$$\frac{\partial \varrho_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{M}} = [\![G_{\mathrm{p}}]\!] + S_{\mathrm{M}}$$

- Mass-average disperse-phase velocity

$$\mathbf{U}_{\mathrm{M}} := \frac{1}{\varrho_{\mathrm{p}}} \int \xi_{\mathrm{p}2} \mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Mass Transport

- If disperse phase has particles with different volumes and different masses (or material densities), at least two internal coordinates are necessary
- First internal coordinate particle volume and second particle mass ($\xi_{\mathrm{p}2} := M_{\mathrm{p}} = \rho_{\mathrm{p}} V_{\mathrm{p}}$; ρ_{p} is material density)
- Disperse-phase mass density is defined by

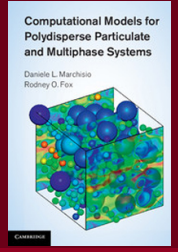
$$\varrho_{\mathrm{p}} := \int \xi_{\mathrm{p}2} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2} = M_{\mathrm{p}}$)

$$\frac{\partial \varrho_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{M}} = [\![G_{\mathrm{p}}]\!] + \mathcal{S}_{\mathrm{M}}$$

- Mass-average disperse-phase velocity

$$\mathbf{U}_{\mathrm{M}} := \frac{1}{\varrho_{\mathrm{p}}} \int \xi_{\mathrm{p}2} \mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Mass Transport

- If disperse phase has particles with different volumes and different masses (or material densities), at least two internal coordinates are necessary
- First internal coordinate particle volume and second particle mass ($\xi_{\mathrm{p}2} := M_{\mathrm{p}} = \rho_{\mathrm{p}} V_{\mathrm{p}}$; ρ_{p} is material density)
- Disperse-phase mass density is defined by

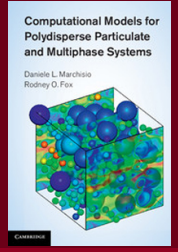
$$\varrho_{\mathrm{p}} := \int \xi_{\mathrm{p}2} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2} = M_{\mathrm{p}}$)

$$\frac{\partial \varrho_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{M}} = [\![G_{\mathrm{p}}]\!] + \mathcal{S}_{\mathrm{M}}$$

- Mass-average disperse-phase velocity

$$\mathbf{U}_{\mathrm{M}} := \frac{1}{\varrho_{\mathrm{p}}} \int \xi_{\mathrm{p}2} \mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Mass Transport

- Since total mass of fluid–particle system is conserved, we can define fluid-phase mass density ϱ_f by

$$\frac{\partial \varrho_\mathrm{f}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} = -[\![G_\mathrm{p}]\!] - \mathcal{S}_\mathrm{M}$$

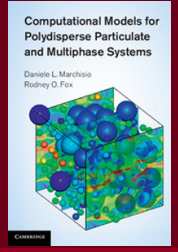
- ϱ_p and ϱ_f are not actual material densities ρ_p and ρ_f, they are equal to mass of solid (fluid) per unit volume of fluid–particle system
- Similarly we can define mixture mass density $\varrho_\mathrm{mix} := \varrho_\mathrm{p} + \varrho_\mathrm{f}$, that satisfies mixture continuity equation:

$$\frac{\partial \varrho_\mathrm{mix}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{mix} \mathbf{U}_\mathrm{mix} = 0$$

- Mass-average mixture velocity is defined by

$$\mathbf{U}_\mathrm{mix} := \frac{1}{\varrho_\mathrm{mix}} \left(\varrho_\mathrm{p} \mathbf{U}_\mathrm{p} + \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} \right)$$

- Transport equations for ϱ_p and ϱ_f are used in two-fluid models and follow directly from GPBE



Disperse-Phase Mass Transport

- Since total mass of fluid–particle system is conserved, we can define fluid-phase mass density ϱ_f by

$$\frac{\partial \varrho_\mathrm{f}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} = -[\![G_\mathrm{p}]\!] - \mathcal{S}_\mathrm{M}$$

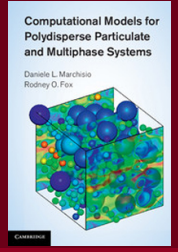
- ϱ_p and ϱ_f are not actual material densities ρ_p and ρ_f, they are equal to mass of solid (fluid) per unit volume of fluid–particle system
- Similarly we can define mixture mass density $\varrho_\mathrm{mix} := \varrho_\mathrm{p} + \varrho_\mathrm{f}$, that satisfies mixture continuity equation:

$$\frac{\partial \varrho_\mathrm{mix}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{mix} \mathbf{U}_\mathrm{mix} = 0$$

- Mass-average mixture velocity is defined by

$$\mathbf{U}_\mathrm{mix} := \frac{1}{\varrho_\mathrm{mix}} \left(\varrho_\mathrm{p} \mathbf{U}_\mathrm{p} + \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} \right)$$

- Transport equations for ϱ_p and ϱ_f are used in two-fluid models and follow directly from GPBE



Disperse-Phase Mass Transport

- Since total mass of fluid–particle system is conserved, we can define fluid-phase mass density ϱ_f by

$$\frac{\partial \varrho_\mathrm{f}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} = -[\![G_\mathrm{p}]\!] - \mathcal{S}_\mathrm{M}$$

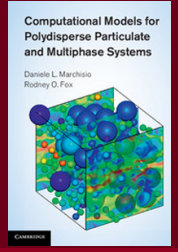
- ϱ_p and ϱ_f are not actual material densities ρ_p and ρ_f, they are equal to mass of solid (fluid) per unit volume of fluid–particle system
- Similarly we can define mixture mass density $\varrho_\mathrm{mix} := \varrho_\mathrm{p} + \varrho_\mathrm{f}$, that satisfies mixture continuity equation:

$$\frac{\partial \varrho_\mathrm{mix}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{mix} \mathbf{U}_\mathrm{mix} = 0$$

- Mass-average mixture velocity is defined by

$$\mathbf{U}_\mathrm{mix} := \frac{1}{\varrho_\mathrm{mix}} \left(\varrho_\mathrm{p} \mathbf{U}_\mathrm{p} + \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} \right)$$

- Transport equations for ϱ_p and ϱ_f are used in two-fluid models and follow directly from GPBE



Disperse-Phase Mass Transport

- Since total mass of fluid–particle system is conserved, we can define fluid-phase mass density ϱ_f by

$$\frac{\partial \varrho_\mathrm{f}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} = -[\![G_\mathrm{p}]\!] - \mathcal{S}_\mathrm{M}$$

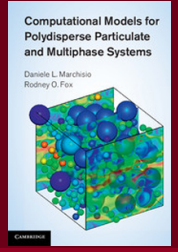
- ϱ_p and ϱ_f are not actual material densities ρ_p and ρ_f, they are equal to mass of solid (fluid) per unit volume of fluid–particle system
- Similarly we can define mixture mass density $\varrho_\mathrm{mix} := \varrho_\mathrm{p} + \varrho_\mathrm{f}$, that satisfies mixture continuity equation:

$$\frac{\partial \varrho_\mathrm{mix}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{mix} \mathbf{U}_\mathrm{mix} = 0$$

- Mass-average mixture velocity is defined by

$$\mathbf{U}_\mathrm{mix} := \frac{1}{\varrho_\mathrm{mix}} \left(\varrho_\mathrm{p} \mathbf{U}_\mathrm{p} + \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} \right)$$

- Transport equations for ϱ_p and ϱ_f are used in two-fluid models and follow directly from GPBE



Disperse-Phase Mass Transport

- Since total mass of fluid–particle system is conserved, we can define fluid-phase mass density ϱ_f by

$$\frac{\partial \varrho_\mathrm{f}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} = -[\![G_\mathrm{p}]\!] - \mathcal{S}_\mathrm{M}$$

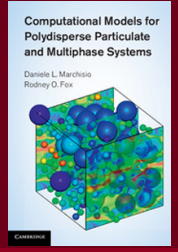
- ϱ_p and ϱ_f are not actual material densities ρ_p and ρ_f, they are equal to mass of solid (fluid) per unit volume of fluid–particle system
- Similarly we can define mixture mass density $\varrho_\mathrm{mix} := \varrho_\mathrm{p} + \varrho_\mathrm{f}$, that satisfies mixture continuity equation:

$$\frac{\partial \varrho_\mathrm{mix}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{mix} \mathbf{U}_\mathrm{mix} = 0$$

- Mass-average mixture velocity is defined by

$$\mathbf{U}_\mathrm{mix} := \frac{1}{\varrho_\mathrm{mix}} \left(\varrho_\mathrm{p} \mathbf{U}_\mathrm{p} + \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} \right)$$

- Transport equations for ϱ_p and ϱ_f are used in two-fluid models and follow directly from GPBE



Disperse-Phase Mass Transport

- Since total mass of fluid–particle system is conserved, we can define fluid-phase mass density ϱ_f by

$$\frac{\partial \varrho_\mathrm{f}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} = -[\![G_\mathrm{p}]\!] - \mathcal{S}_\mathrm{M}$$

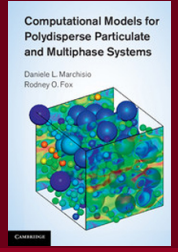
- ϱ_p and ϱ_f are not actual material densities ρ_p and ρ_f, they are equal to mass of solid (fluid) per unit volume of fluid–particle system
- Similarly we can define mixture mass density $\varrho_\mathrm{mix} := \varrho_\mathrm{p} + \varrho_\mathrm{f}$, that satisfies mixture continuity equation:

$$\frac{\partial \varrho_\mathrm{mix}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{mix} \mathbf{U}_\mathrm{mix} = 0$$

- Mass-average mixture velocity is defined by

$$\mathbf{U}_\mathrm{mix} := \frac{1}{\varrho_\mathrm{mix}} \left(\varrho_\mathrm{p} \mathbf{U}_\mathrm{p} + \varrho_\mathrm{f} \mathbf{U}_\mathrm{f} \right)$$

- Transport equations for ϱ_p and ϱ_f are used in two-fluid models and follow directly from GPBE



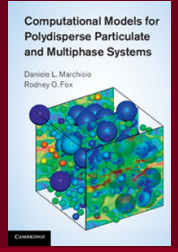
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- In case where $\xi_{\mathrm{p}1} = V_{\mathrm{p}}$ and $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ are constant and equal for all particles (monodisperse): $\varrho_{\mathrm{p}} = M_{\mathrm{p}} N = \rho_{\mathrm{p}} \alpha_{\mathrm{p}}$
- Average particle velocities are all equal: $\mathbf{U}_{\mathrm{p}} := \mathbf{U}_{\mathrm{M}} = \mathbf{U}_{\mathrm{V}} = \mathbf{U}_{\mathrm{N}}$
- Governing equations become (incompressible fluid):

$$\frac{\partial \alpha_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} = 0 \qquad \frac{\partial \alpha_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}} = 0$$

- Sum of two continuity equations yields realizability constraint for velocity fields: $\nabla_{\mathbf{x}} \cdot \mathbf{U}_{\mathrm{vol}} = 0$, $\mathbf{U}_{\mathrm{vol}} := \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} + \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}$, that must be incorporated into conditional source terms in fluid-phase momentum transport equation (fluid-phase pressure)
- Note that, in general, $\mathbf{U}_{\mathrm{vol}} \neq \mathbf{U}_{\mathrm{mix}}$ unless fluid and particles have same material densities



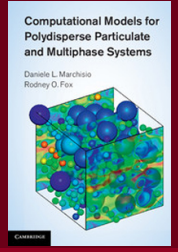
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- In case where $\xi_{\mathrm{p}1} = V_{\mathrm{p}}$ and $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ are constant and equal for all particles (monodisperse): $\varrho_{\mathrm{p}} = M_{\mathrm{p}} N = \rho_{\mathrm{p}} \alpha_{\mathrm{p}}$
- Average particle velocities are all equal: $\mathbf{U}_{\mathrm{p}} := \mathbf{U}_{\mathrm{M}} = \mathbf{U}_{\mathrm{V}} = \mathbf{U}_{\mathrm{N}}$
- Governing equations become (incompressible fluid):

$$\frac{\partial \alpha_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} = 0 \qquad \frac{\partial \alpha_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}} = 0$$

- Sum of two continuity equations yields realizability constraint for velocity fields: $\nabla_{\mathbf{x}} \cdot \mathbf{U}_{\mathrm{vol}} = 0$, $\mathbf{U}_{\mathrm{vol}} := \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} + \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}$, that must be incorporated into conditional source terms in fluid-phase momentum transport equation (fluid-phase pressure)
- Note that, in general, $\mathbf{U}_{\mathrm{vol}} \neq \mathbf{U}_{\mathrm{mix}}$ unless fluid and particles have same material densities



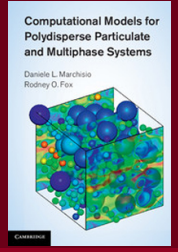
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- In case where $\xi_{\mathrm{p}1} = V_{\mathrm{p}}$ and $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ are constant and equal for all particles (monodisperse): $\varrho_{\mathrm{p}} = M_{\mathrm{p}} N = \rho_{\mathrm{p}} \alpha_{\mathrm{p}}$
- Average particle velocities are all equal: $\mathbf{U}_{\mathrm{p}} := \mathbf{U}_{\mathrm{M}} = \mathbf{U}_{\mathrm{V}} = \mathbf{U}_{\mathrm{N}}$
- Governing equations become (incompressible fluid):

$$\frac{\partial \alpha_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} = 0 \qquad \frac{\partial \alpha_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}} = 0$$

- Sum of two continuity equations yields realizability constraint for velocity fields: $\nabla_{\mathbf{x}} \cdot \mathbf{U}_{\mathrm{vol}} = 0$, $\mathbf{U}_{\mathrm{vol}} := \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} + \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}$, that must be incorporated into conditional source terms in fluid-phase momentum transport equation (fluid-phase pressure)
- Note that, in general, $\mathbf{U}_{\mathrm{vol}} \neq \mathbf{U}_{\mathrm{mix}}$ unless fluid and particles have same material densities



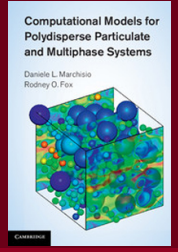
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- In case where $\xi_{\mathrm{p}1} = V_{\mathrm{p}}$ and $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ are constant and equal for all particles (monodisperse): $\varrho_{\mathrm{p}} = M_{\mathrm{p}} N = \rho_{\mathrm{p}} \alpha_{\mathrm{p}}$
- Average particle velocities are all equal: $\mathbf{U}_{\mathrm{p}} := \mathbf{U}_{\mathrm{M}} = \mathbf{U}_{\mathrm{V}} = \mathbf{U}_{\mathrm{N}}$
- Governing equations become (incompressible fluid):

$$\frac{\partial \alpha_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} = 0 \qquad \frac{\partial \alpha_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}} = 0$$

- Sum of two continuity equations yields realizability constraint for velocity fields: $\nabla_{\mathbf{x}} \cdot \mathbf{U}_{\mathrm{vol}} = 0$, $\mathbf{U}_{\mathrm{vol}} := \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} + \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}$, that must be incorporated into conditional source terms in fluid-phase momentum transport equation (fluid-phase pressure)
- Note that, in general, $\mathbf{U}_{\mathrm{vol}} \neq \mathbf{U}_{\mathrm{mix}}$ unless fluid and particles have same material densities



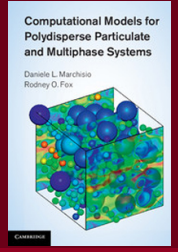
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- In case where $\xi_{\mathrm{p}1} = V_{\mathrm{p}}$ and $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ are constant and equal for all particles (monodisperse): $\varrho_{\mathrm{p}} = M_{\mathrm{p}} N = \rho_{\mathrm{p}} \alpha_{\mathrm{p}}$
- Average particle velocities are all equal: $\mathbf{U}_{\mathrm{p}} := \mathbf{U}_{\mathrm{M}} = \mathbf{U}_{\mathrm{V}} = \mathbf{U}_{\mathrm{N}}$
- Governing equations become (incompressible fluid):

$$\frac{\partial \alpha_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} = 0 \qquad \frac{\partial \alpha_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}} = 0$$

- Sum of two continuity equations yields realizability constraint for velocity fields: $\boldsymbol{\nabla}_{\mathbf{x}} \cdot \mathbf{U}_{\mathrm{vol}} = 0$, $\mathbf{U}_{\mathrm{vol}} := \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} + \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}$, that must be incorporated into conditional source terms in fluid-phase momentum transport equation (fluid-phase pressure)
- Note that, in general, $\mathbf{U}_{\mathrm{vol}} \neq \mathbf{U}_{\mathrm{mix}}$ unless fluid and particles have same material densities



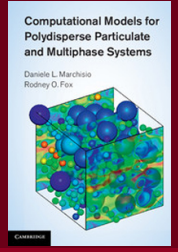
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- In case where $\xi_{\mathrm{p}1} = V_{\mathrm{p}}$ and $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ are constant and equal for all particles (monodisperse): $\varrho_{\mathrm{p}} = M_{\mathrm{p}} N = \rho_{\mathrm{p}} \alpha_{\mathrm{p}}$
- Average particle velocities are all equal: $\mathbf{U}_{\mathrm{p}} := \mathbf{U}_{\mathrm{M}} = \mathbf{U}_{\mathrm{V}} = \mathbf{U}_{\mathrm{N}}$
- Governing equations become (incompressible fluid):

$$\frac{\partial \alpha_{\mathrm{p}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} = 0 \qquad \frac{\partial \alpha_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}} = 0$$

- Sum of two continuity equations yields realizability constraint for velocity fields: $\boldsymbol{\nabla}_{\mathbf{x}} \cdot \mathbf{U}_{\mathrm{vol}} = 0$, $\mathbf{U}_{\mathrm{vol}} := \alpha_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} + \alpha_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}$, that must be incorporated into conditional source terms in fluid-phase momentum transport equation (fluid-phase pressure)
- Note that, in general, $\mathbf{U}_{\mathrm{vol}} \neq \mathbf{U}_{\mathrm{mix}}$ unless fluid and particles have same material densities



Disperse-Phase Momentum Transport

- Disperse-phase momentum density is defined by

$$\varrho_{\mathrm{p}}\mathbf{U}_{\mathrm{M}} := \int \xi_{\mathrm{p}2}\mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

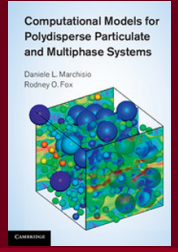
  where $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ corresponds to particle mass

- As usual its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2}\mathbf{v}_{\mathrm{p}}$):

$$\frac{\partial \varrho_{\mathrm{p}}\mathbf{U}_{\mathrm{M}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}}\mathbf{F}_{\mathrm{p}} = \varrho_{\mathrm{p}}[\![\mathbf{A}_{\mathrm{fp}}]\!]_{\mathrm{p}} + \varrho_{\mathrm{p}}[\![\mathbf{A}_{\mathrm{p}}]\!]_{\mathrm{p}} + [\![\mathbf{G}_{\mathrm{p}}]\!]_{\mathrm{p}} + \varrho_{\mathrm{p}}[\![\mathbf{S}]\!]_{\mathrm{p}}$$

  where (unclosed) momentum-convection term (second-order tensor) is defined by

$$\varrho_{\mathrm{p}}\mathbf{F}_{\mathrm{p}} := \int \xi_{\mathrm{p}2}(\mathbf{v}_{\mathrm{p}} \otimes \mathbf{v}_{\mathrm{p}}) n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Momentum Transport

- Disperse-phase momentum density is defined by

$$\varrho_{\mathrm{p}}\mathbf{U}_{\mathrm{M}} := \int \xi_{\mathrm{p}2}\mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

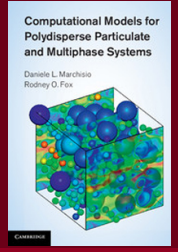
  where $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ corresponds to particle mass

- As usual its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2}\mathbf{v}_{\mathrm{p}}$):

$$\frac{\partial \varrho_{\mathrm{p}}\mathbf{U}_{\mathrm{M}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}}\mathbf{F}_{\mathrm{p}} = \varrho_{\mathrm{p}}[\![\mathbf{A}_{\mathrm{fp}}]\!]_{\mathrm{p}} + \varrho_{\mathrm{p}}[\![\mathbf{A}_{\mathrm{p}}]\!]_{\mathrm{p}} + [\![\mathbf{G}_{\mathrm{p}}]\!]_{\mathrm{p}} + \varrho_{\mathrm{p}}[\![\mathbf{S}]\!]_{\mathrm{p}}$$

  where (unclosed) momentum-convection term (second-order tensor) is defined by

$$\varrho_{\mathrm{p}}\mathbf{F}_{\mathrm{p}} := \int \xi_{\mathrm{p}2}(\mathbf{v}_{\mathrm{p}} \otimes \mathbf{v}_{\mathrm{p}}) n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Momentum Transport

- Disperse-phase momentum density is defined by

$$\varrho_{\mathrm{p}}\mathbf{U}_{\mathrm{M}} := \int \xi_{\mathrm{p}2}\mathbf{v}_{\mathrm{p}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

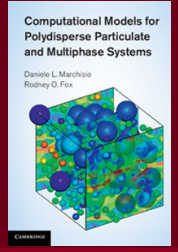
  where $\xi_{\mathrm{p}2} = M_{\mathrm{p}}$ corresponds to particle mass

- As usual its transport equation is derived from GPBE ($g = \xi_{\mathrm{p}2}\mathbf{v}_{\mathrm{p}}$):

$$\frac{\partial \varrho_{\mathrm{p}}\mathbf{U}_{\mathrm{M}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{p}}\mathbf{F}_{\mathrm{p}} = \varrho_{\mathrm{p}}[\![\mathbf{A}_{\mathrm{fp}}]\!]_{\mathrm{p}} + \varrho_{\mathrm{p}}[\![\mathbf{A}_{\mathrm{p}}]\!]_{\mathrm{p}} + [\![\mathbf{G}_{\mathrm{p}}]\!]_{\mathrm{p}} + \varrho_{\mathrm{p}}[\![\mathbf{S}]\!]_{\mathrm{p}}$$

  where (unclosed) momentum-convection term (second-order tensor) is defined by

$$\varrho_{\mathrm{p}}\mathbf{F}_{\mathrm{p}} := \int \xi_{\mathrm{p}2}(\mathbf{v}_{\mathrm{p}} \otimes \mathbf{v}_{\mathrm{p}}) n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



Disperse-Phase Momentum Transport

- Surface forces on particles due to surrounding fluid (e.g., drag, lift, and pressure forces)

$$\varrho_{\mathrm{p}} [\![\mathbf{A}_{\mathrm{fp}}]\!]_{\mathrm{p}} := \int \xi_{\mathrm{p}2} \langle \mathbf{A}_{\mathrm{fp}2} \rangle_1 n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Body forces (such as gravity):

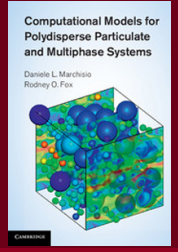
$$\varrho_{\mathrm{p}} [\![\mathbf{A}_{\mathrm{p}}]\!]_{\mathrm{p}} := \int \xi_{\mathrm{p}2} \langle \mathbf{A}_{\mathrm{p}2} \rangle_1 n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Momentum added to particle phase due to mass transfer from liquid

$$[\![\mathbf{G}_{\mathrm{p}}]\!]_{\mathrm{p}} := \int \mathbf{v}_{\mathrm{p}} \langle G_{\mathrm{p}2} \rangle_1 n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

- Discontinuous changes in particle momentum due to collisions and particle nucleation:

$$\varrho_{\mathrm{p}} [\![\mathbf{S}]\!]_{\mathrm{p}} := \int \xi_{\mathrm{p}2} \mathbf{v}_{\mathrm{p}} \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$



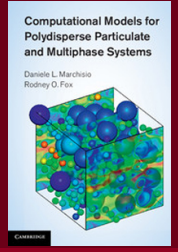
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- For monodisperse particles, disperse-phase mass density is $\varrho_\mathrm{p} = \rho_\mathrm{p}\alpha_\mathrm{p}$, mass average can be replaced by number average, and transport equation for disperse-phase momentum density is

$$\frac{\partial \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{fp}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{p}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{N}$$

- Disperse-phase pressure tensor, $[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \mathbf{F}_\mathrm{N} - \mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p}$, is central second-order moment
- Term for collisions can also generate separate spatial flux term (collisional flux) which is added to disperse-phase pressure tensor



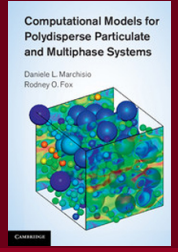
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- For monodisperse particles, disperse-phase mass density is $\varrho_\mathrm{p} = \rho_\mathrm{p}\alpha_\mathrm{p}$, mass average can be replaced by number average, and transport equation for disperse-phase momentum density is

$$\frac{\partial \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{fp}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{p}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{N}$$

- Disperse-phase pressure tensor, $[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \mathbf{F}_\mathrm{N} - \mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p}$, is central second-order moment
- Term for collisions can also generate separate spatial flux term (collisional flux) which is added to disperse-phase pressure tensor



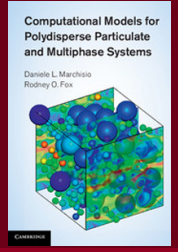
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- For monodisperse particles, disperse-phase mass density is $\varrho_\mathrm{p} = \rho_\mathrm{p}\alpha_\mathrm{p}$, mass average can be replaced by number average, and transport equation for disperse-phase momentum density is

$$\frac{\partial \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{fp}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{p}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{N}$$

- Disperse-phase pressure tensor, $[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \mathbf{F}_\mathrm{N} - \mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p}$, is central second-order moment
- Term for collisions can also generate separate spatial flux term (collisional flux) which is added to disperse-phase pressure tensor



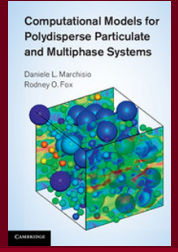
Towards the Two-Fluid Model

Constant (and equal) volume and mass

- For monodisperse particles, disperse-phase mass density is $\varrho_\mathrm{p} = \rho_\mathrm{p}\alpha_\mathrm{p}$, mass average can be replaced by number average, and transport equation for disperse-phase momentum density is

$$\frac{\partial \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}\mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p} + \frac{\partial}{\partial \mathbf{x}} \cdot \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{fp}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}[\![\mathbf{A}_\mathrm{p}]\!]_\mathrm{N} + \rho_\mathrm{p}\alpha_\mathrm{p}\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{N}$$

- Disperse-phase pressure tensor, $[\![\mathbf{u}_\mathrm{p}\mathbf{u}_\mathrm{p}]\!]_\mathrm{N} = \mathbf{F}_\mathrm{N} - \mathbf{U}_\mathrm{p} \otimes \mathbf{U}_\mathrm{p}$, is central second-order moment
- Term for collisions can also generate separate spatial flux term (collisional flux) which is added to disperse-phase pressure tensor



Towards the Two-Fluid Model

Constant (and equal) volume and mass

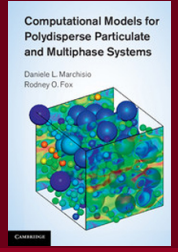
- From GPBE we can also derive a transport equation for fluid-phase momentum density (after applying constraints)

$$\frac{\partial \varrho_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{f}} [\![\mathbf{U}_{\mathrm{f}} \otimes \mathbf{U}_{\mathrm{f}}]\!]_{\mathrm{f}} = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_{\mathrm{f}} + \varrho_{\mathrm{f}} \mathbf{A}_{\mathrm{f}} - \varrho_{\mathrm{p}} [\![\mathbf{A}_{\mathrm{fp}}]\!]_{\mathrm{p}} - [\![\mathbf{G}_{\mathrm{p}}]\!]_{\mathrm{p}} - \varrho_{\mathrm{p}} [\![\mathbf{S}]\!]_{\mathrm{p}}$$

- $\mathbf{S}_{\mathrm{f}}$ and $\mathbf{A}_{\mathrm{f}}$ denote the viscous/pressure stress tensor and body forces acting on fluid phase
- Convection term

$$[\![\mathbf{U}_{\mathrm{f}} \otimes \mathbf{U}_{\mathrm{f}}]\!]_{\mathrm{f}} := \int \xi_{\mathrm{f}1} \mathbf{v}_{\mathrm{f}} \otimes \mathbf{v}_{\mathrm{f}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

requires separate model for microscale (pseudo) turbulence



Towards the Two-Fluid Model

Constant (and equal) volume and mass

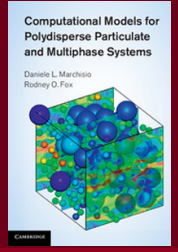
- From GPBE we can also derive a transport equation for fluid-phase momentum density (after applying constraints)

$$\frac{\partial \varrho_\mathrm{f} \mathbf{U}_\mathrm{f}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{f} [\![\mathbf{U}_\mathrm{f} \otimes \mathbf{U}_\mathrm{f}]\!]_\mathrm{f} = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_\mathrm{f} + \varrho_\mathrm{f} \mathbf{A}_\mathrm{f} - \varrho_\mathrm{p} [\![\mathbf{A}_\mathrm{fp}]\!]_\mathrm{p} - [\![\mathbf{G}_\mathrm{p}]\!]_\mathrm{p} - \varrho_\mathrm{p} [\![\mathbf{S}]\!]_\mathrm{p}$$

- $\mathbf{S}_\mathrm{f}$ and $\mathbf{A}_\mathrm{f}$ denote the viscous/pressure stress tensor and body forces acting on fluid phase
- Convection term

$$[\![\mathbf{U}_\mathrm{f} \otimes \mathbf{U}_\mathrm{f}]\!]_\mathrm{f} := \int \xi_{\mathrm{f}1} \mathbf{v}_\mathrm{f} \otimes \mathbf{v}_\mathrm{f} n \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\xi_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\xi_\mathrm{f}$$

requires separate model for microscale (pseudo) turbulence



Towards the Two-Fluid Model

Constant (and equal) volume and mass

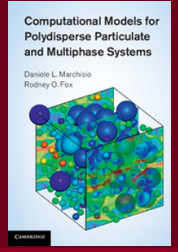
- From GPBE we can also derive a transport equation for fluid-phase momentum density (after applying constraints)

$$\frac{\partial \varrho_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{f}} [\![\mathbf{U}_{\mathrm{f}} \otimes \mathbf{U}_{\mathrm{f}}]\!]_{\mathrm{f}} = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_{\mathrm{f}} + \varrho_{\mathrm{f}} \mathbf{A}_{\mathrm{f}} - \varrho_{\mathrm{p}} [\![\mathbf{A}_{\mathrm{fp}}]\!]_{\mathrm{p}} - [\![\mathbf{G}_{\mathrm{p}}]\!]_{\mathrm{p}} - \varrho_{\mathrm{p}} [\![\mathbf{S}]\!]_{\mathrm{p}}$$

- $\mathbf{S}_{\mathrm{f}}$ and $\mathbf{A}_{\mathrm{f}}$ denote the viscous/pressure stress tensor and body forces acting on fluid phase
- Convection term

$$[\![\mathbf{U}_{\mathrm{f}} \otimes \mathbf{U}_{\mathrm{f}}]\!]_{\mathrm{f}} := \int \xi_{\mathrm{f}1} \mathbf{v}_{\mathrm{f}} \otimes \mathbf{v}_{\mathrm{f}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\xi_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\xi_{\mathrm{f}}$$

requires separate model for microscale (pseudo) turbulence



Towards the Two-Fluid Model

Constant (and equal) volume and mass

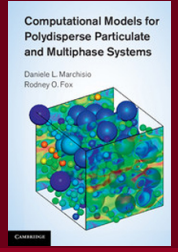
- From GPBE we can also derive a transport equation for fluid-phase momentum density (after applying constraints)

$$\frac{\partial \varrho_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\mathrm{f}} [\![\mathbf{U}_{\mathrm{f}} \otimes \mathbf{U}_{\mathrm{f}}]\!]_{\mathrm{f}} = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_{\mathrm{f}} + \varrho_{\mathrm{f}} \mathbf{A}_{\mathrm{f}} - \varrho_{\mathrm{p}} [\![\mathbf{A}_{\mathrm{fp}}]\!]_{\mathrm{p}} - [\![\mathbf{G}_{\mathrm{p}}]\!]_{\mathrm{p}} - \varrho_{\mathrm{p}} [\![\mathbf{S}]\!]_{\mathrm{p}}$$

- $\mathbf{S}_{\mathrm{f}}$ and $\mathbf{A}_{\mathrm{f}}$ denote the viscous/pressure stress tensor and body forces acting on fluid phase
- Convection term

$$[\![\mathbf{U}_{\mathrm{f}} \otimes \mathbf{U}_{\mathrm{f}}]\!]_{\mathrm{f}} := \int \xi_{\mathrm{f}1} \mathbf{v}_{\mathrm{f}} \otimes \mathbf{v}_{\mathrm{f}} n \, \mathrm{d}\mathbf{v}_{\mathrm{p}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{p}} \, \mathrm{d}\mathbf{v}_{\mathrm{f}} \, \mathrm{d}\boldsymbol{\xi}_{\mathrm{f}}$$

requires separate model for microscale (pseudo) turbulence



Mixture Model

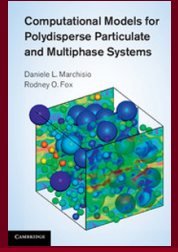
- Recalling definition of mixture mass density, $\varrho_{\text{mix}} \equiv \varrho_{\text{p}} + \varrho_{\text{f}}$, whose transport equation is:

$$\frac{\partial \varrho_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\text{mix}} \mathbf{U}_{\text{mix}} = 0$$

  and $\mathbf{U}_{\text{mix}} := \frac{1}{\varrho_{\text{mix}}} \left(\varrho_{\text{p}} \mathbf{U}_{\text{p}} + \varrho_{\text{f}} \mathbf{U}_{\text{f}} \right)$ is mass-average mixture velocity
- We can derive (by summing up equations for $\mathbf{U}_{\text{p}}$ and $\mathbf{U}_{\text{f}}$):

$$\frac{\partial \varrho_{\text{mix}} \mathbf{U}_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \left(\varrho_{\text{p}} \mathbf{F}_{\text{p}} + \varrho_{\text{f}} [\![\mathbf{U}_{\text{f}} \otimes \mathbf{U}_{\text{f}}]\!]_{\text{f}} \right) = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_{\text{f}} + \varrho_{\text{p}} [\![\mathbf{A}_{\text{p}}]\!]_{\text{p}} + \varrho_{\text{f}} \mathbf{A}_{\text{f}}$$

- This expression does not contain terms for momentum transfer between phases but can be solved only when $\mathbf{U}_{\text{mix}}$ is very close to $\mathbf{U}_{\text{f}}$ (i.e., very small Stokes numbers)



Mixture Model

- Recalling definition of mixture mass density, $\varrho_{\text{mix}} \equiv \varrho_{\text{p}} + \varrho_{\text{f}}$, whose transport equation is:

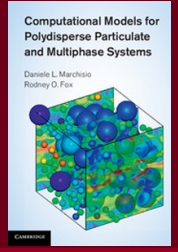
$$\frac{\partial \varrho_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\text{mix}} \mathbf{U}_{\text{mix}} = 0$$

  and $\mathbf{U}_{\text{mix}} := \frac{1}{\varrho_{\text{mix}}} \left(\varrho_{\text{p}} \mathbf{U}_{\text{p}} + \varrho_{\text{f}} \mathbf{U}_{\text{f}} \right)$ is mass-average mixture velocity

- We can derive (by summing up equations for $\mathbf{U}_{\text{p}}$ and $\mathbf{U}_{\text{f}}$):

$$\frac{\partial \varrho_{\text{mix}} \mathbf{U}_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \left(\varrho_{\text{p}} \mathbf{F}_{\text{p}} + \varrho_{\text{f}} [\![\mathbf{U}_{\text{f}} \otimes \mathbf{U}_{\text{f}}]\!]_{\text{f}} \right) = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_{\text{f}} + \varrho_{\text{p}} [\![\mathbf{A}_{\text{p}}]\!]_{\text{p}} + \varrho_{\text{f}} \mathbf{A}_{\text{f}}$$

- This expression does not contain terms for momentum transfer between phases but can be solved only when $\mathbf{U}_{\text{mix}}$ is very close to $\mathbf{U}_{\text{f}}$ (i.e., very small Stokes numbers)



Mixture Model

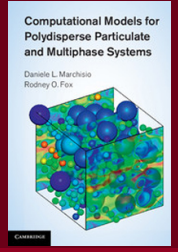
- Recalling definition of mixture mass density, $\varrho_{\text{mix}} \equiv \varrho_{\text{p}} + \varrho_{\text{f}}$, whose transport equation is:

$$\frac{\partial \varrho_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\text{mix}} \mathbf{U}_{\text{mix}} = 0$$

  and $\mathbf{U}_{\text{mix}} := \frac{1}{\varrho_{\text{mix}}} \left(\varrho_{\text{p}} \mathbf{U}_{\text{p}} + \varrho_{\text{f}} \mathbf{U}_{\text{f}} \right)$ is mass-average mixture velocity
- We can derive (by summing up equations for $\mathbf{U}_{\text{p}}$ and $\mathbf{U}_{\text{f}}$):

$$\frac{\partial \varrho_{\text{mix}} \mathbf{U}_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \left(\varrho_{\text{p}} \mathbf{F}_{\text{p}} + \varrho_{\text{f}} [\![\mathbf{U}_{\text{f}} \otimes \mathbf{U}_{\text{f}}]\!]_{\text{f}} \right) = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_{\text{f}} + \varrho_{\text{p}} [\![\mathbf{A}_{\text{p}}]\!]_{\text{p}} + \varrho_{\text{f}} \mathbf{A}_{\text{f}}$$

- This expression does not contain terms for momentum transfer between phases but can be solved only when $\mathbf{U}_{\text{mix}}$ is very close to $\mathbf{U}_{\text{f}}$ (i.e., very small Stokes numbers)



Mixture Model

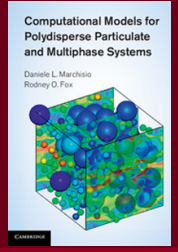
- Recalling definition of mixture mass density, $\varrho_{\text{mix}} \equiv \varrho_{\text{p}} + \varrho_{\text{f}}$, whose transport equation is:

$$\frac{\partial \varrho_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_{\text{mix}} \mathbf{U}_{\text{mix}} = 0$$

  and $\mathbf{U}_{\text{mix}} := \frac{1}{\varrho_{\text{mix}}} \left(\varrho_{\text{p}} \mathbf{U}_{\text{p}} + \varrho_{\text{f}} \mathbf{U}_{\text{f}} \right)$ is mass-average mixture velocity
- We can derive (by summing up equations for $\mathbf{U}_{\text{p}}$ and $\mathbf{U}_{\text{f}}$):

$$\frac{\partial \varrho_{\text{mix}} \mathbf{U}_{\text{mix}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \left(\varrho_{\text{p}} \mathbf{F}_{\text{p}} + \varrho_{\text{f}} [\![\mathbf{U}_{\text{f}} \otimes \mathbf{U}_{\text{f}}]\!]_{\text{f}} \right) = \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{S}_{\text{f}} + \varrho_{\text{p}} [\![\mathbf{A}_{\text{p}}]\!]_{\text{p}} + \varrho_{\text{f}} \mathbf{A}_{\text{f}}$$

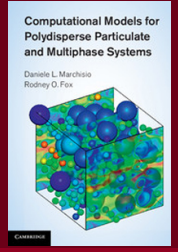
- This expression does not contain terms for momentum transfer between phases but can be solved only when $\mathbf{U}_{\text{mix}}$ is very close to $\mathbf{U}_{\text{f}}$ (i.e., very small Stokes numbers)



Moment Transport Equations

Summary

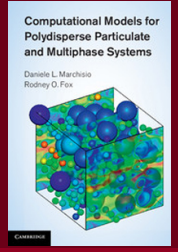
- Through ensemble average and by using some simplifying hypotheses NDF for multiphase system has been defined
- NDF completely defines multiphase system
- Moments of NDF correspond to relevant measurable properties: disperse-phase number concentration, disperse- and fluid-phase volume fractions and mass densities, mixture mass density, number-, volume- and mass-average velocities for disperse and fluid phases and mass-average mixture velocity
- Evolution of NDF is dictated by GPBE and from it, transport equations for moments can be derived simply by applying moment transform
- Two-fluid model and mixture model are directly derivable from moment transport equations with additional hypothesis of constant and equal particle mass and volume



Moment Transport Equations

Summary

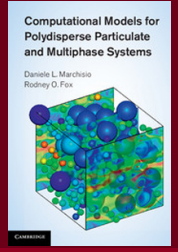
- Through ensemble average and by using some simplifying hypotheses NDF for multiphase system has been defined
- NDF completely defines multiphase system
- Moments of NDF correspond to relevant measurable properties: disperse-phase number concentration, disperse- and fluid-phase volume fractions and mass densities, mixture mass density, number-, volume- and mass-average velocities for disperse and fluid phases and mass-average mixture velocity
- Evolution of NDF is dictated by GPBE and from it, transport equations for moments can be derived simply by applying moment transform
- Two-fluid model and mixture model are directly derivable from moment transport equations with additional hypothesis of constant and equal particle mass and volume



Moment Transport Equations

Summary

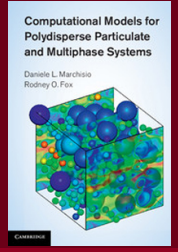
- Through ensemble average and by using some simplifying hypotheses NDF for multiphase system has been defined
- NDF completely defines multiphase system
- Moments of NDF correspond to relevant measurable properties: disperse-phase number concentration, disperse- and fluid-phase volume fractions and mass densities, mixture mass density, number-, volume- and mass-average velocities for disperse and fluid phases and mass-average mixture velocity
- Evolution of NDF is dictated by GPBE and from it, transport equations for moments can be derived simply by applying moment transform
- Two-fluid model and mixture model are directly derivable from moment transport equations with additional hypothesis of constant and equal particle mass and volume



Moment Transport Equations

Summary

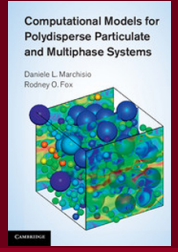
- Through ensemble average and by using some simplifying hypotheses NDF for multiphase system has been defined
- NDF completely defines multiphase system
- Moments of NDF correspond to relevant measurable properties: disperse-phase number concentration, disperse- and fluid-phase volume fractions and mass densities, mixture mass density, number-, volume- and mass-average velocities for disperse and fluid phases and mass-average mixture velocity
- Evolution of NDF is dictated by GPBE and from it, transport equations for moments can be derived simply by applying moment transform
- Two-fluid model and mixture model are directly derivable from moment transport equations with additional hypothesis of constant and equal particle mass and volume



Moment Transport Equations

Summary

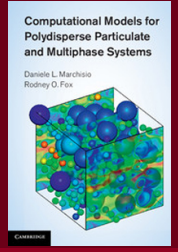
- Through ensemble average and by using some simplifying hypotheses NDF for multiphase system has been defined
- NDF completely defines multiphase system
- Moments of NDF correspond to relevant measurable properties: disperse-phase number concentration, disperse- and fluid-phase volume fractions and mass densities, mixture mass density, number-, volume- and mass-average velocities for disperse and fluid phases and mass-average mixture velocity
- Evolution of NDF is dictated by GPBE and from it, transport equations for moments can be derived simply by applying moment transform
- Two-fluid model and mixture model are directly derivable from moment transport equations with additional hypothesis of constant and equal particle mass and volume



Moment Transport Equations

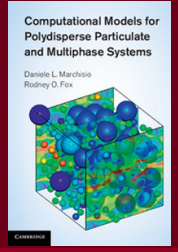
Summary

- Through ensemble average and by using some simplifying hypotheses NDF for multiphase system has been defined
- NDF completely defines multiphase system
- Moments of NDF correspond to relevant measurable properties: disperse-phase number concentration, disperse- and fluid-phase volume fractions and mass densities, mixture mass density, number-, volume- and mass-average velocities for disperse and fluid phases and mass-average mixture velocity
- Evolution of NDF is dictated by GPBE and from it, transport equations for moments can be derived simply by applying moment transform
- Two-fluid model and mixture model are directly derivable from moment transport equations with additional hypothesis of constant and equal particle mass and volume



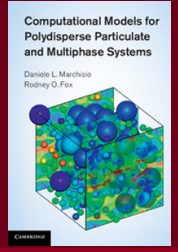
Moment Closure

- In general **moment transport equations are not closed**, as terms appearing in them cannot be written in terms of moments themselves
- In all these cases **NDF has to be reconstructed**
- **Parameterized NDF:** functional form for NDF in terms of a few lower-order moments (e.g. single Dirac delta function, log-normal distribution)
- **Functional expansion for NDF:** similar to previous one but in order to increase number of degrees of freedom in systematic manner functional expansion can be used to represent NDF
- **Discrete NDF:** discrete form for phase-space variables, discretized on grid (e.g. uniform, geometric series, etc.)
- **Quadrature-based moment methods:** reconstruction of NDF using sum of Dirac delta functions located at nodes of quadrature approximation (i.e. zeros of orthogonal polynomials)



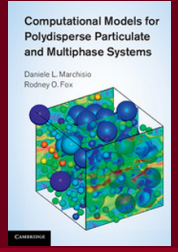
Moment Closure

- In general **moment transport equations are not closed**, as terms appearing in them cannot be written in terms of moments themselves
- In all these cases **NDF has to be reconstructed**
- **Parameterized NDF:** functional form for NDF in terms of a few lower-order moments (e.g. single Dirac delta function, log-normal distribution)
- **Functional expansion for NDF:** similar to previous one but in order to increase number of degrees of freedom in systematic manner functional expansion can be used to represent NDF
- **Discrete NDF:** discrete form for phase-space variables, discretized on grid (e.g. uniform, geometric series, etc.)
- **Quadrature-based moment methods:** reconstruction of NDF using sum of Dirac delta functions located at nodes of quadrature approximation (i.e. zeros of orthogonal polynomials)



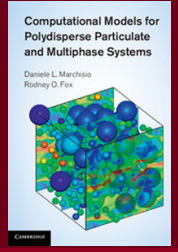
Moment Closure

- In general **moment transport equations are not closed**, as terms appearing in them cannot be written in terms of moments themselves
- In all these cases **NDF has to be reconstructed**
- **Parameterized NDF:** functional form for NDF in terms of a few lower-order moments (e.g. single Dirac delta function, log-normal distribution)
- **Functional expansion for NDF:** similar to previous one but in order to increase number of degrees of freedom in systematic manner functional expansion can be used to represent NDF
- **Discrete NDF:** discrete form for phase-space variables, discretized on grid (e.g. uniform, geometric series, etc.)
- **Quadrature-based moment methods:** reconstruction of NDF using sum of Dirac delta functions located at nodes of quadrature approximation (i.e. zeros of orthogonal polynomials)



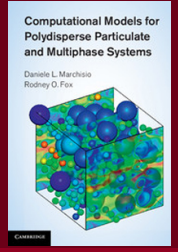
Moment Closure

- In general **moment transport equations are not closed**, as terms appearing in them cannot be written in terms of moments themselves
- In all these cases **NDF has to be reconstructed**
- **Parameterized NDF:** functional form for NDF in terms of a few lower-order moments (e.g. single Dirac delta function, log-normal distribution)
- **Functional expansion for NDF:** similar to previous one but in order to increase number of degrees of freedom in systematic manner functional expansion can be used to represent NDF
- **Discrete NDF:** discrete form for phase-space variables, discretized on grid (e.g. uniform, geometric series, etc.)
- **Quadrature-based moment methods:** reconstruction of NDF using sum of Dirac delta functions located at nodes of quadrature approximation (i.e. zeros of orthogonal polynomials)



Moment Closure

- In general **moment transport equations are not closed**, as terms appearing in them cannot be written in terms of moments themselves
- In all these cases **NDF has to be reconstructed**
- **Parameterized NDF:** functional form for NDF in terms of a few lower-order moments (e.g. single Dirac delta function, log-normal distribution)
- **Functional expansion for NDF:** similar to previous one but in order to increase number of degrees of freedom in systematic manner functional expansion can be used to represent NDF
- **Discrete NDF:** discrete form for phase-space variables, discretized on grid (e.g. uniform, geometric series, etc.)
- **Quadrature-based moment methods:** reconstruction of NDF using sum of Dirac delta functions located at nodes of quadrature approximation (i.e. zeros of orthogonal polynomials)



Real-Space Advection and Integration in CFD Codes

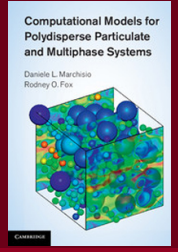
- **Real-space advection** (second term on LHS)

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$

- It represents the movement of particles in real space:

$$\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n = \frac{\partial}{\partial x_1} v_{\mathrm{p}1} n + \frac{\partial}{\partial x_2} v_{\mathrm{p}2} n + \frac{\partial}{\partial x_3} v_{\mathrm{p}3} n$$

- Particle velocity plays an important role since it generates spatial transport: **active variable**
- Other internal coordinates (e.g. particle size) are **passive variables**: carried along with given velocity



Real-Space Advection and Integration in CFD Codes

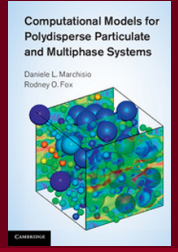
- **Real-space advection** (second term on LHS)

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$

- It represents the movement of particles in real space:

$$\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n = \frac{\partial}{\partial x_1} v_{\mathrm{p}1} n + \frac{\partial}{\partial x_2} v_{\mathrm{p}2} n + \frac{\partial}{\partial x_3} v_{\mathrm{p}3} n$$

- Particle velocity plays an important role since it generates spatial transport: **active variable**
- Other internal coordinates (e.g. particle size) are **passive variables**: carried along with given velocity



Real-Space Advection and Integration in CFD Codes

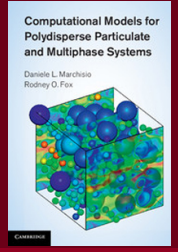
- **Real-space advection** (second term on LHS)

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$

- It represents the movement of particles in real space:

$$\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n = \frac{\partial}{\partial x_1} v_{\mathrm{p}1} n + \frac{\partial}{\partial x_2} v_{\mathrm{p}2} n + \frac{\partial}{\partial x_3} v_{\mathrm{p}3} n$$

- Particle velocity plays an important role since it generates spatial transport: **active variable**
- Other internal coordinates (e.g. particle size) are **passive variables**: carried along with given velocity



Real-Space Advection and Integration in CFD Codes

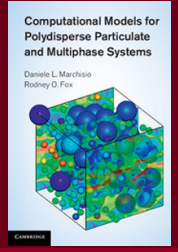
- **Real-space advection** (second term on LHS)

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$

- It represents the movement of particles in real space:

$$\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n = \frac{\partial}{\partial x_1} v_{\mathrm{p}1} n + \frac{\partial}{\partial x_2} v_{\mathrm{p}2} n + \frac{\partial}{\partial x_3} v_{\mathrm{p}3} n$$

- Particle velocity plays an important role since it generates spatial transport: **active variable**
- Other internal coordinates (e.g. particle size) are **passive variables**: carried along with given velocity



Real-Space Advection and Integration in CFD Codes

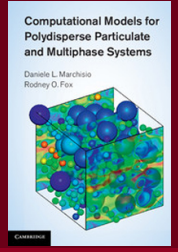
- **Real-space advection** (second term on LHS)

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n + \frac{\partial}{\partial \mathbf{v}_\mathrm{p}} \cdot \left(\langle \mathbf{A}_\mathrm{fp} \rangle_1 + \langle \mathbf{A}_\mathrm{p} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{p}} \cdot \langle \mathbf{G}_\mathrm{p} \rangle_1 n + \frac{\partial}{\partial \mathbf{v}_\mathrm{f}} \cdot \left(\langle \mathbf{A}_\mathrm{pf} \rangle_1 + \langle \mathbf{A}_\mathrm{f} \rangle_1 \right) n + \frac{\partial}{\partial \boldsymbol{\xi}_\mathrm{f}} \cdot \langle \mathbf{G}_\mathrm{f} \rangle_1 n = \mathcal{S}_1$$

- It represents the movement of particles in real space:

$$\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v}_\mathrm{p} n = \frac{\partial}{\partial x_1} v_{\mathrm{p}1} n + \frac{\partial}{\partial x_2} v_{\mathrm{p}2} n + \frac{\partial}{\partial x_3} v_{\mathrm{p}3} n$$

- Particle velocity plays an important role since it generates spatial transport: **active variable**
- Other internal coordinates (e.g. particle size) are **passive variables**: carried along with given velocity



Real-Space Advection and Integration in CFD Codes

- One strategy that facilitates implementation in CFD codes is to integrate NDF with respect to particle velocity (and to assume that phase-space variables for fluid are slaved to their average values: monokinetic fluid limit):

$$n^*(t,\mathbf{x},\boldsymbol{\xi}_\mathrm{p}) := \int n(t,\mathbf{x},\mathbf{v}_\mathrm{p},\boldsymbol{\xi}_\mathrm{p})\delta(\mathbf{v}_\mathrm{f}-\mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f}-\boldsymbol{\phi}_\mathrm{f})\,\mathrm{d}\mathbf{v}_\mathrm{p}\,\mathrm{d}\mathbf{v}_\mathrm{f}\,\mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

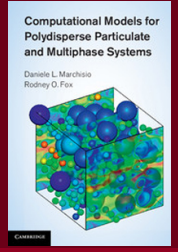
- An expression for the real-space advection term is obtained:

$$\frac{\partial}{\partial\mathbf{x}}\cdot\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* = \frac{\partial}{\partial x_1}\langle U_{\mathrm{p}1}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_2}\langle U_{\mathrm{p}2}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_3}\langle U_{\mathrm{p}3}|\boldsymbol{\xi}_\mathrm{p}\rangle n^*$$

- By definition, **conditional particle-phase velocity** is

$$\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle := \frac{\int\mathbf{v}_\mathrm{p}n\,\mathrm{d}\mathbf{v}_\mathrm{p}}{\int n\,\mathrm{d}\mathbf{v}_\mathrm{p}} = \frac{1}{n^*}\left(\int v_{\mathrm{p}1}n\,\mathrm{d}\mathbf{v}_\mathrm{p}, \int v_{\mathrm{p}2}n\,\mathrm{d}\mathbf{v}_\mathrm{p}, \int v_{\mathrm{p}3}n\,\mathrm{d}\mathbf{v}_\mathrm{p}\right)$$

and represents mean velocity of particles with internal coordinates equal to $\boldsymbol{\xi}_\mathrm{p}$, or equivalently average particle velocity conditioned on internal coordinates



Real-Space Advection and Integration in CFD Codes

- One strategy that facilitates implementation in CFD codes is to integrate NDF with respect to particle velocity (and to assume that phase-space variables for fluid are slaved to their average values: monokinetic fluid limit):

$$n^*(t,\mathbf{x},\boldsymbol{\xi}_\mathrm{p}) := \int n(t,\mathbf{x},\mathbf{v}_\mathrm{p},\boldsymbol{\xi}_\mathrm{p})\delta(\mathbf{v}_\mathrm{f}-\mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f}-\boldsymbol{\phi}_\mathrm{f})\,\mathrm{d}\mathbf{v}_\mathrm{p}\,\mathrm{d}\mathbf{v}_\mathrm{f}\,\mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

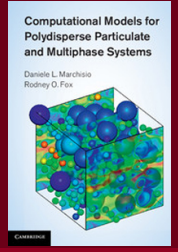
- An expression for the real-space advection term is obtained:

$$\frac{\partial}{\partial \mathbf{x}}\cdot\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* = \frac{\partial}{\partial x_1}\langle U_{\mathrm{p}1}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_2}\langle U_{\mathrm{p}2}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_3}\langle U_{\mathrm{p}3}|\boldsymbol{\xi}_\mathrm{p}\rangle n^*$$

- By definition, **conditional particle-phase velocity** is

$$\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle := \frac{\int \mathbf{v}_\mathrm{p} n\,\mathrm{d}\mathbf{v}_\mathrm{p}}{\int n\,\mathrm{d}\mathbf{v}_\mathrm{p}} = \frac{1}{n^*}\left(\int v_{\mathrm{p}1} n\,\mathrm{d}\mathbf{v}_\mathrm{p}, \int v_{\mathrm{p}2} n\,\mathrm{d}\mathbf{v}_\mathrm{p}, \int v_{\mathrm{p}3} n\,\mathrm{d}\mathbf{v}_\mathrm{p}\right)$$

and represents mean velocity of particles with internal coordinates equal to $\boldsymbol{\xi}_\mathrm{p}$, or equivalently average particle velocity conditioned on internal coordinates



Real-Space Advection and Integration in CFD Codes

- One strategy that facilitates implementation in CFD codes is to integrate NDF with respect to particle velocity (and to assume that phase-space variables for fluid are slaved to their average values: monokinetic fluid limit):

$$n^*(t,\mathbf{x},\boldsymbol{\xi}_\mathrm{p}) := \int n(t,\mathbf{x},\mathbf{v}_\mathrm{p},\boldsymbol{\xi}_\mathrm{p})\delta(\mathbf{v}_\mathrm{f}-\mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f}-\boldsymbol{\phi}_\mathrm{f})\,\mathrm{d}\mathbf{v}_\mathrm{p}\,\mathrm{d}\mathbf{v}_\mathrm{f}\,\mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

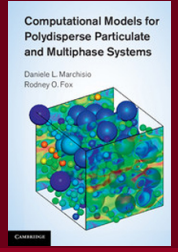
- An expression for the real-space advection term is obtained:

$$\frac{\partial}{\partial\mathbf{x}}\cdot\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* = \frac{\partial}{\partial x_1}\langle U_{\mathrm{p}1}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_2}\langle U_{\mathrm{p}2}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_3}\langle U_{\mathrm{p}3}|\boldsymbol{\xi}_\mathrm{p}\rangle n^*$$

- By definition, conditional particle-phase velocity is

$$\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle := \frac{\int\mathbf{v}_\mathrm{p}n\,\mathrm{d}\mathbf{v}_\mathrm{p}}{\int n\,\mathrm{d}\mathbf{v}_\mathrm{p}} = \frac{1}{n^*}\left(\int v_{\mathrm{p}1}n\,\mathrm{d}\mathbf{v}_\mathrm{p}, \int v_{\mathrm{p}2}n\,\mathrm{d}\mathbf{v}_\mathrm{p}, \int v_{\mathrm{p}3}n\,\mathrm{d}\mathbf{v}_\mathrm{p}\right)$$

and represents mean velocity of particles with internal coordinates equal to $\boldsymbol{\xi}_\mathrm{p}$, or equivalently average particle velocity conditioned on internal coordinates



Real-Space Advection and Integration in CFD Codes

- One strategy that facilitates implementation in CFD codes is to integrate NDF with respect to particle velocity (and to assume that phase-space variables for fluid are slaved to their average values: monokinetic fluid limit):

$$n^*(t,\mathbf{x},\boldsymbol{\xi}_\mathrm{p}) := \int n(t,\mathbf{x},\mathbf{v}_\mathrm{p},\boldsymbol{\xi}_\mathrm{p})\delta(\mathbf{v}_\mathrm{f}-\mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f}-\boldsymbol{\phi}_\mathrm{f})\,\mathrm{d}\mathbf{v}_\mathrm{p}\,\mathrm{d}\mathbf{v}_\mathrm{f}\,\mathrm{d}\boldsymbol{\xi}_\mathrm{f}$$

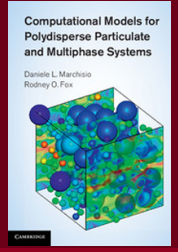
- An expression for the real-space advection term is obtained:

$$\frac{\partial}{\partial\mathbf{x}}\cdot\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* = \frac{\partial}{\partial x_1}\langle U_{\mathrm{p}1}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_2}\langle U_{\mathrm{p}2}|\boldsymbol{\xi}_\mathrm{p}\rangle n^* + \frac{\partial}{\partial x_3}\langle U_{\mathrm{p}3}|\boldsymbol{\xi}_\mathrm{p}\rangle n^*$$

- By definition, **conditional particle-phase velocity** is

$$\langle\mathbf{U}_\mathrm{p}|\boldsymbol{\xi}_\mathrm{p}\rangle := \frac{\int\mathbf{v}_\mathrm{p}n\,\mathrm{d}\mathbf{v}_\mathrm{p}}{\int n\,\mathrm{d}\mathbf{v}_\mathrm{p}} = \frac{1}{n^*}\left(\int v_{\mathrm{p}1}n\,\mathrm{d}\mathbf{v}_\mathrm{p},\int v_{\mathrm{p}2}n\,\mathrm{d}\mathbf{v}_\mathrm{p},\int v_{\mathrm{p}3}n\,\mathrm{d}\mathbf{v}_\mathrm{p}\right)$$

and represents mean velocity of particles with internal coordinates equal to $\boldsymbol{\xi}_\mathrm{p}$, or equivalently average particle velocity conditioned on internal coordinates



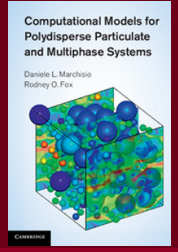
Real-Space Advection and Integration in CFD Codes

Conditioned particle-phase velocity $\langle \mathbf{U}_\mathrm{p} | \boldsymbol{\xi}_\mathrm{p} \rangle$ can be determined (in an Eulerian framework) with different approaches:

- **pseudo-homogeneous or dusty gas model**: particles move with velocity of continuous phase ($\mathrm{St}_\mathrm{p} \to 0$) including (for submicron particles) or neglecting (with larger particles) Brownian diffusion
- **equilibrium or algebraic Eulerian model**: particles move with velocity calculated from algebraic expression ($0 < \mathrm{St}_\mathrm{p} < 0.1$)
- **Eulerian two-fluid model**: particles move with velocity calculated with differential equation ($\mathrm{St}_\mathrm{p} > 0.1$)

Choice depends on particle Stokes number (ratio between particle and fluid relaxation time scales):

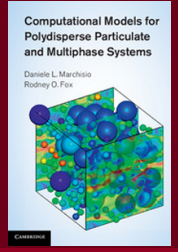
$$\mathrm{St}_\mathrm{p} = \frac{\rho_\mathrm{p} d_\mathrm{p}^2}{18 \rho_\mathrm{f} \nu_\mathrm{f} \tau_\mathrm{f}}$$



Real-Space Advection and Integration in CFD Codes

Pseudo-homogeneous or dusty gas model:

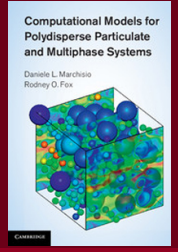
- There is one particle velocity field, identical to fluid velocity
- Preferential accumulation and segregation are not predicted as particles are transported as scalars in continuous phase
- If system is very dilute (one-way coupling), properties of continuous phase (i.e., density and viscosity) are assumed to be equal to those of fluid
- If particle concentration starts to influence fluid phase (two-way coupling), a modified density and viscosity for continuous phase are generally introduced → mixture
- Depending on size of particles Brownian motion may have to be included



Real-Space Advection and Integration in CFD Codes

Pseudo-homogeneous or dusty gas model:

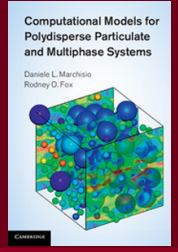
- There is one particle velocity field, identical to fluid velocity
- Preferential accumulation and segregation are not predicted as particles are transported as scalars in continuous phase
- If system is very dilute (one-way coupling), properties of continuous phase (i.e., density and viscosity) are assumed to be equal to those of fluid
- If particle concentration starts to influence fluid phase (two-way coupling), a modified density and viscosity for continuous phase are generally introduced $\rightarrow$ mixture
- Depending on size of particles Brownian motion may have to be included



Real-Space Advection and Integration in CFD Codes

Pseudo-homogeneous or dusty gas model:

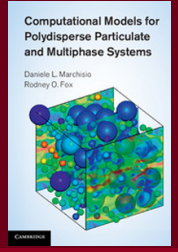
- There is one particle velocity field, identical to fluid velocity
- Preferential accumulation and segregation are not predicted as particles are transported as scalars in continuous phase
- If system is very dilute (one-way coupling), properties of continuous phase (i.e., density and viscosity) are assumed to be equal to those of fluid
- If particle concentration starts to influence fluid phase (two-way coupling), a modified density and viscosity for continuous phase are generally introduced → mixture
- Depending on size of particles Brownian motion may have to be included



Real-Space Advection and Integration in CFD Codes

Pseudo-homogeneous or dusty gas model:

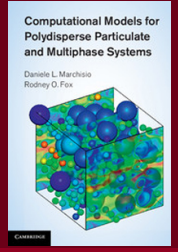
- There is one particle velocity field, identical to fluid velocity
- Preferential accumulation and segregation are not predicted as particles are transported as scalars in continuous phase
- If system is very dilute (one-way coupling), properties of continuous phase (i.e., density and viscosity) are assumed to be equal to those of fluid
- If particle concentration starts to influence fluid phase (two-way coupling), a modified density and viscosity for continuous phase are generally introduced → mixture
- Depending on size of particles Brownian motion may have to be included



Real-Space Advection and Integration in CFD Codes

Pseudo-homogeneous or dusty gas model:

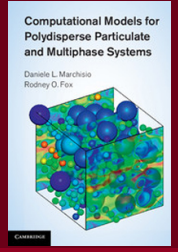
- There is one particle velocity field, identical to fluid velocity
- Preferential accumulation and segregation are not predicted as particles are transported as scalars in continuous phase
- If system is very dilute (one-way coupling), properties of continuous phase (i.e., density and viscosity) are assumed to be equal to those of fluid
- If particle concentration starts to influence fluid phase (two-way coupling), a modified density and viscosity for continuous phase are generally introduced → mixture
- Depending on size of particles Brownian motion may have to be included



Real-Space Advection and Integration in CFD Codes

Pseudo-homogeneous or dusty gas model:

- There is one particle velocity field, identical to fluid velocity
- Preferential accumulation and segregation are not predicted as particles are transported as scalars in continuous phase
- If system is very dilute (one-way coupling), properties of continuous phase (i.e., density and viscosity) are assumed to be equal to those of fluid
- If particle concentration starts to influence fluid phase (two-way coupling), a modified density and viscosity for continuous phase are generally introduced → mixture
- Depending on size of particles Brownian motion may have to be included



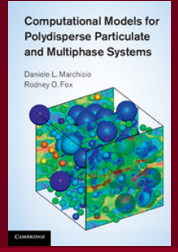
Real-Space Advection and Integration in CFD Codes

Equilibrium or algebraic Eulerian model:

- Considering only drag and gravity:

$$\langle \mathbf{U}_\mathrm{p} | \boldsymbol{\xi}_\mathrm{p} \rangle = \mathbf{U}_\mathrm{f} + \left(\frac{\mathrm{D}\mathbf{U}_\mathrm{f}}{\mathrm{D}t} - \mathbf{g} \right) \tau_\mathrm{p}(\boldsymbol{\xi}_\mathrm{p}) + O\left(\tau_\mathrm{p}^2\right)$$

- One-way coupling (dilute systems): momentum balance equation for continuous phase and algebraic equation for different particle classes: $\mathbf{U}_{\mathrm{p}k} := \langle \mathbf{U}_\mathrm{p} | \boldsymbol{\xi}_\mathrm{p} = \boldsymbol{\xi}_{\mathrm{p}k} \rangle$ with $k = 1, \ldots, N$
- Denser particulate systems (two- or three-way coupling): momentum balance equation for mixture
- Moderately dense particulate systems are generally not well described due to importance of particle–particle collisions and particle trajectory crossing



Real-Space Advection and Integration in CFD Codes

Eulerian two-fluid model:

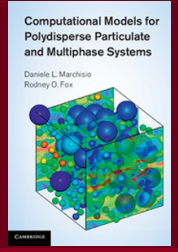
- When particle Stokes number is not small particle velocity must be calculated from disperse-phase momentum equation:

$$\frac{\partial \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{p}}}{\partial t} + \boldsymbol{\nabla} \cdot \varrho_{\mathrm{p}} \mathbf{U}_{\mathrm{p}} \otimes \mathbf{U}_{\mathrm{p}} = -\alpha_{\mathrm{p}} \boldsymbol{\nabla} p_{\mathrm{f}} + \varrho_{\mathrm{p}} \mathbf{g} + \frac{\varrho_{\mathrm{p}} \rho_{\mathrm{f}} C_{\mathrm{D}} A_{\mathrm{d}}}{2\gamma \rho_{\mathrm{p}}} |\mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{p}}| (\mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{p}})$$
$$+ \frac{\varrho_{\mathrm{p}} C_{\mathrm{vm}} \rho_{\mathrm{f}}}{\gamma \rho_{\mathrm{p}}} (\mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{p}}) \cdot \boldsymbol{\nabla} \mathbf{U}_{\mathrm{f}} + \frac{\varrho_{\mathrm{p}} C_{\mathrm{L}} \rho_{\mathrm{f}}}{\gamma \rho_{\mathrm{p}}} (\mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{p}}) \times (\boldsymbol{\nabla} \times \mathbf{U}_{\mathrm{f}})$$

- Likewise, fluid-phase momentum balance is

$$\frac{\partial \varrho_{\mathrm{f}} \mathbf{U}_{\mathrm{f}}}{\partial t} + \boldsymbol{\nabla} \cdot \varrho_{\mathrm{f}} \mathbf{U}_{\mathrm{f}} \otimes \mathbf{U}_{\mathrm{f}} = \boldsymbol{\nabla} \cdot \boldsymbol{\tau}_{\mathrm{f}} - \alpha_{\mathrm{f}} \boldsymbol{\nabla} p_{\mathrm{f}} + \varrho_{\mathrm{f}} \mathbf{g} + \frac{\varrho_{\mathrm{p}} \rho_{\mathrm{f}} C_{\mathrm{D}} A_{\mathrm{d}}}{2\gamma \rho_{\mathrm{p}}} |\mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{p}}| (\mathbf{U}_{\mathrm{p}} - \mathbf{U}_{\mathrm{f}})$$
$$+ \frac{\varrho_{\mathrm{p}} C_{\mathrm{vm}} \rho_{\mathrm{f}}}{\gamma \rho_{\mathrm{p}}} (\mathbf{U}_{\mathrm{p}} - \mathbf{U}_{\mathrm{f}}) \cdot \boldsymbol{\nabla} \mathbf{U}_{\mathrm{f}} + \frac{\varrho_{\mathrm{p}} C_{\mathrm{L}} \rho_{\mathrm{f}}}{\gamma \rho_{\mathrm{p}}} (\mathbf{U}_{\mathrm{p}} - \mathbf{U}_{\mathrm{f}}) \times (\boldsymbol{\nabla} \times \mathbf{U}_{\mathrm{f}})$$

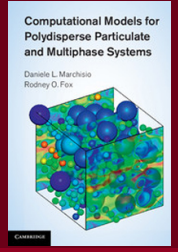
- Approach can be extended to different particle classes, $\mathbf{U}_{\mathrm{p}k} := \langle \mathbf{U}_{\mathrm{p}} | \boldsymbol{\xi}_{\mathrm{p}} = \boldsymbol{\xi}_{\mathrm{p}k} \rangle$ with $k = 1, \ldots, N$, by solving multiple momentum balance equations: **multi-fluid model**



Guidelines for Real-Space Advection

There are three key parameters: particle loading, ϱ_p/ϱ_f, particle Stokes number, St, and polydispersity, PDI

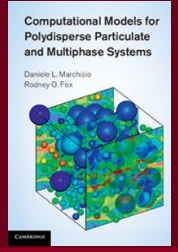
1. **Diffusion equation:** sub-micron particles subject to Brownian motion
2. **Pseudo-homogeneous or dusty-gas model:** very small particle Stokes number and limited polydispersity (mom. bal. eq. for cont. phase if system is dilute or mixture otherwise)
3. **Equilibrium or algebraic Eulerian model with single conditional velocity based on mean particle size:** small particle Stokes number and limited polydispersity (mom. bal. eq. ...)
4. **Equilibrium or algebraic Eulerian model with conditional velocities:** small particle Stokes number and non-negligible polydispersity (mom. bal. eq. ...)
5. **Eulerian two-fluid model with particle-phase velocity based on mean particle size:** small particle Stokes number and limited polydispersity (in both dilute and dense systems)
6. **Eulerian multi-fluid model:** small particle Stokes number and large polydispersity (in both dilute and dense systems)



Guidelines for Real-Space Advection

There are three key parameters: particle loading, ϱ_p/ϱ_f, particle Stokes number, St, and polydispersity, PDI

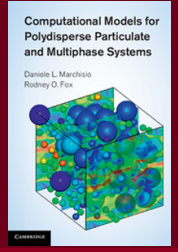
1. **Diffusion equation:** sub-micron particles subject to Brownian motion
2. **Pseudo-homogeneous or dusty-gas model:** very small particle Stokes number and limited polydispersity (mom. bal. eq. for cont. phase if system is dilute or mixture otherwise)
3. **Equilibrium or algebraic Eulerian model with single conditional velocity based on mean particle size:** small particle Stokes number and limited polydispersity (mom. bal. eq. ...)
4. **Equilibrium or algebraic Eulerian model with conditional velocities:** small particle Stokes number and non-negligible polydispersity (mom. bal. eq. ...)
5. **Eulerian two-fluid model with particle-phase velocity based on mean particle size:** small particle Stokes number and limited polydispersity (in both dilute and dense systems)
6. **Eulerian multi-fluid model:** small particle Stokes number and large polydispersity (in both dilute and dense systems)



Guidelines for Real-Space Advection

There are three key parameters: particle loading, ϱ_p/ϱ_f, particle Stokes number, St, and polydispersity, PDI

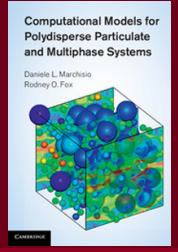
1. **Diffusion equation:** sub-micron particles subject to Brownian motion
2. **Pseudo-homogeneous or dusty-gas model:** very small particle Stokes number and limited polydispersity (mom. bal. eq. for cont. phase if system is dilute or mixture otherwise)
3. **Equilibrium or algebraic Eulerian model with single conditional velocity based on mean particle size:** small particle Stokes number and limited polydispersity (mom. bal. eq. ...)
4. **Equilibrium or algebraic Eulerian model with conditional velocities:** small particle Stokes number and non-negligible polydispersity (mom. bal. eq. ...)
5. **Eulerian two-fluid model with particle-phase velocity based on mean particle size:** small particle Stokes number and limited polydispersity (in both dilute and dense systems)
6. **Eulerian multi-fluid model:** small particle Stokes number and large polydispersity (in both dilute and dense systems)



Guidelines for Real-Space Advection

There are three key parameters: particle loading, ϱ_p/ϱ_f, particle Stokes number, St, and polydispersity, PDI

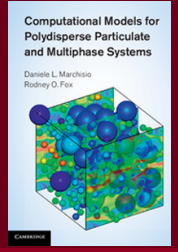
1. **Diffusion equation:** sub-micron particles subject to Brownian motion
2. **Pseudo-homogeneous or dusty-gas model:** very small particle Stokes number and limited polydispersity (mom. bal. eq. for cont. phase if system is dilute or mixture otherwise)
3. **Equilibrium or algebraic Eulerian model with single conditional velocity based on mean particle size:** small particle Stokes number and limited polydispersity (mom. bal. eq. ...)
4. **Equilibrium or algebraic Eulerian model with conditional velocities:** small particle Stokes number and non-negligible polydispersity (mom. bal. eq. ...)
5. **Eulerian two-fluid model with particle-phase velocity based on mean particle size:** small particle Stokes number and limited polydispersity (in both dilute and dense systems)
6. **Eulerian multi-fluid model:** small particle Stokes number and large polydispersity (in both dilute and dense systems)



Guidelines for Real-Space Advection

There are three key parameters: particle loading, $\varrho_{\mathrm{p}}/\varrho_{\mathrm{f}}$, particle Stokes number, St, and polydispersity, PDI

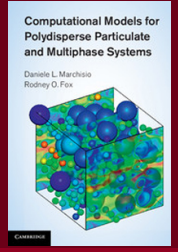
1. **Diffusion equation:** sub-micron particles subject to Brownian motion
2. **Pseudo-homogeneous or dusty-gas model:** very small particle Stokes number and limited polydispersity (mom. bal. eq. for cont. phase if system is dilute or mixture otherwise)
3. **Equilibrium or algebraic Eulerian model with single conditional velocity based on mean particle size:** small particle Stokes number and limited polydispersity (mom. bal. eq. ...)
4. **Equilibrium or algebraic Eulerian model with conditional velocities:** small particle Stokes number and non-negligible polydispersity (mom. bal. eq. ...)
5. **Eulerian two-fluid model with particle-phase velocity based on mean particle size:** small particle Stokes number and limited polydispersity (in both dilute and dense systems)
6. **Eulerian multi-fluid model:** small particle Stokes number and large polydispersity (in both dilute and dense systems)



Guidelines for Real-Space Advection

There are three key parameters: particle loading, ϱ_p/ϱ_f, particle Stokes number, St, and polydispersity, PDI

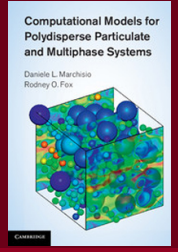
1. **Diffusion equation:** sub-micron particles subject to Brownian motion
2. **Pseudo-homogeneous or dusty-gas model:** very small particle Stokes number and limited polydispersity (mom. bal. eq. for cont. phase if system is dilute or mixture otherwise)
3. **Equilibrium or algebraic Eulerian model with single conditional velocity based on mean particle size:** small particle Stokes number and limited polydispersity (mom. bal. eq. ...)
4. **Equilibrium or algebraic Eulerian model with conditional velocities:** small particle Stokes number and non-negligible polydispersity (mom. bal. eq. ...)
5. **Eulerian two-fluid model with particle-phase velocity based on mean particle size:** small particle Stokes number and limited polydispersity (in both dilute and dense systems)
6. **Eulerian multi-fluid model:** small particle Stokes number and large polydispersity (in both dilute and dense systems)



Guidelines for Real-Space Advection

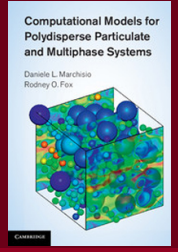
There are three key parameters: particle loading, ϱ_p/ϱ_f, particle Stokes number, St, and polydispersity, PDI

1. **Diffusion equation:** sub-micron particles subject to Brownian motion
2. **Pseudo-homogeneous or dusty-gas model:** very small particle Stokes number and limited polydispersity (mom. bal. eq. for cont. phase if system is dilute or mixture otherwise)
3. **Equilibrium or algebraic Eulerian model with single conditional velocity based on mean particle size:** small particle Stokes number and limited polydispersity (mom. bal. eq. ...)
4. **Equilibrium or algebraic Eulerian model with conditional velocities:** small particle Stokes number and non-negligible polydispersity (mom. bal. eq. ...)
5. **Eulerian two-fluid model with particle-phase velocity based on mean particle size:** small particle Stokes number and limited polydispersity (in both dilute and dense systems)
6. **Eulerian multi-fluid model:** small particle Stokes number and large polydispersity (in both dilute and dense systems)



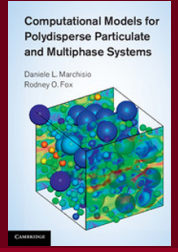
Part 3

MESOSCALE MODELS FOR PHYSICAL AND CHEMICAL PROCESSES



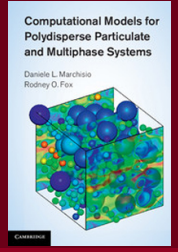
Formulation of Mesoscale Models

- Conditional phase-space velocities and discontinuous "source" term need mesoscale closures: averaged models for microscale chemical and physical processes
- Simplest procedure is to derive them from single-particle statistics by assuming that all particles are statistically identical and interacting with mean-field quantities which model effects of all other particles
- Effect of multi-particle statistics on phase-space velocities is generally introduced in form of corrections, based for example on disperse-phase volume fraction, that will reduce to isolated-particle statistics in dilute limit
- We will discuss in particular mesoscale models for continuous processes due to fluid–particle momentum transfer ($\langle \mathbf{A}_{\mathrm{fp}} \rangle_1$), heat and mass transfer ($\langle \mathbf{G}_{\mathrm{p}} \rangle_1$) and discontinuous jumps (or point processes) contained in the source term $\mathcal{S}_1$ (zero-, first- and second-order processes)



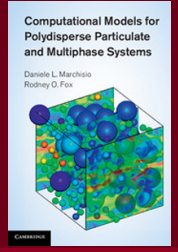
Formulation of Mesoscale Models

- Conditional phase-space velocities and discontinuous "source" term need mesoscale closures: averaged models for microscale chemical and physical processes
- Simplest procedure is to derive them from single-particle statistics by assuming that all particles are statistically identical and interacting with mean-field quantities which model effects of all other particles
- Effect of multi-particle statistics on phase-space velocities is generally introduced in form of corrections, based for example on disperse-phase volume fraction, that will reduce to isolated-particle statistics in dilute limit
- We will discuss in particular mesoscale models for continuous processes due to fluid–particle momentum transfer ($\langle \mathbf{A}_{\mathrm{fp}} \rangle_1$), heat and mass transfer ($\langle \mathbf{G}_{\mathrm{p}} \rangle_1$) and discontinuous jumps (or point processes) contained in the source term $\mathcal{S}_1$ (zero-, first- and second-order processes)



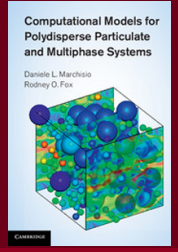
Formulation of Mesoscale Models

- Conditional phase-space velocities and discontinuous "source" term need mesoscale closures: averaged models for microscale chemical and physical processes
- Simplest procedure is to derive them from single-particle statistics by assuming that all particles are statistically identical and interacting with mean-field quantities which model effects of all other particles
- Effect of multi-particle statistics on phase-space velocities is generally introduced in form of corrections, based for example on disperse-phase volume fraction, that will reduce to isolated-particle statistics in dilute limit
- We will discuss in particular mesoscale models for continuous processes due to fluid–particle momentum transfer ($\langle \mathbf{A}_{\mathrm{fp}} \rangle_1$), heat and mass transfer ($\langle \mathbf{G}_{\mathrm{p}} \rangle_1$) and discontinuous jumps (or point processes) contained in the source term $\mathcal{S}_1$ (zero-, first- and second-order processes)



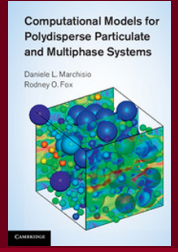
Formulation of Mesoscale Models

- Conditional phase-space velocities and discontinuous "source" term need mesoscale closures: averaged models for microscale chemical and physical processes
- Simplest procedure is to derive them from single-particle statistics by assuming that all particles are statistically identical and interacting with mean-field quantities which model effects of all other particles
- Effect of multi-particle statistics on phase-space velocities is generally introduced in form of corrections, based for example on disperse-phase volume fraction, that will reduce to isolated-particle statistics in dilute limit
- We will discuss in particular mesoscale models for continuous processes due to fluid–particle momentum transfer ($\langle \mathbf{A}_{\mathrm{fp}} \rangle_1$), heat and mass transfer ($\langle \mathbf{G}_{\mathrm{p}} \rangle_1$) and discontinuous jumps (or point processes) contained in the source term $\mathcal{S}_1$ (zero-, first- and second-order processes)



Formulation of Mesoscale Models

- Conditional phase-space velocities and discontinuous "source" term need mesoscale closures: averaged models for microscale chemical and physical processes
- Simplest procedure is to derive them from single-particle statistics by assuming that all particles are statistically identical and interacting with mean-field quantities which model effects of all other particles
- Effect of multi-particle statistics on phase-space velocities is generally introduced in form of corrections, based for example on disperse-phase volume fraction, that will reduce to isolated-particle statistics in dilute limit
- We will discuss in particular mesoscale models for continuous processes due to fluid–particle momentum transfer ($\langle \mathbf{A}_{\mathrm{fp}} \rangle_1$), heat and mass transfer ($\langle \mathbf{G}_{\mathrm{p}} \rangle_1$) and discontinuous jumps (or point processes) contained in the source term $\mathcal{S}_1$ (zero-, first- and second-order processes)



Mesoscale Model for Momentum Transfer

Buoyancy and drag forces ($A_D/V_p = 3/(2d_p)$ for spheres)

$$\langle \mathbf{A}_{fp} \rangle_1 = -\frac{1}{\rho_p}\frac{\partial p_f}{\partial \mathbf{x}} + \frac{C_D}{2}\frac{A_D}{V_p}\frac{\rho_f}{\rho_p}|\mathbf{U}_f - \mathbf{U}_p|(\mathbf{U}_f - \mathbf{U}_p)$$

- Drag model of Schiller and Nauman ($\mathrm{Re}_p = |\mathbf{U}_f - \mathbf{U}_p| d_p/\nu_f$):

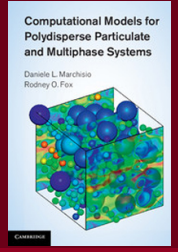
$$C_D = \begin{cases} \frac{24}{\mathrm{Re}_p}\left(1 + 0.15\mathrm{Re}_p^{0.687}\right) & \text{for } \mathrm{Re}_p \le 1000 \\ 0.445 & \text{for } \mathrm{Re}_p > 1000 \end{cases}$$

- Drag Cunningham correction factor for rarefied fluid phase:

$$\frac{C_D^*}{C_D} = \frac{1}{1 + \mathrm{Kn}\left[2.49 + 0.84\exp\left(-\frac{1.74}{\mathrm{Kn}}\right)\right]}$$

- Drag model of Wen and Yu ($\beta = 3.65$):

$$C_D = \frac{24}{\mathrm{Re}_p}\left[1 + 0.15(\alpha_f \mathrm{Re}_p)^{0.687}\right]\alpha_f^{-\beta}$$



Mesoscale Model for Momentum Transfer

Buoyancy and drag forces (Eo $= g\Delta\rho d_\mathrm{p}/\sigma$)

- Drag model by Tomiyama for pure fluid–fluid systems:

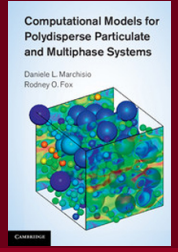
$$C_\mathrm{D} = \max\left\{\min\left[\frac{16}{\mathrm{Re}_\mathrm{p}}(1+0.15\mathrm{Re}_\mathrm{p}^{0.687}), \frac{48}{\mathrm{Re}_\mathrm{p}}\right], \frac{8}{3}\left(\frac{\mathrm{Eo}}{\mathrm{Eo}+4}\right)\right\}$$

- Slightly contaminated systems:

$$C_\mathrm{D} = \max\left\{\min\left[\frac{24}{\mathrm{Re}_\mathrm{p}}(1+0.15\mathrm{Re}_\mathrm{p}^{0.687}), \frac{72}{\mathrm{Re}_\mathrm{p}}\right], \frac{8}{3}\left(\frac{\mathrm{Eo}}{\mathrm{Eo}+4}\right)\right\}$$

- Fully contaminated systems:

$$C_\mathrm{D} = \max\left[\frac{24}{\mathrm{Re}_\mathrm{p}}(1+0.15\mathrm{Re}_\mathrm{p}^{0.687}), \frac{8}{3}\left(\frac{\mathrm{Eo}}{\mathrm{Eo}+4}\right)\right]$$



Mesoscale Model for Momentum Transfer

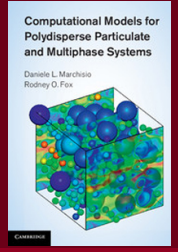
- Drag model by Buffo (for contaminated systems) with correction to account for bubble-swarm effect:

$$C_D = (1-\alpha_g)^{C_1} \max\left[\frac{24}{\mathrm{Re}_{\mathrm{eff}}}\left(1+0.15\mathrm{Re}_{\mathrm{eff}}^{0.687}\right), \frac{8}{3}\frac{Eo}{Eo+4}\right]$$

  and second correction to account for fluid-phase turbulence:

$$\mathrm{Re}_{\mathrm{eff}} = \frac{d_{\mathrm{p}}\rho_{\mathrm{f}}|\mathbf{U}_{\mathrm{f}}-\mathbf{U}_{\mathrm{p}}|}{\mu_{\mathrm{eff}}}, \quad \mu_{\mathrm{eff}} = \mu_{\mathrm{f}} + C_2\rho_{\mathrm{f}}\frac{k^2}{\varepsilon}$$

- C_1 and C_2 are fitting parameters
- For gas–liquid systems: $C_2 = 0.01$ and $C_1 = -1.8$, predict well gas hold-up



Mesoscale Model for Momentum Transfer

Additional forces

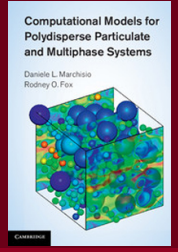
- Virtual mass and (Saffman) lift force:

$$\langle \mathbf{A}_{\mathrm{fp}} \rangle_1 = \frac{\rho_{\mathrm{f}} C_{\mathrm{vm}}}{\rho_{\mathrm{p}} \gamma} \left(\mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{p}} \right) \cdot \boldsymbol{\nabla} \mathbf{U}_{\mathrm{f}} + \frac{\rho_{\mathrm{f}} C_{\mathrm{L}}}{\rho_{\mathrm{p}} \gamma} \left(\mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{p}} \right) \times \left(\boldsymbol{\nabla} \times \mathbf{U}_{\mathrm{f}} \right)$$

  where γ effective volume coefficient, $C_{\mathrm{L}} = 0.25$ and $C_{\mathrm{vm}} = 0.5$

- Boussinesq-Basset history force:

$$\langle \mathbf{A}_{\mathrm{fp}} \rangle_1 = \frac{9}{d_{\mathrm{p}}} \frac{\rho_{\mathrm{f}}}{\rho_{\mathrm{p}}} \sqrt{\frac{\nu_{\mathrm{f}}}{\pi}} \int_{-\infty}^{t} \frac{1}{\sqrt{t - \tau}} \left(\frac{\mathrm{D}\mathbf{U}_{\mathrm{f}}}{\mathrm{D}\tau} - \frac{\mathrm{d}\mathbf{U}_{\mathrm{p}}}{\mathrm{d}\tau} \right) \mathrm{d}\tau$$



Mesoscale Model for Momentum Transfer

Additional forces

- Brownian force:

$$\langle \mathbf{A}_{\mathrm{fp}} \rangle_1 = \frac{1}{M_{\mathrm{p}}} \left(12 d_{\mathrm{p}} \mu_{\mathrm{f}} k_{\mathrm{B}} T_{\mathrm{f}} \right)^{1/2} \mathbf{W}(t)$$

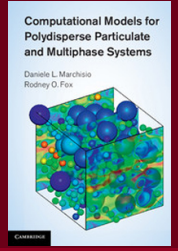
- Thermophoretic force for spherical particles and large Knudsen numbers (e.g., nanoparticles in flames):

$$\langle \mathbf{A}_{\mathrm{fp}} \rangle_1 = -\frac{6 p_{\mathrm{f}} \lambda_{\mathrm{f}}}{\pi T_{\mathrm{f}} \rho_{\mathrm{p}} d_{\mathrm{p}}} \frac{\partial T_{\mathrm{f}}}{\partial \mathbf{x}}$$

- Knudsen numbers smaller than one (temp. gradient within particle):

$$\langle \mathbf{A}_{\mathrm{fp}} \rangle_1 = -\left(\frac{36 \nu_{\mathrm{f}}^2 C_{\mathrm{s}}}{T_{\mathrm{f}} d_{\mathrm{p}}^2} \right) \left(\frac{\rho_{\mathrm{f}}}{\rho_{\mathrm{p}}} \right) \left(\frac{1}{1 + 6 C_{\mathrm{m}} \mathrm{Kn}} \right) \left(\frac{k_{\mathrm{f}}/k_{\mathrm{p}} + 2 C_{\mathrm{t}} \mathrm{Kn}}{1 + 2 k_{\mathrm{f}}/k_{\mathrm{p}} + 4 C_{\mathrm{t}} \mathrm{Kn}} \right) \frac{\partial T_{\mathrm{f}}}{\partial \mathbf{x}}$$

k_{f} and k_{p} thermal conductivities of fluid and particle, $C_{\mathrm{s}} = 1.17$ thermal slip coefficient, $C_{\mathrm{t}} = 2.18$ thermal exchange coefficient, $C_{\mathrm{m}} = 1.14$ momentum exchange coefficient

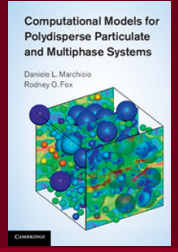


Mesoscale Model for Mass Transfer

Mesoscale variables for particle size

- (Truely) Equidimensional particles: nearly same size or spread in multiple directions and have constant material density
- Single internal coordinate: particle mass (or volume), particle surface area or particle length ($V_\mathrm{p} = k_\mathrm{V} d_\mathrm{p}^3$ and $A_\mathrm{p} = k_\mathrm{A} d_\mathrm{p}^2$)
- Extension to other systems: possible when particle shrinks or grows without changing its shape and morphology
- Define one characteristic length, d_p^*, express real particle volume and surface area: $V_\mathrm{p} = k_\mathrm{V}^* (d_\mathrm{p}^*)^3$ and $A_\mathrm{p} = k_\mathrm{A}^* (d_\mathrm{p}^*)^2$, and introduce an equivalent length, $d_\mathrm{p} = \phi_\mathrm{c} d_\mathrm{p}^*$, equal to: $\phi_\mathrm{c} = 6 k_\mathrm{V}^* / k_\mathrm{A}^*$, forcing equivalent or averaged shape factors to be: $k_\mathrm{V} = \left(k_\mathrm{A}^*\right)^3 / 216 \left(k_\mathrm{V}^*\right)^2$ and $k_\mathrm{A} = \left(k_\mathrm{A}^*\right)^3 / 36 \left(k_\mathrm{V}^*\right)^2$
- If particle size is (first) internal coordinate then:

$$\langle G_{\mathrm{p}1} \rangle_1 = \dot{\xi}_{\mathrm{p}1} = \dot{d}_\mathrm{p} = \frac{M_\mathrm{w}}{\rho_\mathrm{p}} \frac{k_\mathrm{A}}{3 k_\mathrm{V}} J$$



Mesoscale Model for Mass Transfer

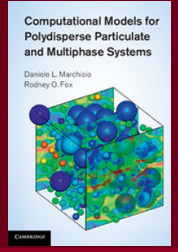
Mesoscale variables for particle size

- In case of non-equidimensional particles more than one internal coordinate must be used
- Example: needle-like particles described as rectangular parallelepiped with length $\xi_{\mathrm{p}1}$ and equal width and depth $\xi_{\mathrm{p}2}$
- Particle volume: $V_{\mathrm{p}} = \xi_{\mathrm{p}1}(\xi_{\mathrm{p}2})^2$; particle surface area: $A_{\mathrm{p}} = 2(\xi_{\mathrm{p}2})^2 + 4\xi_{\mathrm{p}1}\xi_{\mathrm{p}2}$
- If $J_{12}(\mathbf{U}_{\mathrm{p}}, \boldsymbol{\xi}_{\mathrm{p}}, \mathbf{U}_{\mathrm{f}}, \boldsymbol{\xi}_{\mathrm{f}})$ molar flux on surface delimited by $\xi_{\mathrm{p}1}$ and $\xi_{\mathrm{p}2}$, and $J_{22}(\mathbf{U}_{\mathrm{p}}, \boldsymbol{\xi}_{\mathrm{p}}, \mathbf{U}_{\mathrm{f}}, \boldsymbol{\xi}_{\mathrm{f}})$ molar flux on surface delimited by $\xi_{\mathrm{p}2}$ and $\xi_{\mathrm{p}2}$, resulting mesoscale models are:

$$\langle G_{\mathrm{p}1}\rangle_1 = \dot{\xi}_{\mathrm{p}1} = 2M_{\mathrm{w}}J_{22}/\rho_{\mathrm{p}}, \quad \langle G_{\mathrm{p}2}\rangle_1 = \dot{\xi}_{\mathrm{p}2} = 4M_{\mathrm{w}}J_{12}/\rho_{\mathrm{p}}$$

- Alternative description: particle volume and surface area

$$\dot{V}_{\mathrm{p}} = (\xi_{\mathrm{p}2})^2\dot{\xi}_{\mathrm{p}1} + 2\xi_{\mathrm{p}1}\xi_{\mathrm{p}2}\dot{\xi}_{\mathrm{p}2}, \quad \dot{A}_{\mathrm{p}} = (2\xi_{\mathrm{p}2} + 4\xi_{\mathrm{p}1})\dot{\xi}_{\mathrm{p}2} + 4\xi_{\mathrm{p}2}\dot{\xi}_{\mathrm{p}1}$$



Mesoscale Model for Mass Transfer

- Particles described as fractal objects: size of the primary particle, $\xi_{p1} = d_p$, and number of primary particles, $\xi_{p2} = N_p$
- Fractal dimension: $R_g = d_p/2\left(N_p/k_g\right)^{1/D_f}$
- Mass transfer does not change number of primary particles

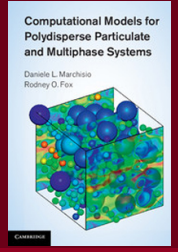
$$\dot{\xi}_{p1} = \dot{d}_p = \frac{M_w}{\rho_p}\frac{k_A k_c}{3k_V}J \quad \text{and} \quad \dot{\xi}_{p2} = \dot{N}_p = 0$$

- Mass transfer causes obliteration

$$\dot{\xi}_{p1} = \dot{d}_p = \frac{M_w}{\rho_p}\frac{k_A k_c}{3k_V}\frac{1}{(3-D_f)}J \quad \text{and} \quad \dot{\xi}_{p2} = \dot{N}_p = -\frac{M_w}{\rho_p}\frac{k_A k_c}{3k_V}\left(\frac{D_f}{3-D_f}\right)\frac{N_p}{d_p}J$$

- Mass transfer changes number of primary particles

$$\dot{\xi}_{p1} = \dot{d}_p = 0 \quad \text{and} \quad \dot{\xi}_{p2} = \dot{N}_p = \frac{M_w}{\rho_p}\frac{N_p}{d_p}\frac{k_A k_c}{k_V}J$$



Mesoscale Model for Mass Transfer

Diffusion-controlled growth for equi-dimensional particles

- Resulting growth rate

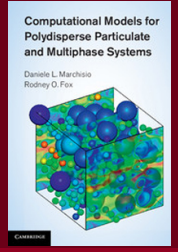
$$\dot{d}_\mathrm{p} = \frac{k_\mathrm{A}}{3k_\mathrm{V}} \frac{M_\mathrm{w}}{\rho_\mathrm{p}} k_\mathrm{d} c_\mathrm{eq} (S-1)$$

- Creeping (or Stokes) flow conditions and rigid sphere

$$\mathrm{Sh} = 1 + (1 + \mathrm{Pe})^{1/3}$$

where $\mathrm{Pe} = \mathrm{Re_p Sc}$ ($\mathrm{Sc} = \nu/D$, $\mathrm{Sh} = k_\mathrm{d} d_\mathrm{p}/D$)

- For larger particle Reynolds numbers: $\mathrm{Sh} = 2 + 0.724 \mathrm{Re}_\mathrm{p}^{0.48} \mathrm{Sc}^{1/3}$, valid for $100 < \mathrm{Re_p} \leq 2000$ and $\mathrm{Sh} = 2 + 0.425 \mathrm{Re}_\mathrm{p}^{0.55} \mathrm{Sc}^{1/3}$, valid for $2000 < \mathrm{Re_p} \leq 10^5$
- For micron-sized (or smaller) particles moving in turbulent fluids: $\mathrm{Sh} = 2.0 + 0.52 \left(\mathrm{Re}_\mathrm{p}^*\right)^{0.52} \mathrm{Sc}^{1/3}$ where $\mathrm{Re}_\mathrm{p}^* = \varepsilon_\mathrm{f}^{1/3} d_\mathrm{p}^{4/3} / \nu_\mathrm{f}$ is modified particle Reynolds number



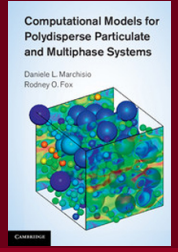
Mesoscale Model for Mass Transfer

Mass transfer to gas bubbles

- Rate of change of bubble mass: $\dot{M}_\mathrm{p} = M_\mathrm{w} k_\mathrm{d} A_\mathrm{p} \Delta c$
- In case of gas bubbles their shape is not fixed, bubble aspect ratio: e.g. $E = 1/(1 + 0.1613\mathrm{Eo}^{0.757})$
- Real area of bubble:

$$\frac{A_\mathrm{p}}{A_\mathrm{eq}} = \frac{1}{2E^{2/3}} \left[1 + \frac{E^2}{2\sqrt{1-E^2}} \ln\left(\frac{1+\sqrt{1-E^2}}{1-\sqrt{1-E^2}} \right) \right]$$

- Under creeping (or Stokes) flow conditions:
  $\mathrm{Sh} = 1 + (1 + 0.564\mathrm{Pe}^{2/3})^{3/4}$, spherical particles with $\mathrm{Re}_\mathrm{p} \geq 70$
  $\mathrm{Sh} = \frac{2}{\sqrt{\pi}} \left(1 - \frac{2.96}{\mathrm{Re}_\mathrm{p}^{1/2}} \right)^{1/2} \mathrm{Pe}^{1/2}$; Reynolds number sufficiently large, term in parenthesis is small compared to unity: $\mathrm{Sh} = 2\mathrm{Pe}^{1/2}/\sqrt{\pi}$
- For turbulent systems: $k_\mathrm{d} = cD^{0.5} \left(\varepsilon_\mathrm{f}/\nu_\mathrm{f}\right)^{0.25}$



Mesoscale Model for Zero-Order Point Processes

- In zero-order point processes $\mathcal{S}_1$ does not depend on NDF and usually represents rate of formation (i.e., nucleation or spinodal decomposition) of disperse phase:

$$\mathcal{S}_1 = J(\boldsymbol{\phi}_\mathrm{f})\delta(\mathbf{v}_\mathrm{p} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{p} - \boldsymbol{\xi}_\mathrm{pc})\delta(\mathbf{v}_\mathrm{f} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f} - \boldsymbol{\xi}_\mathrm{fc})$$

  where $J(\boldsymbol{\phi}_\mathrm{f}) \geq 0$ is rate of formation of particles with properties $\boldsymbol{\xi}_\mathrm{pc}$
- Definitions consistent with mass balance: for example if $\xi_{\mathrm{p}1}$ and $\xi_{\mathrm{f}1}$ are masses, then $\xi_{\mathrm{pc}1} = m_\mathrm{c}$ and $\xi_{\mathrm{fc}1} = -m_\mathrm{c}$
- Source terms for transport equations for total number concentration:

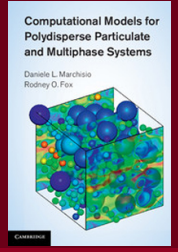
$$\mathcal{S}_\mathrm{N} = \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = J$$

  disperse-phase mass density:

$$\mathcal{S}_\mathrm{M} = \int \xi_{\mathrm{p}1}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c} J$$

  disperse-phase momentum density:

$$\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{p} = \int \xi_{\mathrm{p}1}\mathbf{v}_\mathrm{p}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c}\mathbf{U}_\mathrm{f} J$$



Mesoscale Model for Zero-Order Point Processes

- In zero-order point processes $\mathcal{S}_1$ does not depend on NDF and usually represents rate of formation (i.e., nucleation or spinodal decomposition) of disperse phase:

$$\mathcal{S}_1 = J(\boldsymbol{\phi}_\mathrm{f})\delta(\mathbf{v}_\mathrm{p} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{p} - \boldsymbol{\xi}_\mathrm{pc})\delta(\mathbf{v}_\mathrm{f} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f} - \boldsymbol{\xi}_\mathrm{fc})$$

  where $J(\boldsymbol{\phi}_\mathrm{f}) \geq 0$ is rate of formation of particles with properties $\boldsymbol{\xi}_\mathrm{pc}$
- Definitions consistent with mass balance: for example if $\xi_{\mathrm{p}1}$ and $\xi_{\mathrm{f}1}$ are masses, then $\xi_{\mathrm{pc}1} = m_\mathrm{c}$ and $\xi_{\mathrm{fc}1} = -m_\mathrm{c}$
- Source terms for transport equations for total number concentration:

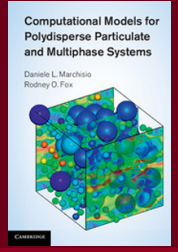
$$\mathcal{S}_\mathrm{N} = \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = J$$

  disperse-phase mass density:

$$\mathcal{S}_\mathrm{M} = \int \xi_{\mathrm{p}1}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c} J$$

  disperse-phase momentum density:

$$\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{p} = \int \xi_{\mathrm{p}1}\mathbf{v}_\mathrm{p}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c}\mathbf{U}_\mathrm{f} J$$



Mesoscale Model for Zero-Order Point Processes

- In zero-order point processes $\mathcal{S}_1$ does not depend on NDF and usually represents rate of formation (i.e., nucleation or spinodal decomposition) of disperse phase:

$$\mathcal{S}_1 = J(\boldsymbol{\phi}_\mathrm{f})\delta(\mathbf{v}_\mathrm{p} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{p} - \boldsymbol{\xi}_\mathrm{pc})\delta(\mathbf{v}_\mathrm{f} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f} - \boldsymbol{\xi}_\mathrm{fc})$$

  where $J(\boldsymbol{\phi}_\mathrm{f}) \geq 0$ is rate of formation of particles with properties $\boldsymbol{\xi}_\mathrm{pc}$
- Definitions consistent with mass balance: for example if $\xi_{\mathrm{p}1}$ and $\xi_{\mathrm{f}1}$ are masses, then $\xi_{\mathrm{pc}1} = m_\mathrm{c}$ and $\xi_{\mathrm{fc}1} = -m_\mathrm{c}$
- Source terms for transport equations for total number concentration:

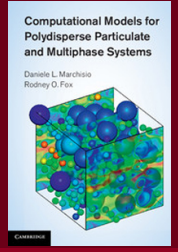
$$\mathcal{S}_\mathrm{N} = \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = J$$

  disperse-phase mass density:

$$\mathcal{S}_\mathrm{M} = \int \xi_{\mathrm{p}1}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c} J$$

  disperse-phase momentum density:

$$\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{p} = \int \xi_{\mathrm{p}1}\mathbf{v}_\mathrm{p}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c}\mathbf{U}_\mathrm{f} J$$



Mesoscale Model for Zero-Order Point Processes

- In zero-order point processes $\mathcal{S}_1$ does not depend on NDF and usually represents rate of formation (i.e., nucleation or spinodal decomposition) of disperse phase:

$$\mathcal{S}_1 = J(\boldsymbol{\phi}_\mathrm{f})\delta(\mathbf{v}_\mathrm{p} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{p} - \boldsymbol{\xi}_\mathrm{pc})\delta(\mathbf{v}_\mathrm{f} - \mathbf{U}_\mathrm{f})\delta(\boldsymbol{\xi}_\mathrm{f} - \boldsymbol{\xi}_\mathrm{fc})$$

  where $J(\boldsymbol{\phi}_\mathrm{f}) \geq 0$ is rate of formation of particles with properties $\boldsymbol{\xi}_\mathrm{pc}$
- Definitions consistent with mass balance: for example if $\xi_{\mathrm{p}1}$ and $\xi_{\mathrm{f}1}$ are masses, then $\xi_{\mathrm{pc}1} = m_\mathrm{c}$ and $\xi_{\mathrm{fc}1} = -m_\mathrm{c}$
- Source terms for transport equations for total number concentration:

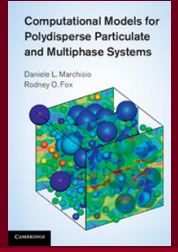
$$\mathcal{S}_\mathrm{N} = \int \mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = J$$

  disperse-phase mass density:

$$\mathcal{S}_\mathrm{M} = \int \xi_{\mathrm{p}1}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c} J$$

  disperse-phase momentum density:

$$\varrho_\mathrm{p}[\![\mathbf{S}]\!]_\mathrm{p} = \int \xi_{\mathrm{p}1}\mathbf{v}_\mathrm{p}\mathcal{S}_1 \, \mathrm{d}\mathbf{v}_\mathrm{p} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{p} \, \mathrm{d}\mathbf{v}_\mathrm{f} \, \mathrm{d}\boldsymbol{\xi}_\mathrm{f} = m_\mathrm{c}\mathbf{U}_\mathrm{f} J$$



Mesoscale Model for First-Order Point Processes

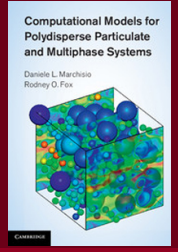
- In first-order point processes $\mathcal{S}_1$ depends linearly on NDF: used to describe particle deposition and filtration, particle breakage, coalescence in foams and dense-packed systems
- Source term in GPBE is usually written as:

$$\mathcal{S}_1 = h^+ - h^-$$

where h^+ is rate of production and h^- is rate of loss of particles due to first-order point processes

- First-order processes are generally quantified by $b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})\,\mathrm{d}t$, which is probability that a particle with velocity $\mathbf{v}_\mathrm{p}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{p}$ in a fluid with velocity $\mathbf{v}_\mathrm{f}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{f}$ undergoes process under investigation in infinitesimal time interval $\mathrm{d}t$
- Quantity b is frequency of process and has units of inverse of time and therefore:

$$h^-(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) = b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) n(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$$



Mesoscale Model for First-Order Point Processes

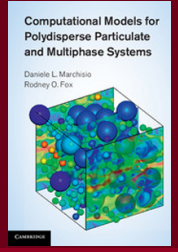
- In first-order point processes $\mathcal{S}_1$ depends linearly on NDF: used to describe particle deposition and filtration, particle breakage, coalescence in foams and dense-packed systems
- Source term in GPBE is usually written as:

$$\mathcal{S}_1 = h^+ - h^-$$

where h^+ is rate of production and h^- is rate of loss of particles due to first-order point processes

- First-order processes are generally quantified by $b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})\,\mathrm{d}t$, which is probability that a particle with velocity $\mathbf{v}_\mathrm{p}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{p}$ in a fluid with velocity $\mathbf{v}_\mathrm{f}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{f}$ undergoes process under investigation in infinitesimal time interval $\mathrm{d}t$
- Quantity b is frequency of process and has units of inverse of time and therefore:

$$h^-(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) = b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) n(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$$



Mesoscale Model for First-Order Point Processes

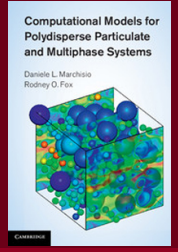
- In first-order point processes $\mathcal{S}_1$ depends linearly on NDF: used to describe particle deposition and filtration, particle breakage, coalescence in foams and dense-packed systems
- Source term in GPBE is usually written as:

$$\mathcal{S}_1 = h^+ - h^-$$

where h^+ is rate of production and h^- is rate of loss of particles due to first-order point processes

- First-order processes are generally quantified by $b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})\,\mathrm{d}t$, which is probability that a particle with velocity $\mathbf{v}_\mathrm{p}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{p}$ in a fluid with velocity $\mathbf{v}_\mathrm{f}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{f}$ undergoes process under investigation in infinitesimal time interval $\mathrm{d}t$
- Quantity b is frequency of process and has units of inverse of time and therefore:

$$h^-(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) = b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) n(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$$



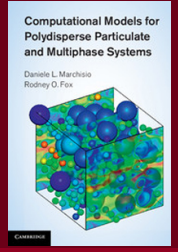
Mesoscale Model for First-Order Point Processes

- In first-order point processes $\mathcal{S}_1$ depends linearly on NDF: used to describe particle deposition and filtration, particle breakage, coalescence in foams and dense-packed systems
- Source term in GPBE is usually written as:

$$\mathcal{S}_1 = h^+ - h^-$$

  where h^+ is rate of production and h^- is rate of loss of particles due to first-order point processes
- First-order processes are generally quantified by $b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})\,\mathrm{d}t$, which is probability that a particle with velocity $\mathbf{v}_\mathrm{p}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{p}$ in a fluid with velocity $\mathbf{v}_\mathrm{f}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{f}$ undergoes process under investigation in infinitesimal time interval $\mathrm{d}t$
- Quantity b is frequency of process and has units of inverse of time and therefore:

$$h^-(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) = b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) n(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$$



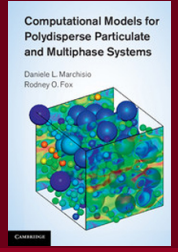
Mesoscale Model for First-Order Point Processes

- In first-order point processes $\mathcal{S}_1$ depends linearly on NDF: used to describe particle deposition and filtration, particle breakage, coalescence in foams and dense-packed systems
- Source term in GPBE is usually written as:

$$\mathcal{S}_1 = h^+ - h^-$$

  where h^+ is rate of production and h^- is rate of loss of particles due to first-order point processes
- First-order processes are generally quantified by $b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})\,\mathrm{d}t$, which is probability that a particle with velocity $\mathbf{v}_\mathrm{p}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{p}$ in a fluid with velocity $\mathbf{v}_\mathrm{f}$ and internal coordinates $\boldsymbol{\xi}_\mathrm{f}$ undergoes process under investigation in infinitesimal time interval $\mathrm{d}t$
- Quantity b is frequency of process and has units of inverse of time and therefore:

$$h^-(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) = b(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) n(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f})$$



Mesoscale Model for First-Order Point Processes

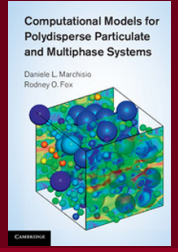
- In order to quantify rate of formation a conditional probability density (PDF) function that states probability of formation of daughter particle with $\mathbf{v}_\mathrm{p}'$ and $\boldsymbol{\xi}_\mathrm{p}'$ from mother particle with $\mathbf{v}_\mathrm{p}$ and $\boldsymbol{\xi}_\mathrm{p}$ has to be formulated: $P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$
- PDF has to satisfy a normalization condition:

$$\int P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = 1$$

- Instead of PDF an equivalent formulation in terms of a corresponding conditional NDF, $N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$, is used:

$$\int N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = \nu(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$$

where ν is number of new particles formed by first-order process



Mesoscale Model for First-Order Point Processes

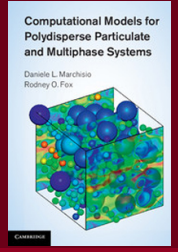
- In order to quantify rate of formation a conditional probability density (PDF) function that states probability of formation of daughter particle with $\mathbf{v}_\mathrm{p}'$ and $\boldsymbol{\xi}_\mathrm{p}'$ from mother particle with $\mathbf{v}_\mathrm{p}$ and $\boldsymbol{\xi}_\mathrm{p}$ has to be formulated: $P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$
- PDF has to satisfy a normalization condition:

$$\int P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = 1$$

- Instead of PDF an equivalent formulation in terms of a corresponding conditional NDF, $N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$, is used:

$$\int N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = \nu(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$$

where ν is number of new particles formed by first-order process



Mesoscale Model for First-Order Point Processes

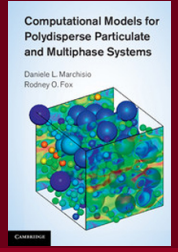
- In order to quantify rate of formation a conditional probability density (PDF) function that states probability of formation of daughter particle with $\mathbf{v}_\mathrm{p}'$ and $\boldsymbol{\xi}_\mathrm{p}'$ from mother particle with $\mathbf{v}_\mathrm{p}$ and $\boldsymbol{\xi}_\mathrm{p}$ has to be formulated: $P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$
- PDF has to satisfy a normalization condition:

$$\int P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = 1$$

- Instead of PDF an equivalent formulation in terms of a corresponding conditional NDF, $N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$, is used:

$$\int N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = \nu(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$$

where ν is number of new particles formed by first-order process



Mesoscale Model for First-Order Point Processes

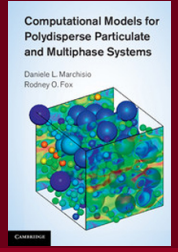
- In order to quantify rate of formation a conditional probability density (PDF) function that states probability of formation of daughter particle with $\mathbf{v}_\mathrm{p}'$ and $\boldsymbol{\xi}_\mathrm{p}'$ from mother particle with $\mathbf{v}_\mathrm{p}$ and $\boldsymbol{\xi}_\mathrm{p}$ has to be formulated: $P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$
- PDF has to satisfy a normalization condition:

$$\int P(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = 1$$

- Instead of PDF an equivalent formulation in terms of a corresponding conditional NDF, $N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$, is used:

$$\int N(\mathbf{v}_\mathrm{p}', \boldsymbol{\xi}_\mathrm{p}' | \mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}) \, \mathrm{d}\mathbf{v}_\mathrm{p}' \mathrm{d}\boldsymbol{\xi}_\mathrm{p}' = \nu(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p})$$

where ν is number of new particles formed by first-order process



Mesoscale Model for First-Order Point Processes

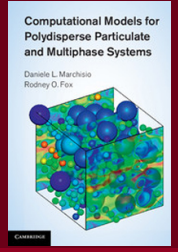
- Rate of formation of new particles due to first-order discontinuous processes can be expressed as

$$h^+(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) = \int N(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p} | \mathbf{v}'_\mathrm{p}, \boldsymbol{\xi}'_\mathrm{p}) b(\mathbf{v}'_\mathrm{p}, \boldsymbol{\xi}'_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) n(\mathbf{v}'_\mathrm{p}, \boldsymbol{\xi}'_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) \, \mathrm{d}\mathbf{v}'_\mathrm{p} \mathrm{d}\boldsymbol{\xi}'_\mathrm{p}$$

- In terms of P

$$h^+(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) = \int \nu(\mathbf{v}'_\mathrm{p}, \boldsymbol{\xi}'_\mathrm{p}) P(\mathbf{v}_\mathrm{p}, \boldsymbol{\xi}_\mathrm{p} | \mathbf{v}'_\mathrm{p}, \boldsymbol{\xi}'_\mathrm{p}) b(\mathbf{v}'_\mathrm{p}, \boldsymbol{\xi}'_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) n(\mathbf{v}'_\mathrm{p}, \boldsymbol{\xi}'_\mathrm{p}, \mathbf{v}_\mathrm{f}, \boldsymbol{\xi}_\mathrm{f}) \, \mathrm{d}\mathbf{v}'_\mathrm{p} \mathrm{d}\boldsymbol{\xi}'_\mathrm{p}$$

- Note that these expressions are written under assumption that fluid velocity and composition do not change during first-order process



Turbulent Breakage Kernel (or Frequency) for Fluid–Fluid Systems

- **Coulaloglou & Tavlarides (1977)**:

$$b = C_1 d_\mathrm{p}^{-2/3} \varepsilon_\mathrm{f}^{1/3} \exp\left(- \frac{C_2 \sigma}{\rho_\mathrm{p} \varepsilon_\mathrm{f}^{2/3} d_\mathrm{p}^{5/3}} \right)$$

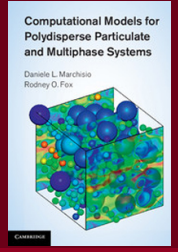
- **Alopaeus et al. (2002)**:

$$b = C_4 \varepsilon_\mathrm{f}^{1/3} \mathrm{erfc}\left(\sqrt{C_5 \frac{\sigma}{\rho_\mathrm{f} \varepsilon_\mathrm{f}^{2/3} d_\mathrm{p}^{5/3}} + C_6 \frac{\mu_\mathrm{f}}{\sqrt{\rho_\mathrm{f} \rho_\mathrm{p}} \varepsilon_\mathrm{f}^{1/3} d_\mathrm{p}^{4/3}}} \right)$$

- **Luo & Svendsen (1996)**:

$$bN = 0.92 \left(\frac{\varepsilon_\mathrm{f}}{{d'_\mathrm{p}}^2} \right)^{1/3} \int_{\zeta_\mathrm{min}}^1 \frac{(1+\zeta)^2}{\zeta^{11/3}} \exp\left(- \frac{12 c_\mathrm{f} \sigma}{\beta \rho_\mathrm{p} \varepsilon_\mathrm{f}^{2/3} {d'_\mathrm{p}}^{5/3} \zeta^{11/3}} \right) \mathrm{d}\zeta$$

where $\zeta = \eta_\mathrm{k}/d'_\mathrm{p}$, $\beta = 3/2$, $c_\mathrm{f} = f^{2/3} + (1-f)^{2/3} - 1]$, $f = (d_\mathrm{p}/d'_\mathrm{p})^3$, d_p is size of daughter particle, and d'_p is size of parent particle

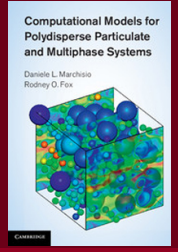


Turbulent Breakage Kernel (or Frequency) for Fluid–Particle Systems

- Particles larger than turbulent microscale: breakage caused by instantaneous normal stresses
- **Particles smaller than turbulent microscale**: breakage caused by shear stresses due to turbulent dynamic velocity differences:

$$b = \frac{1}{\sqrt{15}} \left(\frac{\varepsilon_\mathrm{f}}{\nu_\mathrm{f}} \right)^{1/2} \exp\left(- \frac{\tau_\mathrm{s}}{\mu_\mathrm{f} (\varepsilon_\mathrm{f}/\nu_\mathrm{f})^{1/2}} \right)$$

- Aggregate strength: $\tau_\mathrm{s} = 9 k_\mathrm{co} \phi_\mathrm{p} F / (8 \pi d_\mathrm{o})$ where d_o diameter of primary particle, k_co coordination number, F inter-particle force can be computed as $F = A_\mathrm{H} d_\mathrm{o} / (12 H_\mathrm{o}^2)$
- Coordination number depends on aggregate structure: $k_\mathrm{c} \approx 15 \phi_\mathrm{p}^{1.2}$ where ϕ_p is volume fraction of solid within aggregates $\phi_\mathrm{p} = (0.414 D_\mathrm{f} - 0.211)(d_\mathrm{p}/d_\mathrm{o})^{D_\mathrm{f}-3}$



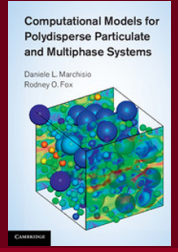
Daughter Distribution Function

- Simplest functional form is summation of delta functions (ν: number of daughters; $\xi'_{\rm p}$: parent particle; $\xi_{\rm p}$: daughter particle; $\xi_i(\xi'_{\rm p})$ function stating relationship between $i^{\rm th}$ daughter and parent):

$$N(\xi_{\rm p}|\xi'_{\rm p}) = \sum_{i=1}^{\nu} \delta\left[\xi_{\rm p} - \xi_i(\xi'_{\rm p})\right]$$

- $\xi_i(\xi'_{\rm p})$ is formulated by respecting additional constraints (conservation of mass!)
- Binary breakage ($\nu = 2$): two identical fragments $\rightarrow$ symmetric breakage; very unequal fragments: erosion
- Often diversity of particles induces many simultaneous phenomena and continuous distributions are used
- Uniform distribution (particle mass internal coordinate)

$$N(m|m') = \begin{cases} 0 & m < 0 \\ \frac{\nu}{m'} & 0 \le m \le m' \\ 0 & m > m' \end{cases}$$



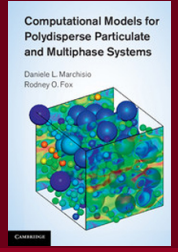
Daughter Distribution Function

- Simplest functional form is summation of delta functions (ν: number of daughters; $\boldsymbol{\xi}'_{\mathrm{p}}$: parent particle; $\boldsymbol{\xi}_{\mathrm{p}}$: daughter particle; $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ function stating relationship between i^{th} daughter and parent):

$$N(\boldsymbol{\xi}_{\mathrm{p}}|\boldsymbol{\xi}'_{\mathrm{p}}) = \sum_{i=1}^{\nu} \delta\left[\boldsymbol{\xi}_{\mathrm{p}} - \boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})\right]$$

- $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ is formulated by respecting additional constraints (conservation of mass!)
- Binary breakage ($\nu = 2$): two identical fragments $\rightarrow$ symmetric breakage; very unequal fragments: erosion
- Often diversity of particles induces many simultaneous phenomena and continuous distributions are used
- Uniform distribution (particle mass internal coordinate)

$$N(m|m') = \begin{cases} 0 & m < 0 \\ \frac{\nu}{m'} & 0 \le m \le m' \\ 0 & m > m' \end{cases}$$



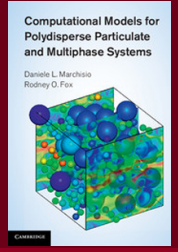
Daughter Distribution Function

- Simplest functional form is summation of delta functions (ν: number of daughters; $\boldsymbol{\xi}'_{\mathrm{p}}$: parent particle; $\boldsymbol{\xi}_{\mathrm{p}}$: daughter particle; $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ function stating relationship between i^{th} daughter and parent):

$$N(\boldsymbol{\xi}_{\mathrm{p}}|\boldsymbol{\xi}'_{\mathrm{p}}) = \sum_{i=1}^{\nu} \delta\left[\boldsymbol{\xi}_{\mathrm{p}} - \boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})\right]$$

- $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ is formulated by respecting additional constraints (conservation of mass!)
- Binary breakage ($\nu = 2$): two identical fragments $\rightarrow$ symmetric breakage; very unequal fragments: erosion
- Often diversity of particles induces many simultaneous phenomena and continuous distributions are used
- Uniform distribution (particle mass internal coordinate)

$$N(m|m') = \begin{cases} 0 & m < 0 \\ \frac{\nu}{m'} & 0 \le m \le m' \\ 0 & m > m' \end{cases}$$



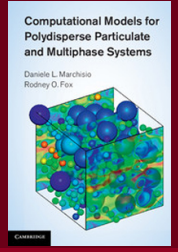
Daughter Distribution Function

- Simplest functional form is summation of delta functions (ν: number of daughters; $\boldsymbol{\xi}'_{\mathrm{p}}$: parent particle; $\boldsymbol{\xi}_{\mathrm{p}}$: daughter particle; $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ function stating relationship between i^{th} daughter and parent):

$$N(\boldsymbol{\xi}_{\mathrm{p}}|\boldsymbol{\xi}'_{\mathrm{p}}) = \sum_{i=1}^{\nu} \delta\left[\boldsymbol{\xi}_{\mathrm{p}} - \boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})\right]$$

- $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ is formulated by respecting additional constraints (conservation of mass!)
- Binary breakage ($\nu = 2$): two identical fragments → symmetric breakage; very unequal fragments: erosion
- Often diversity of particles induces many simultaneous phenomena and continuous distributions are used
- Uniform distribution (particle mass internal coordinate)

$$N(m|m') = \begin{cases} 0 & m < 0 \\ \frac{\nu}{m'} & 0 \le m \le m' \\ 0 & m > m' \end{cases}$$



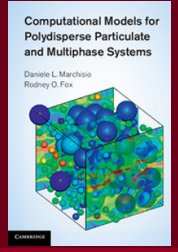
Daughter Distribution Function

- Simplest functional form is summation of delta functions (ν: number of daughters; $\boldsymbol{\xi}'_{\mathrm{p}}$: parent particle; $\boldsymbol{\xi}_{\mathrm{p}}$: daughter particle; $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ function stating relationship between i^{th} daughter and parent):

$$N(\boldsymbol{\xi}_{\mathrm{p}}|\boldsymbol{\xi}'_{\mathrm{p}}) = \sum_{i=1}^{\nu} \delta\left[\boldsymbol{\xi}_{\mathrm{p}} - \boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})\right]$$

- $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ is formulated by respecting additional constraints (conservation of mass!)
- Binary breakage ($\nu = 2$): two identical fragments → symmetric breakage; very unequal fragments: erosion
- Often diversity of particles induces many simultaneous phenomena and continuous distributions are used
- Uniform distribution (particle mass internal coordinate)

$$N(m|m') = \begin{cases} 0 & m < 0 \\ \frac{\nu}{m'} & 0 \le m \le m' \\ 0 & m > m' \end{cases}$$



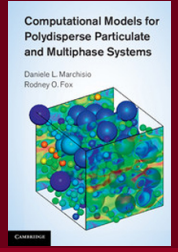
Daughter Distribution Function

- Simplest functional form is summation of delta functions (ν: number of daughters; $\boldsymbol{\xi}'_{\mathrm{p}}$: parent particle; $\boldsymbol{\xi}_{\mathrm{p}}$: daughter particle; $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ function stating relationship between i^{th} daughter and parent):

$$N(\boldsymbol{\xi}_{\mathrm{p}}|\boldsymbol{\xi}'_{\mathrm{p}}) = \sum_{i=1}^{\nu} \delta\left[\boldsymbol{\xi}_{\mathrm{p}} - \boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})\right]$$

- $\boldsymbol{\xi}_i(\boldsymbol{\xi}'_{\mathrm{p}})$ is formulated by respecting additional constraints (conservation of mass!)
- Binary breakage ($\nu = 2$): two identical fragments $\rightarrow$ symmetric breakage; very unequal fragments: erosion
- Often diversity of particles induces many simultaneous phenomena and continuous distributions are used
- Uniform distribution (particle mass internal coordinate)

$$N(m|m') = \begin{cases} 0 & m < 0 \\ \frac{\nu}{m'} & 0 \le m \le m' \\ 0 & m > m' \end{cases}$$



Daughter Distribution Function

- By imposing that mass of formed fragments sum up to mass of parent particle

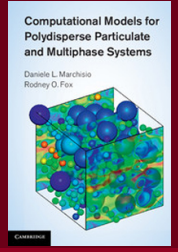
$$\int_0^{m'} mN(m|m')\,\mathrm{d}m = \frac{\nu m'}{2}$$

  it is easy to show that such a function is consistent only for binary breakage (i.e., $\nu = 2$)

- Binary parabolic distribution $(0 \le C \le 3)$

$$N(m|m') = \begin{cases} 0 & m < 0 \\ C + \left(1 - \frac{C}{2}\right)\left[24\left(\frac{m}{m'}\right)^2 - 24\left(\frac{m}{m'}\right) + 6\right] & 0 \le m \le m' \\ 0 & m > m' \end{cases}$$

- For $0 \le C < 2$ formation of different-sized fragments (i.e., erosion) is likely, whereas for $2 < C \le 3$ symmetric breakage is more probable
- $C = 2$ recovers uniform distribution



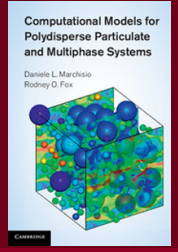
Daughter Distribution Function

- Distribution can also be written in terms of particle size d_p: $N(d_\mathrm{p}|d'_\mathrm{p}) = N(m|m')\partial m/\partial d_\mathrm{p}$ where $\partial m/\partial d_\mathrm{p} = 3\rho_\mathrm{p} k_\mathrm{V} d_\mathrm{p}^2$
- Formation of three fragments (Coulaloglou & Tavlarides, 1977)

$$N(d_\mathrm{p}|d'_\mathrm{p}) = \frac{135 d_\mathrm{p}^2}{{d'_\mathrm{p}}^3 \sqrt{2\pi}} \exp\left(-\frac{25}{2}\left[3\left(\frac{d_\mathrm{p}}{d'_\mathrm{p}}\right)^3\right]^2\right)$$

- β-distribution (ν number of fragments formed; q greater than one: formation of ν equally-sized fragments; q smaller than one: formation of one large fragment with $\nu - 1$ satellite fragments)

$$N(d_\mathrm{p}|d'_\mathrm{p}) = \frac{\Gamma(q\nu)}{\Gamma(q)\Gamma(q(\nu-1))}\left[1-\left(\frac{d_\mathrm{p}}{d'_\mathrm{p}}\right)^3\right]^{q(\nu-1)-1}\left(\frac{d_\mathrm{p}}{d'_\mathrm{p}}\right)^{3q-1}\frac{3\nu}{d'_\mathrm{p}}$$



Daughter Distribution Function

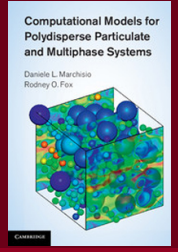
- In case of multiple internal coordinates (particle size, d_p, and composition, ϕ)

$$N(d_\mathrm{p},\phi|d'_\mathrm{p},\phi') = 180\frac{d_\mathrm{p}^2}{{d'_\mathrm{p}}^3}\left(\frac{d_\mathrm{p}^3}{{d'_\mathrm{p}}^3}\right)^2\left(1-\frac{d_\mathrm{p}^3}{{d'_\mathrm{p}}^3}\right)^2\delta\left(\phi-\frac{d_\mathrm{p}^3}{{d'_\mathrm{p}}^3}\phi'\right)$$

  where d'_p and ϕ' represent properties of parent particle, while d_p and ϕ properties of daughter particles

- Fragmentation of fractal aggregates (particle volume, V_p, and surface area, A_p)

$$P(V_\mathrm{p},A_\mathrm{p}|V'_\mathrm{p},A'_\mathrm{p}) = 2\delta\left(V_\mathrm{p}-\frac{V'_\mathrm{p}}{2}\right)\delta\left(A_\mathrm{p}-\frac{A'_\mathrm{p}}{2}\right)$$



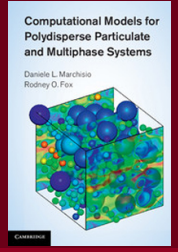
Mesoscale Model for Second-Order Point Processes

- Second-order point processes involve two particles (i.e. coalescence, aggregation, collision) and their rate shows a quadratic dependence on NDF
- Source term in GPBE:

$$S_1 = h^+ - h^-$$

where h^+ is rate of production and h^- is rate of loss of particles

- One particle is located in $\mathbf{x}'$ and is characterized by phase-space vector $\boldsymbol{\eta}'$; second particle is located in $\tilde{\mathbf{x}}$ and has phase-space vector $\tilde{\boldsymbol{\eta}}$
- Frequency of second-order point process: $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ (symmetric with respect to permutation of particles)
- Quantity $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}t$ represents fraction of particles undergoing point process in time interval $\mathrm{d}t$



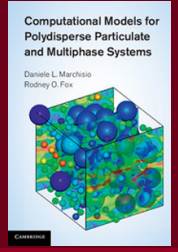
Mesoscale Model for Second-Order Point Processes

- Second-order point processes involve two particles (i.e. coalescence, aggregation, collision) and their rate shows a quadratic dependence on NDF
- Source term in GPBE:

$$S_1 = h^+ - h^-$$

where h^+ is rate of production and h^- is rate of loss of particles

- One particle is located in $\mathbf{x}'$ and is characterized by phase-space vector $\boldsymbol{\eta}'$; second particle is located in $\tilde{\mathbf{x}}$ and has phase-space vector $\tilde{\boldsymbol{\eta}}$
- Frequency of second-order point process: $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ (symmetric with respect to permutation of particles)
- Quantity $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}t$ represents fraction of particles undergoing point process in time interval $\mathrm{d}t$



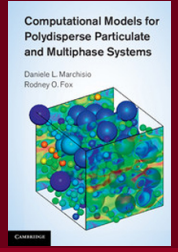
Mesoscale Model for Second-Order Point Processes

- Second-order point processes involve two particles (i.e. coalescence, aggregation, collision) and their rate shows a quadratic dependence on NDF
- Source term in GPBE:

$$\mathcal{S}_1 = h^+ - h^-$$

where h^+ is rate of production and h^- is rate of loss of particles

- One particle is located in $\mathbf{x}'$ and is characterized by phase-space vector $\boldsymbol{\eta}'$; second particle is located in $\tilde{\mathbf{x}}$ and has phase-space vector $\tilde{\boldsymbol{\eta}}$
- Frequency of second-order point process: $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ (symmetric with respect to permutation of particles)
- Quantity $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\,\mathrm{d}t$ represents fraction of particles undergoing point process in time interval $\mathrm{d}t$

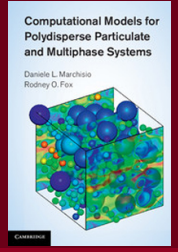


Mesoscale Model for Second-Order Point Processes

- Second-order point processes involve two particles (i.e. coalescence, aggregation, collision) and their rate shows a quadratic dependence on NDF
- Source term in GPBE:

$$\mathcal{S}_1 = h^+ - h^-$$

  where h^+ is rate of production and h^- is rate of loss of particles
- One particle is located in $\mathbf{x}'$ and is characterized by phase-space vector $\boldsymbol{\eta}'$; second particle is located in $\tilde{\mathbf{x}}$ and has phase-space vector $\tilde{\boldsymbol{\eta}}$
- Frequency of second-order point process: $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ (symmetric with respect to permutation of particles)
- Quantity $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\,\mathrm{d}t$ represents fraction of particles undergoing point process in time interval $\mathrm{d}t$

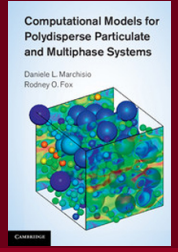


Mesoscale Model for Second-Order Point Processes

- Second-order point processes involve two particles (i.e. coalescence, aggregation, collision) and their rate shows a quadratic dependence on NDF
- Source term in GPBE:

$$\mathcal{S}_1 = h^+ - h^-$$

  where h^+ is rate of production and h^- is rate of loss of particles
- One particle is located in $\mathbf{x}'$ and is characterized by phase-space vector $\boldsymbol{\eta}'$; second particle is located in $\tilde{\mathbf{x}}$ and has phase-space vector $\tilde{\boldsymbol{\eta}}$
- Frequency of second-order point process: $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ (symmetric with respect to permutation of particles)
- Quantity $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\,\mathrm{d}t$ represents fraction of particles undergoing point process in time interval $\mathrm{d}t$

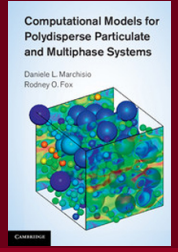


Mesoscale Model for Second-Order Point Processes

- Second-order point processes involve two particles (i.e. coalescence, aggregation, collision) and their rate shows a quadratic dependence on NDF
- Source term in GPBE:

$$\mathcal{S}_1 = h^+ - h^-$$

  where h^+ is rate of production and h^- is rate of loss of particles
- One particle is located in $\mathbf{x}'$ and is characterized by phase-space vector $\boldsymbol{\eta}'$; second particle is located in $\tilde{\mathbf{x}}$ and has phase-space vector $\tilde{\boldsymbol{\eta}}$
- Frequency of second-order point process: $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ (symmetric with respect to permutation of particles)
- Quantity $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\,\mathrm{d}t$ represents fraction of particles undergoing point process in time interval $\mathrm{d}t$

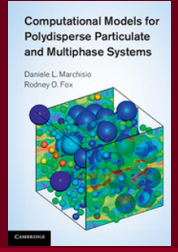


Mesoscale Model for Second-Order Point Processes

- Unknown pair number density function: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$ represents expected number of particle pairs with $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Closure is often used: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Based on these definitions, calculate number of events per unit time involving particle pairs $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ as $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$
- Total number of events occurring per unit time and unit volume involving *test* particles located near $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$:

$$N_{\mathrm{e}}(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'$$

  where integrals are over all locations $\mathbf{x}'$ and phase-space variables $\boldsymbol{\eta}'$ of *field* particle
- In most practical cases a closure is invoked: $n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$

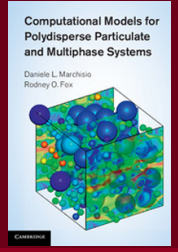


Mesoscale Model for Second-Order Point Processes

- Unknown pair number density function: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$ represents expected number of particle pairs with $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Closure is often used: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Based on these definitions, calculate number of events per unit time involving particle pairs $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ as $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$
- Total number of events occurring per unit time and unit volume involving *test* particles located near $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$:

$$N_{\mathrm{e}}(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'$$

  where integrals are over all locations $\mathbf{x}'$ and phase-space variables $\boldsymbol{\eta}'$ of *field* particle
- In most practical cases a closure is invoked: $n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$

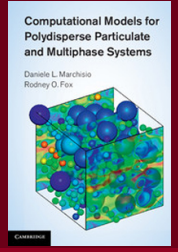


Mesoscale Model for Second-Order Point Processes

- Unknown pair number density function: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$ represents expected number of particle pairs with $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Closure is often used: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Based on these definitions, calculate number of events per unit time involving particle pairs $(\mathbf{x}', \eta')$ and $(\tilde{\mathbf{x}}, \tilde{\eta})$ as $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$
- Total number of events occurring per unit time and unit volume involving *test* particles located near $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$:

$$N_{\mathrm{e}}(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'$$

  where integrals are over all locations $\mathbf{x}'$ and phase-space variables $\boldsymbol{\eta}'$ of *field* particle
- In most practical cases a closure is invoked: $n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$

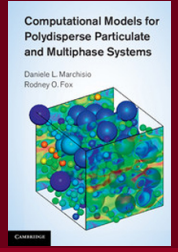


Mesoscale Model for Second-Order Point Processes

- Unknown pair number density function: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$ represents expected number of particle pairs with $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Closure is often used: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Based on these definitions, calculate number of events per unit time involving particle pairs $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ as $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'\mathrm{d}\tilde{\mathbf{x}}\mathrm{d}\tilde{\boldsymbol{\eta}}$
- Total number of events occurring per unit time and unit volume involving *test* particles located near $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$:

$$N_{\mathrm{e}}(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}'\mathrm{d}\boldsymbol{\eta}'$$

  where integrals are over all locations $\mathbf{x}'$ and phase-space variables $\boldsymbol{\eta}'$ of *field* particle
- In most practical cases a closure is invoked: $n(t, \mathbf{x}', \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}')n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$

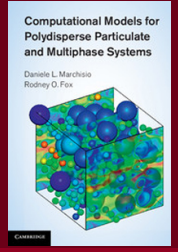


Mesoscale Model for Second-Order Point Processes

- Unknown pair number density function: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}' \mathrm{d}\boldsymbol{\eta}' \mathrm{d}\tilde{\mathbf{x}} \mathrm{d}\tilde{\boldsymbol{\eta}}$ represents expected number of particle pairs with $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Closure is often used: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Based on these definitions, calculate number of events per unit time involving particle pairs $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ as $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}' \mathrm{d}\boldsymbol{\eta}' \mathrm{d}\tilde{\mathbf{x}} \mathrm{d}\tilde{\boldsymbol{\eta}}$
- Total number of events occurring per unit time and unit volume involving *test* particles located near $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$:

$$N_{\mathrm{e}}(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}' \mathrm{d}\boldsymbol{\eta}'$$

  where integrals are over all locations $\mathbf{x}'$ and phase-space variables $\boldsymbol{\eta}'$ of *field* particle
- In most practical cases a closure is invoked: $n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$

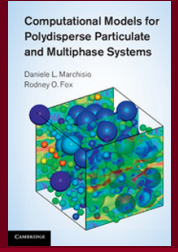


Mesoscale Model for Second-Order Point Processes

- Unknown pair number density function: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}' \mathrm{d}\boldsymbol{\eta}' \mathrm{d}\tilde{\mathbf{x}} \mathrm{d}\tilde{\boldsymbol{\eta}}$ represents expected number of particle pairs with $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Closure is often used: $n^{(2)}(t, \mathbf{x}', \boldsymbol{\eta}', \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$
- Based on these definitions, calculate number of events per unit time involving particle pairs $(\mathbf{x}', \boldsymbol{\eta}')$ and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$ as $a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}' \mathrm{d}\boldsymbol{\eta}' \mathrm{d}\tilde{\mathbf{x}} \mathrm{d}\tilde{\boldsymbol{\eta}}$
- Total number of events occurring per unit time and unit volume involving *test* particles located near $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$:

$$N_{\mathrm{e}}(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})\, \mathrm{d}\mathbf{x}' \mathrm{d}\boldsymbol{\eta}'$$

  where integrals are over all locations $\mathbf{x}'$ and phase-space variables $\boldsymbol{\eta}'$ of *field* particle
- In most practical cases a closure is invoked: $n(t, \mathbf{x}', \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \approx n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}})$



Mesoscale Model for Second-Order Point Processes

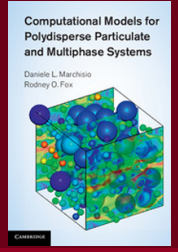
- By using this simplified expression and by inverting order of integrals

$$N_e(t,\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}}) = \int \left(\int a(\mathbf{x}',\boldsymbol{\eta}';\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\mathbf{x}' \right) n(t,\tilde{\mathbf{x}},\boldsymbol{\eta}')n(t,\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\boldsymbol{\eta}'$$

- Integral appearing between parenthesis is called **kernel**

$$\beta(\tilde{\mathbf{x}};\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}',\boldsymbol{\eta}';\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\mathbf{x}'$$

- Kernels have dimensions of spatial volume per unit time (volumetric flux of particles undergoing second-order point process)
- Integral is used to derive kernels for different physical processes
- Particle encounters are dominated by local transport phenomena at distances larger than particle size and interaction potentials at shorter distances
- Kernels are naturally split in two terms: one accounts for particle transport in region of interaction (**collision kernel**) and other accounts for short-range interactions (**aggregation efficiency**)



Mesoscale Model for Second-Order Point Processes

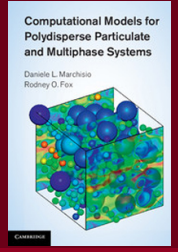
- By using this simplified expression and by inverting order of integrals

$$N_e(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int \left(\int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}' \right) n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\boldsymbol{\eta}'$$

- Integral appearing between parenthesis is called **kernel**

$$\beta(\tilde{\mathbf{x}}; \boldsymbol{\eta}', \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}'$$

- Kernels have dimensions of spatial volume per unit time (volumetric flux of particles undergoing second-order point process)
- Integral is used to derive kernels for different physical processes
- Particle encounters are dominated by local transport phenomena at distances larger than particle size and interaction potentials at shorter distances
- Kernels are naturally split in two terms: one accounts for particle transport in region of interaction (**collision kernel**) and other accounts for short-range interactions (**aggregation efficiency**)



Mesoscale Model for Second-Order Point Processes

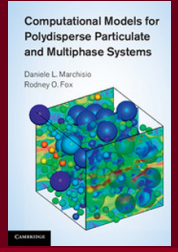
- By using this simplified expression and by inverting order of integrals

$$N_e(t,\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}}) = \int \left(\int a(\mathbf{x}',\boldsymbol{\eta}';\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\mathbf{x}' \right) n(t,\tilde{\mathbf{x}},\boldsymbol{\eta}')n(t,\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\boldsymbol{\eta}'$$

- Integral appearing between parenthesis is called **kernel**

$$\beta(\tilde{\mathbf{x}};\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}',\boldsymbol{\eta}';\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\mathbf{x}'$$

- Kernels have dimensions of spatial volume per unit time (volumetric flux of particles undergoing second-order point process)
- Integral is used to derive kernels for different physical processes
- Particle encounters are dominated by local transport phenomena at distances larger than particle size and interaction potentials at shorter distances
- Kernels are naturally split in two terms: one accounts for particle transport in region of interaction (**collision kernel**) and other accounts for short-range interactions (**aggregation efficiency**)



Mesoscale Model for Second-Order Point Processes

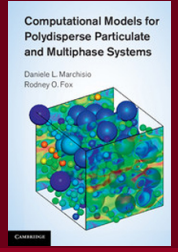
- By using this simplified expression and by inverting order of integrals

$$N_e(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int \left(\int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}' \right) n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\boldsymbol{\eta}'$$

- Integral appearing between parenthesis is called **kernel**

$$\beta(\tilde{\mathbf{x}}; \boldsymbol{\eta}', \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}'$$

- Kernels have dimensions of spatial volume per unit time (volumetric flux of particles undergoing second-order point process)
- Integral is used to derive kernels for different physical processes
- Particle encounters are dominated by local transport phenomena at distances larger than particle size and interaction potentials at shorter distances
- Kernels are naturally split in two terms: one accounts for particle transport in region of interaction (**collision kernel**) and other accounts for short-range interactions (**aggregation efficiency**)



Mesoscale Model for Second-Order Point Processes

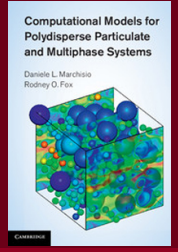
- By using this simplified expression and by inverting order of integrals

$$N_e(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int \left(\int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}' \right) n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\boldsymbol{\eta}'$$

- Integral appearing between parenthesis is called **kernel**

$$\beta(\tilde{\mathbf{x}}; \boldsymbol{\eta}', \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}'$$

- Kernels have dimensions of spatial volume per unit time (volumetric flux of particles undergoing second-order point process)
- Integral is used to derive kernels for different physical processes
- Particle encounters are dominated by local transport phenomena at distances larger than particle size and interaction potentials at shorter distances
- Kernels are naturally split in two terms: one accounts for particle transport in region of interaction (**collision kernel**) and other accounts for short-range interactions (**aggregation efficiency**)



Mesoscale Model for Second-Order Point Processes

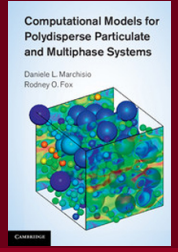
- By using this simplified expression and by inverting order of integrals

$$N_e(t,\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}}) = \int \left(\int a(\mathbf{x}',\boldsymbol{\eta}';\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\mathbf{x}' \right) n(t,\tilde{\mathbf{x}},\boldsymbol{\eta}')n(t,\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\boldsymbol{\eta}'$$

- Integral appearing between parenthesis is called **kernel**

$$\beta(\tilde{\mathbf{x}};\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}',\boldsymbol{\eta}';\tilde{\mathbf{x}},\tilde{\boldsymbol{\eta}})\,\mathrm{d}\mathbf{x}'$$

- Kernels have dimensions of spatial volume per unit time (volumetric flux of particles undergoing second-order point process)
- Integral is used to derive kernels for different physical processes
- Particle encounters are dominated by local transport phenomena at distances larger than particle size and interaction potentials at shorter distances
- Kernels are naturally split in two terms: one accounts for particle transport in region of interaction (**collision kernel**) and other accounts for short-range interactions (**aggregation efficiency**)



Mesoscale Model for Second-Order Point Processes

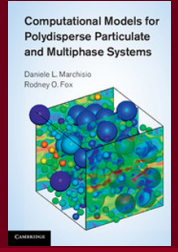
- By using this simplified expression and by inverting order of integrals

$$N_e(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) = \int \left(\int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}' \right) n(t, \tilde{\mathbf{x}}, \boldsymbol{\eta}') n(t, \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\boldsymbol{\eta}'$$

- Integral appearing between parenthesis is called **kernel**

$$\beta(\tilde{\mathbf{x}}; \boldsymbol{\eta}', \tilde{\boldsymbol{\eta}}) = \int a(\mathbf{x}', \boldsymbol{\eta}'; \tilde{\mathbf{x}}, \tilde{\boldsymbol{\eta}}) \, \mathrm{d}\mathbf{x}'$$

- Kernels have dimensions of spatial volume per unit time (volumetric flux of particles undergoing second-order point process)
- Integral is used to derive kernels for different physical processes
- Particle encounters are dominated by local transport phenomena at distances larger than particle size and interaction potentials at shorter distances
- Kernels are naturally split in two terms: one accounts for particle transport in region of interaction (**collision kernel**) and other accounts for short-range interactions (**aggregation efficiency**)



Mesoscale Model for Second-Order Point Processes

- Positive source term is

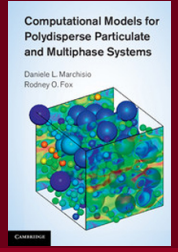
$$h^+(t,\mathbf{x},\boldsymbol{\eta}) = \frac{1}{\delta}\int \beta(\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\tilde{\boldsymbol{\eta}})J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$

- δ is symmetry factor (two for identical particles) and $J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})$ is Jacobian of variable transformation (with $\boldsymbol{\eta}'$ held constant)

$$J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta}) = \left|\frac{\partial\tilde{\boldsymbol{\eta}}}{\partial\boldsymbol{\eta}}\right| = \begin{vmatrix} \frac{\partial\tilde{\eta}_1}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_1}{\partial\eta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial\tilde{\eta}_M}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_M}{\partial\eta_M} \end{vmatrix}$$

- Relationship between $\tilde{\boldsymbol{\eta}}$, $\boldsymbol{\eta}'$ and $\boldsymbol{\eta}$ is derived by continuity statements
- Negative source term is

$$h^-(t,\mathbf{x},\boldsymbol{\eta}) = \int \beta(\boldsymbol{\eta}',\boldsymbol{\eta})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$



Mesoscale Model for Second-Order Point Processes

- Positive source term is

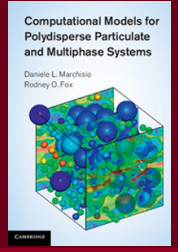
$$h^+(t,\mathbf{x},\boldsymbol{\eta}) = \frac{1}{\delta}\int \beta(\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\tilde{\boldsymbol{\eta}})J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$

- δ is symmetry factor (two for identical particles) and $J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})$ is Jacobian of variable transformation (with $\boldsymbol{\eta}'$ held constant)

$$J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta}) = \left|\frac{\partial\tilde{\boldsymbol{\eta}}}{\partial\boldsymbol{\eta}}\right| = \begin{vmatrix} \frac{\partial\tilde{\eta}_1}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_1}{\partial\eta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial\tilde{\eta}_M}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_M}{\partial\eta_M} \end{vmatrix}$$

- Relationship between $\tilde{\boldsymbol{\eta}}$, $\boldsymbol{\eta}'$ and $\boldsymbol{\eta}$ is derived by continuity statements
- Negative source term is

$$h^-(t,\mathbf{x},\boldsymbol{\eta}) = \int \beta(\boldsymbol{\eta}',\boldsymbol{\eta})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$



Mesoscale Model for Second-Order Point Processes

- Positive source term is

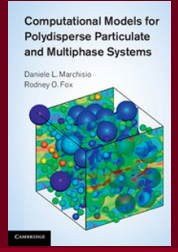
$$h^+(t,\mathbf{x},\boldsymbol{\eta}) = \frac{1}{\delta}\int \beta(\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\tilde{\boldsymbol{\eta}})J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$

- δ is symmetry factor (two for identical particles) and $J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})$ is Jacobian of variable transformation (with $\boldsymbol{\eta}'$ held constant)

$$J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta}) = \left|\frac{\partial\tilde{\boldsymbol{\eta}}}{\partial\boldsymbol{\eta}}\right| = \begin{vmatrix} \frac{\partial\tilde{\eta}_1}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_1}{\partial\eta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial\tilde{\eta}_M}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_M}{\partial\eta_M} \end{vmatrix}$$

- Relationship between $\tilde{\eta}$, η' and η is derived by continuity statements
- Negative source term is

$$h^-(t,\mathbf{x},\boldsymbol{\eta}) = \int \beta(\boldsymbol{\eta}',\boldsymbol{\eta})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$



Mesoscale Model for Second-Order Point Processes

- Positive source term is

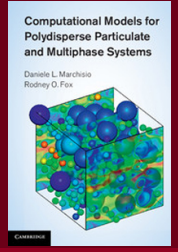
$$h^+(t,\mathbf{x},\boldsymbol{\eta}) = \frac{1}{\delta}\int \beta(\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\tilde{\boldsymbol{\eta}})J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$

- δ is symmetry factor (two for identical particles) and $J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})$ is Jacobian of variable transformation (with $\boldsymbol{\eta}'$ held constant)

$$J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta}) = \left|\frac{\partial\tilde{\boldsymbol{\eta}}}{\partial\boldsymbol{\eta}}\right| = \begin{vmatrix} \frac{\partial\tilde{\eta}_1}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_1}{\partial\eta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial\tilde{\eta}_M}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_M}{\partial\eta_M} \end{vmatrix}$$

- Relationship between $\tilde{\boldsymbol{\eta}}$, $\boldsymbol{\eta}'$ and $\boldsymbol{\eta}$ is derived by continuity statements
- Negative source term is

$$h^-(t,\mathbf{x},\boldsymbol{\eta}) = \int \beta(\boldsymbol{\eta}',\boldsymbol{\eta})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$



Mesoscale Model for Second-Order Point Processes

- Positive source term is

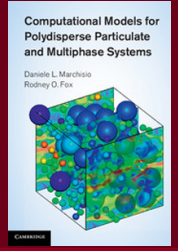
$$h^+(t,\mathbf{x},\boldsymbol{\eta}) = \frac{1}{\delta}\int \beta(\boldsymbol{\eta}',\tilde{\boldsymbol{\eta}})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\tilde{\boldsymbol{\eta}})J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$

- δ is symmetry factor (two for identical particles) and $J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta})$ is Jacobian of variable transformation (with $\boldsymbol{\eta}'$ held constant)

$$J(\tilde{\boldsymbol{\eta}},\boldsymbol{\eta}) = \left|\frac{\partial\tilde{\boldsymbol{\eta}}}{\partial\boldsymbol{\eta}}\right| = \begin{vmatrix} \frac{\partial\tilde{\eta}_1}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_1}{\partial\eta_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial\tilde{\eta}_M}{\partial\eta_1} & \cdots & \frac{\partial\tilde{\eta}_M}{\partial\eta_M} \end{vmatrix}$$

- Relationship between $\tilde{\boldsymbol{\eta}}$, $\boldsymbol{\eta}'$ and $\boldsymbol{\eta}$ is derived by continuity statements
- Negative source term is

$$h^-(t,\mathbf{x},\boldsymbol{\eta}) = \int \beta(\boldsymbol{\eta}',\boldsymbol{\eta})n(t,\mathbf{x},\boldsymbol{\eta}')n(t,\mathbf{x},\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}'$$



Mesoscale Model for Second-Order Point Processes

Aggregation of fine particles

- Internal coordinate particle mass ($\xi_p = \xi'_p + \tilde{\xi}_p$; Jacobian equal to unity)

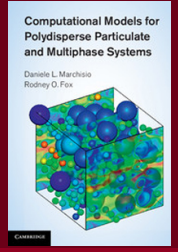
$$\mathcal{S}_1(\xi_p) = \frac{1}{2}\int_0^{\xi_p} \beta(\xi', \tilde{\xi}_p) n(\xi'_p) n(\tilde{\xi}_p)\, \mathrm{d}\xi'_p - \int_0^{\infty} \beta(\xi'_p, \xi_p) n(\xi'_p) n(\xi_p)\, \mathrm{d}\xi'_p$$

or in terms of final mass ξ_p and of field particle mass ξ'_p

$$\mathcal{S}_1(\xi_p) = \frac{1}{2}\int_0^{\xi_p} \beta(\xi'_p, \xi_p - \xi'_p) n(\xi'_p) n(\xi_p - \xi')\, \mathrm{d}\xi' - \int_0^{\infty} \beta(\xi'_p, \xi_p) n(\xi'_p) n(\xi_p)\, \mathrm{d}\xi'_p$$

- Particle size ($\tilde{\xi}_p = [\xi_p^3 - (\xi'_p)^3]^{1/3}$; Jacobian equal to $\xi_p^2[\xi_p^3 - (\xi'_p)^3]^{-2/3}$)

$$\mathcal{S}_1(\xi_p) = \frac{\xi_p^2}{2}\int_0^{\xi_p} \left[\xi_p^3 - (\xi'_p)^3\right]^{-2/3} \beta(\xi'_p, [\xi_p^3 - (\xi'_p)^3]^{1/3}) n(\xi'_p) \times n([\xi_p^3 - (\xi'_p)^3]^{1/3})\, \mathrm{d}\xi'_p - \int_0^{\infty} \beta(\xi'_p, \xi_p) n(\xi'_p) n(\xi_p)\, \mathrm{d}\xi'_p$$



Mesoscale Model for Second-Order Point Processes

Coalescence of droplets

- Coalescing droplets characterized by particle mass ξ_p and velocity $\mathbf{v}_p$; continuity statements

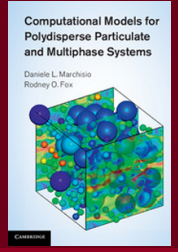
$$\xi_p = \xi'_p + \tilde{\xi}_p, \quad \xi_p \mathbf{v}_p = \xi'_p \mathbf{v}'_p + \tilde{\xi}_p \tilde{\mathbf{v}}_p$$

or, equivalently, as

$$\tilde{\xi}_p = \xi_p - \xi'_p, \quad \tilde{\mathbf{v}}_p = (\xi_p \mathbf{v}_p - \xi'_p \mathbf{v}'_p)/(\xi_p - \xi'_p)$$

- Jacobian for this nonlinear transformation: $J = \frac{\partial(\tilde{\xi}_p, \tilde{\mathbf{v}}_p)}{\partial(\xi_p, \mathbf{v}_p)} = \left(\frac{\xi_p}{\xi_p - \xi'_p}\right)^3$ and resulting source term is

$$\mathcal{S}_1(\xi_p, \mathbf{v}_p) = \frac{1}{2} \int \left(\int_0^{\xi_p} \beta(\xi'_p, \mathbf{v}'_p, \tilde{\xi}_p, \tilde{\mathbf{v}}_p) n(\xi'_p, \mathbf{v}'_p) n(\tilde{\xi}_p, \tilde{\mathbf{v}}_p) \frac{\xi_p^3}{(\xi_p - \xi'_p)^3} \, \mathrm{d}\xi'_p \right) \times \mathrm{d}\mathbf{v}'_p - \int \beta(\xi'_p, \mathbf{v}'_p, \xi_p, \mathbf{v}_p) n(\xi'_p, \mathbf{v}'_p) n(\xi_p, \mathbf{v}_p) \, \mathrm{d}\xi'_p \, \mathrm{d}\mathbf{v}'_p$$



Mesoscale Model for Second-Order Point Processes

Aggregation kernels due to Brownian motions for fine particles (i.e., $\mathrm{St}_\mathrm{p} = 0$)

- Continuum or Stokes-Einstein regime ($\mathrm{Kn}^* = 2\lambda_\mathrm{f}/d_\mathrm{p} < 1$)

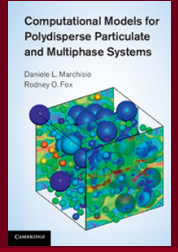
$$\beta(d_\mathrm{p}, d'_\mathrm{p}) = \frac{2k_\mathrm{B}T_\mathrm{f}}{3\mu_\mathrm{f}}\left(\frac{1}{d_\mathrm{p}} + \frac{1}{d'_\mathrm{p}}\right)\left(d_\mathrm{p} + d'_\mathrm{p}\right) = \frac{2k_\mathrm{B}T_\mathrm{f}}{3\mu_\mathrm{f}}\frac{\left(d_\mathrm{p} + d'_\mathrm{p}\right)^2}{d_\mathrm{p}d'_\mathrm{p}}$$

- Free-molecular or Epstein regime ($\mathrm{Kn}^* = 2\lambda_\mathrm{f}/d_\mathrm{p} > 1$)

$$\beta(d_\mathrm{p}, d'_\mathrm{p}) = \left(\frac{3}{4\pi}\right)^{1/6}\left(\frac{6k_\mathrm{B}T_\mathrm{f}}{\rho_\mathrm{p}}\right)^{1/2}\left(d_\mathrm{p} + d'_\mathrm{p}\right)^2\left(\frac{1}{{d_\mathrm{p}}^3} + \frac{1}{{d'_\mathrm{p}}^3}\right)^{1/2}$$

- Fuchs interpolating kernel

$$\beta = 4\pi\left(\Gamma + \Gamma'\right)\left(d_\mathrm{p} + d'_\mathrm{p}\right)\left[\frac{d_\mathrm{p} + d'_\mathrm{p}}{d_\mathrm{p} + d'_\mathrm{p} + \sqrt{g^2 + g'^2}} + \frac{4\left(\Gamma + \Gamma'\right)}{\left(d_\mathrm{p} + d'_\mathrm{p}\right)\sqrt{c^2 + c'^2}}\right]^{-1}$$



Mesoscale Model for Second-Order Point Processes

Aggregation kernels due to shear / velocity gradients

- Presence of spatial gradients in fluid velocity can induce aggregation

$$\beta(d_\mathrm{p}, d'_\mathrm{p}) = \frac{4}{3} G_\mathrm{f} \left(d_\mathrm{p} + d'_\mathrm{p}\right)^3$$

  laminar flow: $G_\mathrm{f} = \frac{1}{2}\left(s_{ij}s_{ij} - s_{kk}^2\right)$, 2nd inv. of rate-of-strain tensor

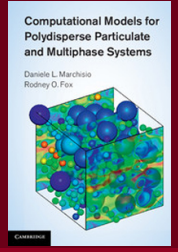
- In case of turbulent flows (particles smaller than Kolmogorov length scale)

$$\beta(d_\mathrm{p}, d'_\mathrm{p}) = \left(\frac{8\pi}{15}\right)^{1/2} \left(\frac{\nu_\mathrm{f}}{\varepsilon_\mathrm{f}}\right)^{1/2} \left(d_\mathrm{p} + d'_\mathrm{p}\right)^3 \approx 1.29 \left(\frac{\nu_\mathrm{f}}{\varepsilon_\mathrm{f}}\right)^{1/2} \left(d_\mathrm{p} + d'_\mathrm{p}\right)^3$$

- Particles larger than Kolmogorov length scale

$$\beta(d_\mathrm{p}, d'_\mathrm{p}) = \left(\frac{8\pi}{3}\right)^{1/2} \varepsilon_\mathrm{f}^{1/3} \left(d_\mathrm{p} + d'_\mathrm{p}\right)^{7/3}$$

- Corrections available for $\mathrm{St}_\mathrm{p} \neq 0$ (Ammar & Reeks, 2009)



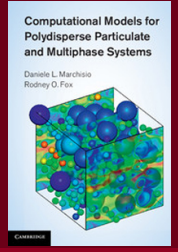
Mesoscale Model for Second-Order Point Processes

Aggregation efficiency for particles

- Short range forces accounted for in terms of aggregation efficiency: balance between attractive (van der Waals) and repulsive forces
- Repulsive forces: due to presence of fluid (double-layer repulsion or drainage of fluid)
- Aggregation efficiency as inverse of Fuch's stability ratio

$$W = \left(\frac{d_\mathrm{p} + d'_\mathrm{p}}{2}\right) \int_0^\infty \exp\left(\frac{V(h)}{k_\mathrm{B} T_\mathrm{f}}\right) \left[\left(\frac{d_\mathrm{p} + d'_\mathrm{p}}{2}\right) + h\right]^{-2} \mathrm{d}h$$

  h inter-particle distance, $V(h) = V_\mathrm{a}(h) + V_\mathrm{r}(h)$ inter-particle potential
- Simplified expressions (van der Waals + double-layer): $\alpha = \frac{d_\mathrm{p}+d'_\mathrm{p}}{2\kappa} \exp\left(-\frac{\phi_\mathrm{max}}{k_\mathrm{B} T_\mathrm{f}}\right)$, where κ is Debye-Huckel parameter and ϕ_max is energy barrier for particle aggregation
- Simplified expressions (van der Waals + viscous): $\alpha = k\mathrm{Fl}^{-0.18}$, where k is pre-factor of order unity and $\mathrm{Fl} = 6\pi\mu_\mathrm{f}(d_\mathrm{p} + d'_\mathrm{p})G_\mathrm{f}/A_\mathrm{H}$



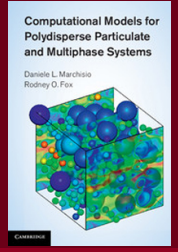
Mesoscale Model for Second-Order Point Processes

Coalescence kernels for droplets (continuous phase gas)

- Coalescence kernel calculated with concept of collisional cylinder

$$\beta(d_\mathrm{p}, d'_\mathrm{p}, \mathbf{v}_\mathrm{r}) = \frac{\pi}{4}\left(d_\mathrm{p} + d'_\mathrm{p}\right)^2 |\mathbf{v}_\mathrm{r}| \eta(|\mathbf{v}_\mathrm{r}|)$$

- Function $\eta \leq 1$ is coalescence efficiency that depends on Weber number: $\mathrm{We} = \frac{\rho_\mathrm{p} |\mathbf{v}_\mathrm{r}| d_\mathrm{p}}{\sigma}$ (relative importance of fluid inertia compared to interfacial tension)
- If We is smaller than 10 coalescence is very likely and efficiency η is very close to one
- As We increases coalescence is possible only for impact factor close to 0.2 (impact angles of about 10–15°; under these conditions coalescence is unlikely to happen (i.e., $\eta \approx 0$) and droplets after colliding are said to undergo lateral or head-on bouncing and reflective or stretching separation



Mesoscale Model for Second-Order Point Processes

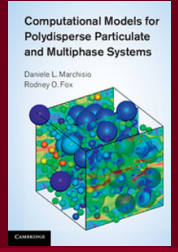
Coalescence kernels for bubbles (continuous phase liquid)

- Small bubbles coalescing due to turbulent velocity fluctuations in liquid

$$\beta(d_\mathrm{p}, d'_\mathrm{p}) = 0.88\epsilon_\mathrm{f}^{1/3}(d_\mathrm{p} + d'_\mathrm{p})^2(d_\mathrm{p}^{2/3} + {d'_\mathrm{p}}^{2/3})^{1/2}\eta(d_\mathrm{p}, d'_\mathrm{p})$$

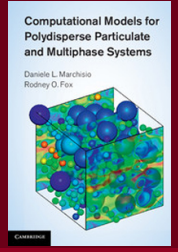
- Coalescence efficiency is

$$\eta(d_\mathrm{p}, d'_\mathrm{p}) = \exp\left(-6\times 10^9\ \frac{\mu_\mathrm{f}\rho_\mathrm{f}\epsilon_\mathrm{f}}{\sigma^2}\left(\frac{d_\mathrm{p}d'_\mathrm{p}}{d_\mathrm{p}+d'_\mathrm{p}}\right)^4\right)$$



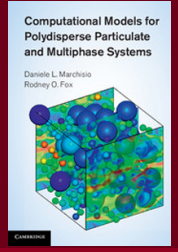
Summary of Mesoscale Models

- Mesoscale models are very problem dependent!
- Models can be classified as continuous and discontinuous processes
- Discontinuous processses further divided into zero-, first- and second-order depending on number of particles involved
- Real physical models tend to be highly nonlinear, so no chance of finding analytical solutions for PBE/GPBE!



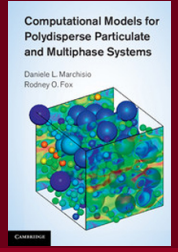
Summary of Mesoscale Models

- Mesoscale models are very problem dependent!
- Models can be classified as continuous and discontinuous processes
- Discontinuous processses further divided into zero-, first- and second-order depending on number of particles involved
- Real physical models tend to be highly nonlinear, so no chance of finding analytical solutions for PBE/GPBE!



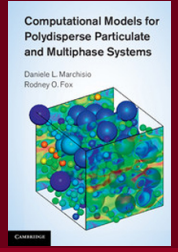
Summary of Mesoscale Models

- Mesoscale models are very problem dependent!
- Models can be classified as continuous and discontinuous processes
- Discontinuous processses further divided into zero-, first- and second-order depending on number of particles involved
- Real physical models tend to be highly nonlinear, so no chance of finding analytical solutions for PBE/GPBE!

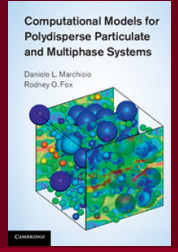


Summary of Mesoscale Models

- Mesoscale models are very problem dependent!
- Models can be classified as continuous and discontinuous processes
- Discontinuous processses further divided into zero-, first- and second-order depending on number of particles involved
- Real physical models tend to be highly nonlinear, so no chance of finding analytical solutions for PBE/GPBE!



SOLUTION METHODS FOR HOMOGENEOUS SYSTEMS



Class Method

- Our objective is to solve GPBE for $n(t, \mathbf{x}, \mathbf{v}, \xi)$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v} n + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}_{\mathrm{p}} n + \frac{\partial}{\partial \xi} \cdot \dot{\xi} n = \mathcal{S}$$

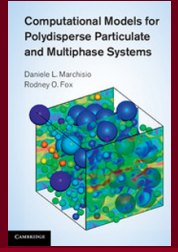
- After integrating out particle velocity, find PBE for $n(t, \mathbf{x}, \xi)$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_{\mathrm{p}} | \xi \rangle n + \frac{\partial}{\partial \xi} \cdot \dot{\xi} n = \mathcal{S}$$

- Since transport in space is handled by CFD code, let us focus on other terms
- PBE for **spatially homogeneous system** with phase-space diffusion

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \xi} \cdot \dot{\xi} n = \frac{\partial^2}{\partial \xi \partial \xi} : \mathbf{D} n + \mathcal{S}$$

where $n(t, \xi)$ and $\xi = (\xi_1, \ldots, \xi_M)$



Class Method

- Our objective is to solve GPBE for $n(t, \mathbf{x}, \mathbf{v}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v} n + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}_{\mathrm{p}} n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

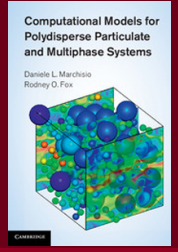
- After integrating out particle velocity, find PBE for $n(t, \mathbf{x}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_{\mathrm{p}} | \boldsymbol{\xi} \rangle n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

- Since transport in space is handled by CFD code, let us focus on other terms
- PBE for **spatially homogeneous system** with phase-space diffusion

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S}$$

where $n(t, \boldsymbol{\xi})$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)$



Class Method

- Our objective is to solve GPBE for $n(t, \mathbf{x}, \mathbf{v}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v} n + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}_{\mathrm{p}} n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

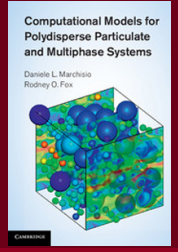
- After integrating out particle velocity, find PBE for $n(t, \mathbf{x}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_{\mathrm{p}} | \boldsymbol{\xi} \rangle n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

- Since transport in space is handled by CFD code, let us focus on other terms
- PBE for **spatially homogeneous system** with phase-space diffusion

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S}$$

where $n(t, \boldsymbol{\xi})$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)$



Class Method

- Our objective is to solve GPBE for $n(t, \mathbf{x}, \mathbf{v}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v} n + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}_{\mathrm{p}} n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

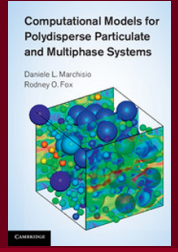
- After integrating out particle velocity, find PBE for $n(t, \mathbf{x}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_{\mathrm{p}} | \boldsymbol{\xi} \rangle n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

- Since transport in space is handled by CFD code, let us focus on other terms
- PBE for **spatially homogeneous system** with phase-space diffusion

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S}$$

where $n(t, \boldsymbol{\xi})$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)$



Class Method

- Our objective is to solve GPBE for $n(t, \mathbf{x}, \mathbf{v}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{v} n + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}_{\mathrm{p}} n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

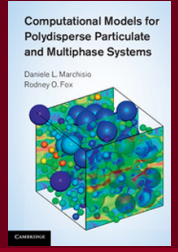
- After integrating out particle velocity, find PBE for $n(t, \mathbf{x}, \boldsymbol{\xi})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_{\mathrm{p}} | \boldsymbol{\xi} \rangle n + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \mathcal{S}$$

- Since transport in space is handled by CFD code, let us focus on other terms
- PBE for **spatially homogeneous system** with phase-space diffusion

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n = \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S}$$

where $n(t, \boldsymbol{\xi})$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)$



Class Method (Univariate)

- Let ξ denote unique internal coordinate and $n(t,\xi)$ NDF
- Univariate PBE is

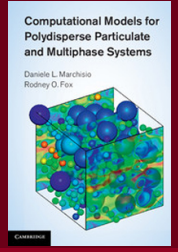
$$\frac{\partial n}{\partial t} + \frac{\partial \dot{\xi} n}{\partial \xi} = \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- Expressing drift, diffusion and point processes as a unique source term

$$\frac{\partial n}{\partial t} = S \equiv -\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- $S(t,\xi,n)$ is functional that maps NDF into function of time and internal coordinate
- Define finite interval I_i for internal coordinate bounded by ξ_i and ξ_{i+1}

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\xi_i}^{\xi_{i+1}} n \,\mathrm{d}\xi = \int_{\xi_i}^{\xi_{i+1}} S \,\mathrm{d}\xi$$



Class Method (Univariate)

- Let ξ denote unique internal coordinate and $n(t,\xi)$ NDF
- Univariate PBE is

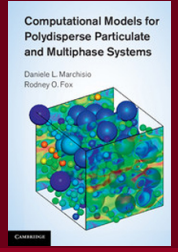
$$\frac{\partial n}{\partial t} + \frac{\partial \dot{\xi} n}{\partial \xi} = \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- Expressing drift, diffusion and point processes as a unique source term

$$\frac{\partial n}{\partial t} = S \equiv -\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- $S(t,\xi,n)$ is functional that maps NDF into function of time and internal coordinate
- Define finite interval I_i for internal coordinate bounded by ξ_i and ξ_{i+1}

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\xi_i}^{\xi_{i+1}} n \,\mathrm{d}\xi = \int_{\xi_i}^{\xi_{i+1}} S \,\mathrm{d}\xi$$



Class Method (Univariate)

- Let ξ denote unique internal coordinate and $n(t,\xi)$ NDF
- Univariate PBE is

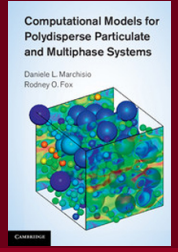
$$\frac{\partial n}{\partial t} + \frac{\partial \dot{\xi} n}{\partial \xi} = \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- Expressing drift, diffusion and point processes as a unique source term

$$\frac{\partial n}{\partial t} = S \equiv -\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- $S(t,\xi,n)$ is functional that maps NDF into function of time and internal coordinate
- Define finite interval I_i for internal coordinate bounded by ξ_i and ξ_{i+1}

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\xi_i}^{\xi_{i+1}} n \,\mathrm{d}\xi = \int_{\xi_i}^{\xi_{i+1}} S \,\mathrm{d}\xi$$



Class Method (Univariate)

- Let ξ denote unique internal coordinate and $n(t,\xi)$ NDF
- Univariate PBE is

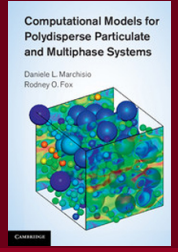
$$\frac{\partial n}{\partial t} + \frac{\partial \dot{\xi} n}{\partial \xi} = \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- Expressing drift, diffusion and point processes as a unique source term

$$\frac{\partial n}{\partial t} = S \equiv -\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S}$$

- $S(t,\xi,n)$ is functional that maps NDF into function of time and internal coordinate
- Define finite interval I_i for internal coordinate bounded by ξ_i and ξ_{i+1}

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\xi_i}^{\xi_{i+1}} n \,\mathrm{d}\xi = \int_{\xi_i}^{\xi_{i+1}} S \,\mathrm{d}\xi$$



Class Method (Univariate)

- If we denote number of particles in I_i interval as

$$N_i := \int_{\xi_i}^{\xi_{i+1}} n \, \mathrm{d}\xi$$

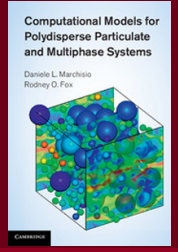
we recognize that

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \int_{\xi_i}^{\xi_{i+1}} S \, \mathrm{d}\xi$$

where $i = 1, \ldots, M$, corresponding to following partition P_M

$$\xi_1 < \xi_2 < \xi_3 < \cdots < \xi_{M-1} < \xi_M < \xi_{M+1}$$

- Values of ξ_1 and ξ_M are chosen so that excluded dynamics is negligible
- Number of intervals is fixed in order to achieve desired accuracy
- Resulting set of M ordinary differential equations (ODE) is unclosed!
- Use of discrete methods depends on **restoration of autonomy**, which lies in our ability to express right-hand side entirely in terms of dependent variables N_i



Class Method (Univariate)

- If we denote number of particles in I_i interval as

$$N_i := \int_{\xi_i}^{\xi_{i+1}} n \, \mathrm{d}\xi$$

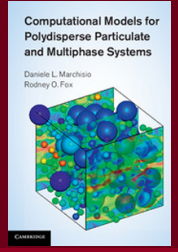
we recognize that

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \int_{\xi_i}^{\xi_{i+1}} S \, \mathrm{d}\xi$$

where $i = 1, \ldots, M$, corresponding to following partition P_M

$$\xi_1 < \xi_2 < \xi_3 < \cdots < \xi_{M-1} < \xi_M < \xi_{M+1}$$

- Values of ξ_1 and ξ_M are chosen so that excluded dynamics is negligible
- Number of intervals is fixed in order to achieve desired accuracy
- Resulting set of M ordinary differential equations (ODE) is unclosed!
- Use of discrete methods depends on **restoration of autonomy**, which lies in our ability to express right-hand side entirely in terms of dependent variables N_i



Class Method (Univariate)

- If we denote number of particles in I_i interval as

$$N_i := \int_{\xi_i}^{\xi_{i+1}} n \, \mathrm{d}\xi$$

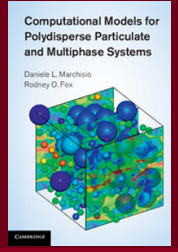
we recognize that

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \int_{\xi_i}^{\xi_{i+1}} S \, \mathrm{d}\xi$$

where $i = 1, \ldots, M$, corresponding to following partition P_M

$$\xi_1 < \xi_2 < \xi_3 < \cdots < \xi_{M-1} < \xi_M < \xi_{M+1}$$

- Values of ξ_1 and ξ_M are chosen so that excluded dynamics is negligible
- Number of intervals is fixed in order to achieve desired accuracy
- Resulting set of M ordinary differential equations (ODE) is unclosed!
- Use of discrete methods depends on **restoration of autonomy**, which lies in our ability to express right-hand side entirely in terms of dependent variables N_i



Class Method (Univariate)

- If we denote number of particles in I_i interval as

$$N_i := \int_{\xi_i}^{\xi_{i+1}} n \, \mathrm{d}\xi$$

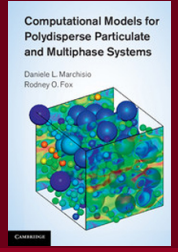
we recognize that

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \int_{\xi_i}^{\xi_{i+1}} S \, \mathrm{d}\xi$$

where $i = 1, \ldots, M$, corresponding to following partition P_M

$$\xi_1 < \xi_2 < \xi_3 < \cdots < \xi_{M-1} < \xi_M < \xi_{M+1}$$

- Values of ξ_1 and ξ_M are chosen so that excluded dynamics is negligible
- Number of intervals is fixed in order to achieve desired accuracy
- Resulting set of M ordinary differential equations (ODE) is unclosed!
- Use of discrete methods depends on **restoration of autonomy**, which lies in our ability to express right-hand side entirely in terms of dependent variables N_i



Class Method (Univariate)

- If we denote number of particles in I_i interval as

$$N_i := \int_{\xi_i}^{\xi_{i+1}} n \, \mathrm{d}\xi$$

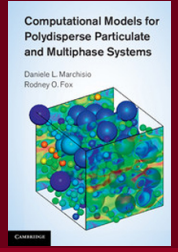
we recognize that

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \int_{\xi_i}^{\xi_{i+1}} S \, \mathrm{d}\xi$$

where $i = 1, \ldots, M$, corresponding to following partition P_M

$$\xi_1 < \xi_2 < \xi_3 < \cdots < \xi_{M-1} < \xi_M < \xi_{M+1}$$

- Values of ξ_1 and ξ_M are chosen so that excluded dynamics is negligible
- Number of intervals is fixed in order to achieve desired accuracy
- Resulting set of M ordinary differential equations (ODE) is unclosed!
- Use of discrete methods depends on **restoration of autonomy**, which lies in our ability to express right-hand side entirely in terms of dependent variables N_i



Class Method (Univariate)

- If we denote number of particles in I_i interval as

$$N_i := \int_{\xi_i}^{\xi_{i+1}} n \, \mathrm{d}\xi$$

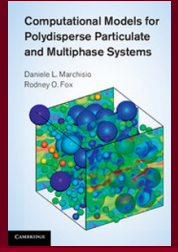
we recognize that

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \int_{\xi_i}^{\xi_{i+1}} S \, \mathrm{d}\xi$$

where $i = 1, \ldots, M$, corresponding to following partition P_M

$$\xi_1 < \xi_2 < \xi_3 < \cdots < \xi_{M-1} < \xi_M < \xi_{M+1}$$

- Values of ξ_1 and ξ_M are chosen so that excluded dynamics is negligible
- Number of intervals is fixed in order to achieve desired accuracy
- Resulting set of M ordinary differential equations (ODE) is unclosed!
- Use of discrete methods depends on **restoration of autonomy**, which lies in our ability to express right-hand side entirely in terms of dependent variables N_i



Class Method (Univariate)

- This is done by making assumptions on functional form for NDF, $n(t,\xi) = N_i(t)k_i(\xi)$ for $\xi_i < \xi < \xi_{i+1}$, where $i = 1, \ldots, M$, and $k_i(\xi)$ is a polynomial that has property

$$\int_{\xi_i}^{\xi_{i+1}} k_i(\xi)\,\mathrm{d}\xi = 1$$

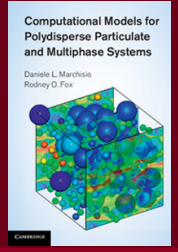
- In case of **zero-order polynomial**, (constant) approximation is employed within each class

$$n(t,\xi) = \frac{N_i(t)}{\xi_{i+1} - \xi_i} \quad \text{for } \xi_i < \xi < \xi_{i+1}$$

- Alternative assumption is that of **pivotal points** ζ_i where all particles in interval are concentrated

$$n(t,\xi) = \sum_{i=1}^{M} N_i(t)\delta(\xi - \zeta_i)$$

- Very often discretization is not written in terms of number of particles, rather in terms of volume (or mass) of particles belonging to class



Class Method (Univariate)

- This is done by making assumptions on functional form for NDF, $n(t,\xi) = N_i(t)k_i(\xi)$ for $\xi_i < \xi < \xi_{i+1}$, where $i = 1,\ldots,M$, and $k_i(\xi)$ is a polynomial that has property

$$\int_{\xi_i}^{\xi_{i+1}} k_i(\xi)\,\mathrm{d}\xi = 1$$

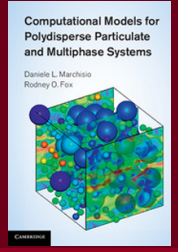
- In case of **zero-order polynomial**, (constant) approximation is employed within each class

$$n(t,\xi) = \frac{N_i(t)}{\xi_{i+1} - \xi_i} \quad \text{for } \xi_i < \xi < \xi_{i+1}$$

- Alternative assumption is that of **pivotal points** ζ_i where all particles in interval are concentrated

$$n(t,\xi) = \sum_{i=1}^{M} N_i(t)\delta(\xi - \zeta_i)$$

- Very often discretization is not written in terms of number of particles, rather in terms of volume (or mass) of particles belonging to class



Class Method (Univariate)

- This is done by making assumptions on functional form for NDF, $n(t,\xi) = N_i(t)k_i(\xi)$ for $\xi_i < \xi < \xi_{i+1}$, where $i = 1,\ldots,M$, and $k_i(\xi)$ is a polynomial that has property

$$\int_{\xi_i}^{\xi_{i+1}} k_i(\xi)\,\mathrm{d}\xi = 1$$

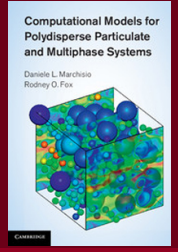
- In case of **zero-order polynomial**, (constant) approximation is employed within each class

$$n(t,\xi) = \frac{N_i(t)}{\xi_{i+1} - \xi_i} \quad \text{for } \xi_i < \xi < \xi_{i+1}$$

- Alternative assumption is that of **pivotal points** ζ_i where all particles in interval are concentrated

$$n(t,\xi) = \sum_{i=1}^{M} N_i(t)\delta(\xi - \zeta_i)$$

- Very often discretization is not written in terms of number of particles, rather in terms of volume (or mass) of particles belonging to class



Class Method (Univariate)

- This is done by making assumptions on functional form for NDF, $n(t,\xi) = N_i(t)k_i(\xi)$ for $\xi_i < \xi < \xi_{i+1}$, where $i = 1,\ldots,M$, and $k_i(\xi)$ is a polynomial that has property

$$\int_{\xi_i}^{\xi_{i+1}} k_i(\xi)\,\mathrm{d}\xi = 1$$

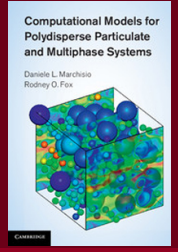
- In case of **zero-order polynomial**, (constant) approximation is employed within each class

$$n(t,\xi) = \frac{N_i(t)}{\xi_{i+1} - \xi_i} \quad \text{for } \xi_i < \xi < \xi_{i+1}$$

- Alternative assumption is that of **pivotal points** ζ_i where all particles in interval are concentrated

$$n(t,\xi) = \sum_{i=1}^{M} N_i(t)\delta(\xi - \zeta_i)$$

- Very often discretization is not written in terms of number of particles, rather in terms of volume (or mass) of particles belonging to class



Class Method (Univariate)

- This is done by making assumptions on functional form for NDF, $n(t,\xi) = N_i(t)k_i(\xi)$ for $\xi_i < \xi < \xi_{i+1}$, where $i = 1, \ldots, M$, and $k_i(\xi)$ is a polynomial that has property

$$\int_{\xi_i}^{\xi_{i+1}} k_i(\xi)\,\mathrm{d}\xi = 1$$

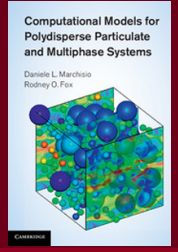
- In case of **zero-order polynomial**, (constant) approximation is employed within each class

$$n(t,\xi) = \frac{N_i(t)}{\xi_{i+1} - \xi_i} \quad \text{for } \xi_i < \xi < \xi_{i+1}$$

- Alternative assumption is that of **pivotal points** ζ_i where all particles in interval are concentrated

$$n(t,\xi) = \sum_{i=1}^{M} N_i(t)\delta(\xi - \zeta_i)$$

- Very often discretization is not written in terms of number of particles, rather in terms of volume (or mass) of particles belonging to class



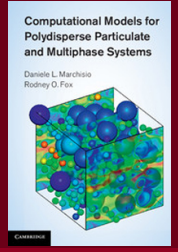
Class Method (Univariate)

- Let us start by discussing implication related to choice of a particular discretization and restoration of autonomy for second-order point processes
- Two different approaches: **fixed-pivot technique** and **cell-average technique**
- If ξ represents an additive property then

$$S = \frac{1}{2}\int_0^{\xi} \beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi' - n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'$$

resulting in

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\int_{\xi_i}^{\xi_{i+1}}\int_0^{\xi}\beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi - \int_{\xi_i}^{\xi_{i+1}} n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$



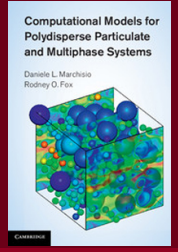
Class Method (Univariate)

- Let us start by discussing implication related to choice of a particular discretization and restoration of autonomy for second-order point processes
- Two different approaches: **fixed-pivot technique** and **cell-average technique**
- If ξ represents an additive property then

$$S = \frac{1}{2}\int_0^{\xi} \beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi' - n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'$$

resulting in

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\int_{\xi_i}^{\xi_{i+1}}\int_0^{\xi}\beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi - \int_{\xi_i}^{\xi_{i+1}} n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$



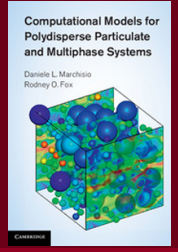
Class Method (Univariate)

- Let us start by discussing implication related to choice of a particular discretization and restoration of autonomy for second-order point processes
- Two different approaches: **fixed-pivot technique** and **cell-average technique**
- If ξ represents an additive property then

$$S = \frac{1}{2}\int_0^{\xi} \beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi' - n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'$$

resulting in

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\int_{\xi_i}^{\xi_{i+1}}\int_0^{\xi}\beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi - \int_{\xi_i}^{\xi_{i+1}} n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$



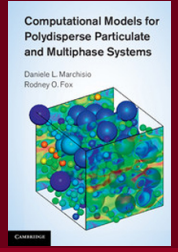
Class Method (Univariate)

- Let us start by discussing implication related to choice of a particular discretization and restoration of autonomy for second-order point processes
- Two different approaches: **fixed-pivot technique** and **cell-average technique**
- If ξ represents an additive property then

$$S = \frac{1}{2}\int_0^{\xi} \beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi' - n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'$$

resulting in

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\int_{\xi_i}^{\xi_{i+1}}\int_0^{\xi}\beta(\xi-\xi',\xi')n(\xi-\xi')n(\xi')\,\mathrm{d}\xi'\mathrm{d}\xi - \int_{\xi_i}^{\xi_{i+1}} n(\xi)\int_0^{\infty}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\mathrm{d}\xi$$



Class Method (Univariate)

- Expressing second integral as summation of integrals

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\int_{\xi_i}^{\xi_{i+1}}\int_{\xi_j}^{\xi_{j+1}}\beta(\xi-\xi',\xi')n_\xi(\xi-\xi')n_\xi(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$

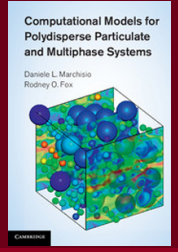
$$-\sum_{j=1}^{M}\int_{\xi_i}^{\xi_{i+1}}n_\xi(\xi)\int_{\xi_j}^{\xi_{j+1}}\beta(\xi,\xi')n_\xi(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$

- Restoration of autonomy by using concept of pivotal points or pivots

$$\int_{\xi_i}^{\xi_{i+1}}n(\xi)\int_{\xi_j}^{\xi_{j+1}}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi = N_iN_j\beta(\zeta_i,\zeta_j)$$

where ζ_i and ζ_j are pivotal points resulting in

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}N_j\sum_{(\zeta_j+\zeta_k)\in I_i}\beta(\zeta_k,\zeta_j)N_k - N_i\sum_{j=1}^{M}\beta(\zeta_i,\zeta_j)N_j$$



Class Method (Univariate)

- Expressing second integral as summation of integrals

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\int_{\xi_i}^{\xi_{i+1}}\int_{\xi_j}^{\xi_{j+1}}\beta(\xi-\xi',\xi')n_\xi(\xi-\xi')n_\xi(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$

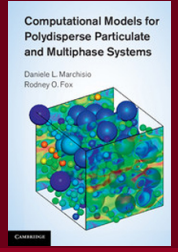
$$-\sum_{j=1}^{M}\int_{\xi_i}^{\xi_{i+1}}n_\xi(\xi)\int_{\xi_j}^{\xi_{j+1}}\beta(\xi,\xi')n_\xi(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$

- Restoration of autonomy by using concept of pivotal points or pivots

$$\int_{\xi_i}^{\xi_{i+1}}n(\xi)\int_{\xi_j}^{\xi_{j+1}}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi = N_iN_j\beta(\zeta_i,\zeta_j)$$

where ζ_i and ζ_j are pivotal points resulting in

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}N_j\sum_{(\zeta_j+\zeta_k)\in I_i}\beta(\zeta_k,\zeta_j)N_k - N_i\sum_{j=1}^{M}\beta(\zeta_i,\zeta_j)N_j$$



Class Method (Univariate)

- Expressing second integral as summation of integrals

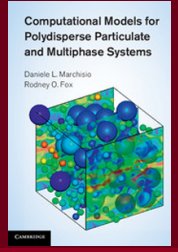
$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\int_{\xi_i}^{\xi_{i+1}}\int_{\xi_j}^{\xi_{j+1}}\beta(\xi-\xi',\xi')n_\xi(\xi-\xi')n_\xi(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi - \sum_{j=1}^{M}\int_{\xi_i}^{\xi_{i+1}}n_\xi(\xi)\int_{\xi_j}^{\xi_{j+1}}\beta(\xi,\xi')n_\xi(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi$$

- Restoration of autonomy by using concept of pivotal points or pivots

$$\int_{\xi_i}^{\xi_{i+1}}n(\xi)\int_{\xi_j}^{\xi_{j+1}}\beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi = N_iN_j\beta(\zeta_i,\zeta_j)$$

where ζ_i and ζ_j are pivotal points resulting in

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}N_j\sum_{(\zeta_j+\zeta_k)\in I_i}\beta(\zeta_k,\zeta_j)N_k - N_i\sum_{j=1}^{M}\beta(\zeta_i,\zeta_j)N_j$$



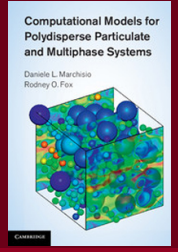
Class Method (Univariate)

- Generation term causes most problems!
- When particle in I_j aggregates with particle in I_k, particle with mass equal to $\zeta_j + \zeta_k$ is formed
- Unless a linear partition is used, it is very unlikely to have exactly a pivot at $\zeta_i = \zeta_j + \zeta_k$
- Newly formed particle has to be shared between neighboring pivotal points
- Equation for moment of order r

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k (\zeta_j+\zeta_k)^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$

- we multiply by ${\zeta_i}^r$ and sum over all possible values of index i:

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k {\zeta_i}^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$



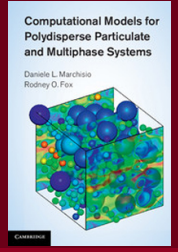
Class Method (Univariate)

- Generation term causes most problems!
- When particle in I_j aggregates with particle in I_k, particle with mass equal to $\zeta_j + \zeta_k$ is formed
- Unless a linear partition is used, it is very unlikely to have exactly a pivot at $\zeta_i = \zeta_j + \zeta_k$
- Newly formed particle has to be shared between neighboring pivotal points
- Equation for moment of order r

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k (\zeta_j+\zeta_k)^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$

- we multiply by ${\zeta_i}^r$ and sum over all possible values of index i:

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k \zeta_i^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$



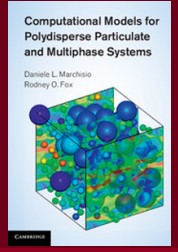
Class Method (Univariate)

- Generation term causes most problems!
- When particle in I_j aggregates with particle in I_k, particle with mass equal to $\zeta_j + \zeta_k$ is formed
- Unless a linear partition is used, it is very unlikely to have exactly a pivot at $\zeta_i = \zeta_j + \zeta_k$
- Newly formed particle has to be shared between neighboring pivotal points
- Equation for moment of order r

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k (\zeta_j+\zeta_k)^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$

- we multiply by ${\zeta_i}^r$ and sum over all possible values of index i:

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k \zeta_i^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$



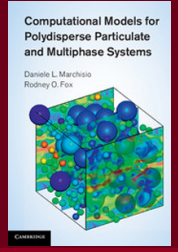
Class Method (Univariate)

- Generation term causes most problems!
- When particle in I_j aggregates with particle in I_k, particle with mass equal to $\zeta_j + \zeta_k$ is formed
- Unless a linear partition is used, it is very unlikely to have exactly a pivot at $\zeta_i = \zeta_j + \zeta_k$
- Newly formed particle has to be shared between neighboring pivotal points
- Equation for moment of order r

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k (\zeta_j+\zeta_k)^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$

- we multiply by ${\zeta_i}^r$ and sum over all possible values of index i:

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k {\zeta_i}^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$



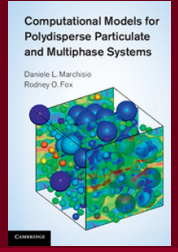
Class Method (Univariate)

- Generation term causes most problems!
- When particle in I_j aggregates with particle in I_k, particle with mass equal to $\zeta_j + \zeta_k$ is formed
- Unless a linear partition is used, it is very unlikely to have exactly a pivot at $\zeta_i = \zeta_j + \zeta_k$
- Newly formed particle has to be shared between neighboring pivotal points
- Equation for moment of order r

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k (\zeta_j+\zeta_k)^r - \sum_{i=1}^{M} N_i \zeta_i{}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$

- we multiply by $\zeta_i{}^r$ and sum over all possible values of index i:

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k \zeta_i^r - \sum_{i=1}^{M} N_i \zeta_i{}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$



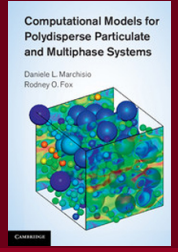
Class Method (Univariate)

- Generation term causes most problems!
- When particle in I_j aggregates with particle in I_k, particle with mass equal to $\zeta_j + \zeta_k$ is formed
- Unless a linear partition is used, it is very unlikely to have exactly a pivot at $\zeta_i = \zeta_j + \zeta_k$
- Newly formed particle has to be shared between neighboring pivotal points
- Equation for moment of order r

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k (\zeta_j+\zeta_k)^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$

- we multiply by ${\zeta_i}^r$ and sum over all possible values of index i:

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k \zeta_i^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$



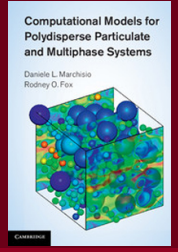
Class Method (Univariate)

- Generation term causes most problems!
- When particle in I_j aggregates with particle in I_k, particle with mass equal to $\zeta_j + \zeta_k$ is formed
- Unless a linear partition is used, it is very unlikely to have exactly a pivot at $\zeta_i = \zeta_j + \zeta_k$
- Newly formed particle has to be shared between neighboring pivotal points
- Equation for moment of order r

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k (\zeta_j+\zeta_k)^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$

- we multiply by ${\zeta_i}^r$ and sum over all possible values of index i:

$$\frac{\mathrm{d}m_r}{\mathrm{d}t} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} N_j \sum_{(\zeta_j+\zeta_k)\in I_i} \beta(\zeta_k,\zeta_j) N_k \zeta_i^r - \sum_{i=1}^{M} N_i {\zeta_i}^r \sum_{j=1}^{M} \beta(\zeta_i,\zeta_j) N_j$$



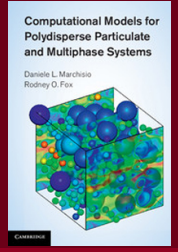
Class Method (Univariate)

- Equations are identical, and therefore method is **internally consistent** if and only if chosen grid is such that a particle belonging to the I_j interval aggregating with particle belonging to the I_k interval and generating a particle belonging to the I_i interval have pivots that respect relationship: $\zeta_i = \zeta_j + \zeta_k$
- When grid does not respect this relationship it is necessary to reassign newly generated particles to neighboring pivotal points
- Fraction of new particle is assigned to I_i interval ($\gamma_i^{(i)}$) and another fraction to I_{i+1} ($\gamma_{i+1}^{(i)}$)
- Preservation of total number of particles

$$1 = \gamma_i^{(i)} + \gamma_{i+1}^{(i)}$$

- Preservation of total mass

$$\zeta_j + \zeta_k = \gamma_i^{(i)} \zeta_i + \gamma_{i+1}^{(i)} \zeta_{i+1}$$



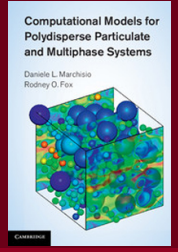
Class Method (Univariate)

- Equations are identical, and therefore method is **internally consistent** if and only if chosen grid is such that a particle belonging to the I_j interval aggregating with particle belonging to the I_k interval and generating a particle belonging to the I_i interval have pivots that respect relationship: $\zeta_i = \zeta_j + \zeta_k$
- When grid does not respect this relationship it is necessary to reassign newly generated particles to neighboring pivotal points
- Fraction of new particle is assigned to I_i interval ($\gamma_i^{(i)}$) and another fraction to I_{i+1} ($\gamma_{i+1}^{(i)}$)
- Preservation of total number of particles

$$1 = \gamma_i^{(i)} + \gamma_{i+1}^{(i)}$$

- Preservation of total mass

$$\zeta_j + \zeta_k = \gamma_i^{(i)} \zeta_i + \gamma_{i+1}^{(i)} \zeta_{i+1}$$



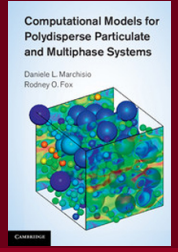
Class Method (Univariate)

- Equations are identical, and therefore method is **internally consistent** if and only if chosen grid is such that a particle belonging to the I_j interval aggregating with particle belonging to the I_k interval and generating a particle belonging to the I_i interval have pivots that respect relationship: $\zeta_i = \zeta_j + \zeta_k$
- When grid does not respect this relationship it is necessary to reassign newly generated particles to neighboring pivotal points
- Fraction of new particle is assigned to I_i interval ($\gamma_i^{(i)}$) and another fraction to I_{i+1} ($\gamma_{i+1}^{(i)}$)
- Preservation of total number of particles

$$1 = \gamma_i^{(i)} + \gamma_{i+1}^{(i)}$$

- Preservation of total mass

$$\zeta_j + \zeta_k = \gamma_i^{(i)} \zeta_i + \gamma_{i+1}^{(i)} \zeta_{i+1}$$



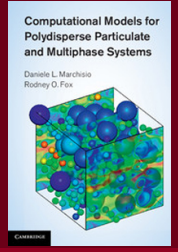
Class Method (Univariate)

- Equations are identical, and therefore method is **internally consistent** if and only if chosen grid is such that a particle belonging to the I_j interval aggregating with particle belonging to the I_k interval and generating a particle belonging to the I_i interval have pivots that respect relationship: $\zeta_i = \zeta_j + \zeta_k$
- When grid does not respect this relationship it is necessary to reassign newly generated particles to neighboring pivotal points
- Fraction of new particle is assigned to I_i interval ($\gamma_i^{(i)}$) and another fraction to I_{i+1} ($\gamma_{i+1}^{(i)}$)
- Preservation of total number of particles

$$1 = \gamma_i^{(i)} + \gamma_{i+1}^{(i)}$$

- Preservation of total mass

$$\zeta_j + \zeta_k = \gamma_i^{(i)} \zeta_i + \gamma_{i+1}^{(i)} \zeta_{i+1}$$



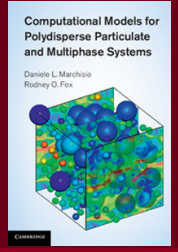
Class Method (Univariate)

- Equations are identical, and therefore method is **internally consistent** if and only if chosen grid is such that a particle belonging to the I_j interval aggregating with particle belonging to the I_k interval and generating a particle belonging to the I_i interval have pivots that respect relationship: $\zeta_i = \zeta_j + \zeta_k$
- When grid does not respect this relationship it is necessary to reassign newly generated particles to neighboring pivotal points
- Fraction of new particle is assigned to I_i interval ($\gamma_i^{(i)}$) and another fraction to I_{i+1} ($\gamma_{i+1}^{(i)}$)
- Preservation of total number of particles

$$1 = \gamma_i^{(i)} + \gamma_{i+1}^{(i)}$$

- Preservation of total mass

$$\zeta_j + \zeta_k = \gamma_i^{(i)} \zeta_i + \gamma_{i+1}^{(i)} \zeta_{i+1}$$



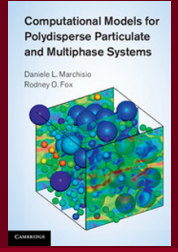
Class Method (Univariate)

- Equations are identical, and therefore method is **internally consistent** if and only if chosen grid is such that a particle belonging to the I_j interval aggregating with particle belonging to the I_k interval and generating a particle belonging to the I_i interval have pivots that respect relationship: $\zeta_i = \zeta_j + \zeta_k$
- When grid does not respect this relationship it is necessary to reassign newly generated particles to neighboring pivotal points
- Fraction of new particle is assigned to I_i interval ($\gamma_i^{(i)}$) and another fraction to I_{i+1} ($\gamma_{i+1}^{(i)}$)
- Preservation of total number of particles

$$1 = \gamma_i^{(i)} + \gamma_{i+1}^{(i)}$$

- Preservation of total mass

$$\zeta_j + \zeta_k = \gamma_i^{(i)} \zeta_i + \gamma_{i+1}^{(i)} \zeta_{i+1}$$



Class Method (Univariate)

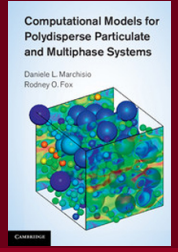
- Preservation of r^{th}-order moment results in following reassignment equation: $(\zeta_j + \zeta_k)^r = \gamma_i^{(i)} (\zeta_i)^r + \gamma_{i+1}^{(i)} (\zeta_{i+1})^r$,
- Two degrees of freedom are available, values of $\gamma_i^{(i)}$ and $\gamma_{i+1}^{(i)}$ are chosen in order to preserve two different moments of distribution
- For preserving moment of order zero, m_0, and moment of order one, m_1

$$\gamma_i^{(i)} = \frac{\zeta_{i+1} - (\zeta_j + \zeta_k)}{\zeta_{i+1} - \zeta_i}, \quad \gamma_{i+1}^{(i)} = \frac{(\zeta_j + \zeta_k) - \zeta_i}{\zeta_{i+1} - \zeta_i}$$

- **This approach appears internally consistent only for two chosen moments of the NDF!**
- For breakage condition for internal consistency is

$$\int_{\xi_i}^{\xi_{i+1}} \xi^r b(\xi|\xi_j)\, \mathrm{d}\xi = \zeta_i{}^r \int_{\xi_i}^{\xi_{i+1}} b(\xi|\xi_j)\, \mathrm{d}\xi$$

with $\zeta_i > \zeta_j$ and where $b(\xi|\xi_j)$ is daughter distribution function



Class Method (Univariate)

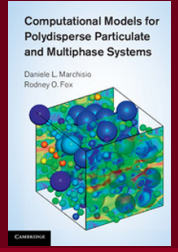
- Preservation of r^{th}-order moment results in following reassignment equation: $(\zeta_j + \zeta_k)^r = \gamma_i^{(i)} (\zeta_i)^r + \gamma_{i+1}^{(i)} (\zeta_{i+1})^r$,
- Two degrees of freedom are available, values of $\gamma_i^{(i)}$ and $\gamma_{i+1}^{(i)}$ are chosen in order to preserve two different moments of distribution
- For preserving moment of order zero, m_0, and moment of order one, m_1

$$\gamma_i^{(i)} = \frac{\zeta_{i+1} - (\zeta_j + \zeta_k)}{\zeta_{i+1} - \zeta_i}, \quad \gamma_{i+1}^{(i)} = \frac{(\zeta_j + \zeta_k) - \zeta_i}{\zeta_{i+1} - \zeta_i}$$

- **This approach appears internally consistent only for two chosen moments of the NDF!**
- For breakage condition for internal consistency is

$$\int_{\xi_i}^{\xi_{i+1}} \xi^r b(\xi|\xi_j)\, \mathrm{d}\xi = \zeta_i{}^r \int_{\xi_i}^{\xi_{i+1}} b(\xi|\xi_j)\, \mathrm{d}\xi$$

with $\zeta_i > \zeta_j$ and where $b(\xi|\xi_j)$ is daughter distribution function



Class Method (Univariate)

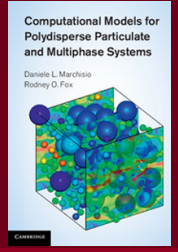
- Preservation of r^{th}-order moment results in following reassignment equation: $(\zeta_j + \zeta_k)^r = \gamma_i^{(i)} (\zeta_i)^r + \gamma_{i+1}^{(i)} (\zeta_{i+1})^r$,
- Two degrees of freedom are available, values of $\gamma_i^{(i)}$ and $\gamma_{i+1}^{(i)}$ are chosen in order to preserve two different moments of distribution
- For preserving moment of order zero, m_0, and moment of order one, m_1

$$\gamma_i^{(i)} = \frac{\zeta_{i+1} - (\zeta_j + \zeta_k)}{\zeta_{i+1} - \zeta_i}, \quad \gamma_{i+1}^{(i)} = \frac{(\zeta_j + \zeta_k) - \zeta_i}{\zeta_{i+1} - \zeta_i}$$

- **This approach appears internally consistent only for two chosen moments of the NDF!**
- For breakage condition for internal consistency is

$$\int_{\xi_i}^{\xi_{i+1}} \xi^r b(\xi|\xi_j)\, \mathrm{d}\xi = \zeta_i{}^r \int_{\xi_i}^{\xi_{i+1}} b(\xi|\xi_j)\, \mathrm{d}\xi$$

with $\zeta_i > \zeta_j$ and where $b(\xi|\xi_j)$ is daughter distribution function



Class Method (Univariate)

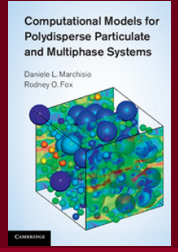
- Preservation of r^{th}-order moment results in following reassignment equation: $(\zeta_j + \zeta_k)^r = \gamma_i^{(i)} (\zeta_i)^r + \gamma_{i+1}^{(i)} (\zeta_{i+1})^r$,
- Two degrees of freedom are available, values of $\gamma_i^{(i)}$ and $\gamma_{i+1}^{(i)}$ are chosen in order to preserve two different moments of distribution
- For preserving moment of order zero, m_0, and moment of order one, m_1

$$\gamma_i^{(i)} = \frac{\zeta_{i+1} - (\zeta_j + \zeta_k)}{\zeta_{i+1} - \zeta_i}, \quad \gamma_{i+1}^{(i)} = \frac{(\zeta_j + \zeta_k) - \zeta_i}{\zeta_{i+1} - \zeta_i}$$

- **This approach appears internally consistent only for two chosen moments of the NDF!**
- For breakage condition for internal consistency is

$$\int_{\xi_i}^{\xi_{i+1}} \xi^r b(\xi|\xi_j)\, \mathrm{d}\xi = \zeta_i{}^r \int_{\xi_i}^{\xi_{i+1}} b(\xi|\xi_j)\, \mathrm{d}\xi$$

with $\zeta_i > \zeta_j$ and where $b(\xi|\xi_j)$ is daughter distribution function



Class Method (Univariate)

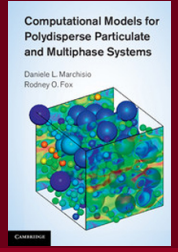
- Preservation of r^{th}-order moment results in following reassignment equation: $(\zeta_j + \zeta_k)^r = \gamma_i^{(i)} (\zeta_i)^r + \gamma_{i+1}^{(i)} (\zeta_{i+1})^r$,
- Two degrees of freedom are available, values of $\gamma_i^{(i)}$ and $\gamma_{i+1}^{(i)}$ are chosen in order to preserve two different moments of distribution
- For preserving moment of order zero, m_0, and moment of order one, m_1

$$\gamma_i^{(i)} = \frac{\zeta_{i+1} - (\zeta_j + \zeta_k)}{\zeta_{i+1} - \zeta_i}, \quad \gamma_{i+1}^{(i)} = \frac{(\zeta_j + \zeta_k) - \zeta_i}{\zeta_{i+1} - \zeta_i}$$

- **This approach appears internally consistent only for two chosen moments of the NDF!**
- For breakage condition for internal consistency is

$$\int_{\xi_i}^{\xi_{i+1}} \xi^r b(\xi|\xi_j)\, \mathrm{d}\xi = \zeta_i{}^r \int_{\xi_i}^{\xi_{i+1}} b(\xi|\xi_j)\, \mathrm{d}\xi$$

with $\zeta_i > \zeta_j$ and where $b(\xi|\xi_j)$ is daughter distribution function



Class Method (Univariate)

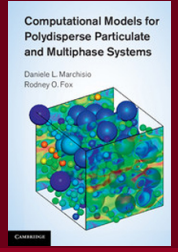
- Preservation of r^{th}-order moment results in following reassignment equation: $(\zeta_j + \zeta_k)^r = \gamma_i^{(i)} (\zeta_i)^r + \gamma_{i+1}^{(i)} (\zeta_{i+1})^r$,
- Two degrees of freedom are available, values of $\gamma_i^{(i)}$ and $\gamma_{i+1}^{(i)}$ are chosen in order to preserve two different moments of distribution
- For preserving moment of order zero, m_0, and moment of order one, m_1

$$\gamma_i^{(i)} = \frac{\zeta_{i+1} - (\zeta_j + \zeta_k)}{\zeta_{i+1} - \zeta_i}, \quad \gamma_{i+1}^{(i)} = \frac{(\zeta_j + \zeta_k) - \zeta_i}{\zeta_{i+1} - \zeta_i}$$

- **This approach appears internally consistent only for two chosen moments of the NDF!**
- For breakage condition for internal consistency is

$$\int_{\xi_i}^{\xi_{i+1}} \xi^r b(\xi|\xi_j)\, \mathrm{d}\xi = \zeta_i{}^r \int_{\xi_i}^{\xi_{i+1}} b(\xi|\xi_j)\, \mathrm{d}\xi$$

with $\zeta_i > \zeta_j$ and where $b(\xi|\xi_j)$ is daughter distribution function



Class Method (Univariate)

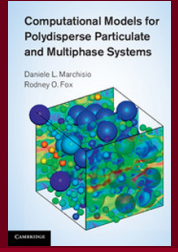
- **Linear grid**: $\xi_i = \frac{1}{2}ih$, $\xi_{i+1} = \frac{3}{2}ih$, $\zeta_i = ih$ for $i \in 1, \ldots, M$; scheme always consistent for aggregation
- Final equations for **fixed-pivot technique**

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\beta_{i-j,j}N_{i-j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j + \sum_{j=i+1}^{M}a_j b_{i,j}N_j - a_i N_i$$

- **Geometrical grid**: $\xi_i = \frac{3}{4}\zeta_i$, $\xi_{i+1} = \frac{3}{2}\zeta_i$, $\zeta_{i+1} = 2\zeta_i$, for $i \in 1, \ldots, M-1$
- Final equations (for aggregation only)

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = N_{i-1}\sum_{j=1}^{i-1}2^{j-i+1}\beta_{i-1,j}N_j + \frac{1}{2}\beta_{i-1,j-1}N_{i-1}^2$$

$$- N_i\sum_{j=1}^{i-1}2^{j-i}\beta_{i,j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j$$



Class Method (Univariate)

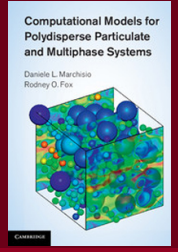
- **Linear grid**: $\xi_i = \frac{1}{2}ih$, $\xi_{i+1} = \frac{3}{2}ih$, $\zeta_i = ih$ for $i \in 1, \ldots, M$; scheme always consistent for aggregation
- Final equations for **fixed-pivot technique**

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\beta_{i-j,j}N_{i-j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j + \sum_{j=i+1}^{M}a_j b_{i,j}N_j - a_i N_i$$

- **Geometrical grid**: $\xi_i = \frac{3}{4}\zeta_i$, $\xi_{i+1} = \frac{3}{2}\zeta_i$, $\zeta_{i+1} = 2\zeta_i$, for $i \in 1, \ldots, M-1$
- Final equations (for aggregation only)

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = N_{i-1}\sum_{j=1}^{i-1}2^{j-i+1}\beta_{i-1,j}N_j + \frac{1}{2}\beta_{i-1,j-1}N_{i-1}^2$$

$$- N_i\sum_{j=1}^{i-1}2^{j-i}\beta_{i,j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j$$



Class Method (Univariate)

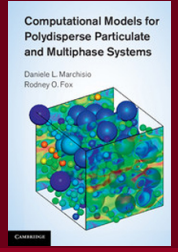
- **Linear grid**: $\xi_i = \frac{1}{2}ih$, $\xi_{i+1} = \frac{3}{2}ih$, $\zeta_i = ih$ for $i \in 1, \ldots, M$; scheme always consistent for aggregation
- Final equations for **fixed-pivot technique**

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\beta_{i-j,j}N_{i-j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j + \sum_{j=i+1}^{M}a_j b_{i,j}N_j - a_i N_i$$

- **Geometrical grid**: $\xi_i = \frac{3}{4}\zeta_i$, $\xi_{i+1} = \frac{3}{2}\zeta_i$, $\zeta_{i+1} = 2\zeta_i$, for $i \in 1, \ldots, M-1$
- Final equations (for aggregation only)

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = N_{i-1}\sum_{j=1}^{i-1}2^{j-i+1}\beta_{i-1,j}N_j + \frac{1}{2}\beta_{i-1,j-1}N_{i-1}^2$$

$$- N_i\sum_{j=1}^{i-1}2^{j-i}\beta_{i,j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j$$



Class Method (Univariate)

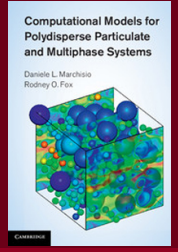
- **Linear grid**: $\xi_i = \frac{1}{2}ih$, $\xi_{i+1} = \frac{3}{2}ih$, $\zeta_i = ih$ for $i \in 1, \ldots, M$; scheme always consistent for aggregation
- Final equations for **fixed-pivot technique**

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\beta_{i-j,j}N_{i-j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j + \sum_{j=i+1}^{M}a_j b_{i,j}N_j - a_i N_i$$

- **Geometrical grid**: $\xi_i = \frac{3}{4}\zeta_i$, $\xi_{i+1} = \frac{3}{2}\zeta_i$, $\zeta_{i+1} = 2\zeta_i$, for $i \in 1, \ldots, M-1$
- Final equations (for aggregation only)

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = N_{i-1}\sum_{j=1}^{i-1}2^{j-i+1}\beta_{i-1,j}N_j + \frac{1}{2}\beta_{i-1,j-1}N_{i-1}^2$$

$$- N_i\sum_{j=1}^{i-1}2^{j-i}\beta_{i,j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j$$



Class Method (Univariate)

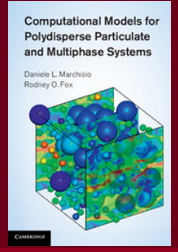
- **Linear grid**: $\xi_i = \frac{1}{2}ih$, $\xi_{i+1} = \frac{3}{2}ih$, $\zeta_i = ih$ for $i \in 1, \ldots, M$; scheme always consistent for aggregation
- Final equations for **fixed-pivot technique**

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \frac{1}{2}\sum_{j=1}^{i-1}\beta_{i-j,j}N_{i-j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j + \sum_{j=i+1}^{M}a_j b_{i,j}N_j - a_i N_i$$

- **Geometrical grid**: $\xi_i = \frac{3}{4}\zeta_i$, $\xi_{i+1} = \frac{3}{2}\zeta_i$, $\zeta_{i+1} = 2\zeta_i$, for $i \in 1, \ldots, M-1$
- Final equations (for aggregation only)

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = N_{i-1}\sum_{j=1}^{i-1}2^{j-i+1}\beta_{i-1,j}N_j + \frac{1}{2}\beta_{i-1,j-1}N_{i-1}^2$$

$$- N_i\sum_{j=1}^{i-1}2^{j-i}\beta_{i,j}N_j - N_i\sum_{j=1}^{M}\beta_{i,j}N_j$$



Class Method (Univariate)

Fixed-pivot technique can, in general, be formulated for any grid while preserving two moments of distribution of orders r_1 and r_2

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \sum_{j=i}^{M} N_j a_j \pi_{i,j} - a_i N_i + \sum_{\zeta_{i-1} \le (\zeta_j+\zeta_k) < \zeta_i}^{j \ge k} \left(1 - \frac{1}{2}\delta_{jk}\right)\left[\gamma_i^{(i-1)}(\zeta_j+\zeta_k)\beta_{j,k}N_jN_k\right]$$

$$+ \sum_{\zeta_i \le (\zeta_j+\zeta_k) < \zeta_{i+1}}^{j \ge k} \left(1 - \frac{1}{2}\delta_{jk}\right)\left[\gamma_i^{(i)}(\zeta_j+\zeta_k)\beta_{j,k}N_jN_k\right] - N_i \sum_{j=1}^{M} \beta_{i,j}N_j$$

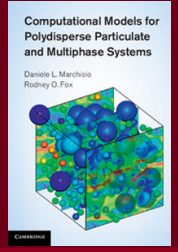
where

$$\gamma_i^{(i)} = \frac{(\zeta_j+\zeta_k)^{r_1}(\zeta_{i+1})^{r_2} - (\zeta_j+\zeta_k)^{r_2}(\zeta_{i+1})^{r_1}}{(\zeta_i)^{r_1}(\zeta_{i+1})^{r_2} - (\zeta_i)^{r_2}(\zeta_{i+1})^{r_1}}$$

$$\gamma_i^{(i+1)} = \frac{(\zeta_j+\zeta_k)^{r_2}(\zeta_i)^{r_1} - (\zeta_j+\zeta_k)^{r_1}(\zeta_i)^{r_2}}{(\zeta_i)^{r_1}(\zeta_{i+1})^{r_2} - (\zeta_i)^{r_2}(\zeta_{i+1})^{r_1}}$$

and

$$\pi_{i,j} = \int_{\zeta_{i-1}}^{\zeta_i} \gamma_i^{(i-1)} b(\xi|\xi_j)\,\mathrm{d}\xi + \int_{\zeta_i}^{\zeta_{i+1}} \gamma_i^{(i-1)} b(\xi|\xi_j)\,\mathrm{d}\xi$$



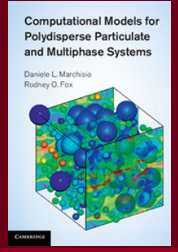
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



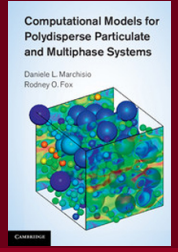
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



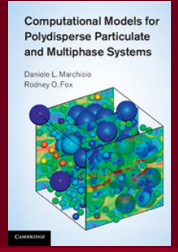
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



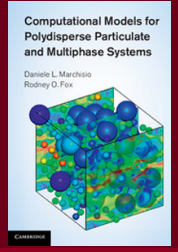
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



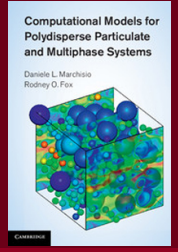
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



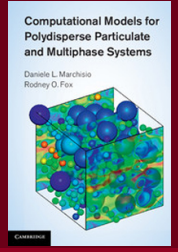
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



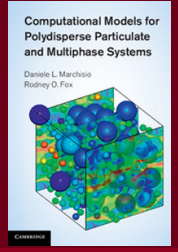
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



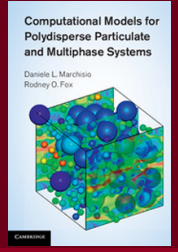
Class Method (Univariate)

ADVANTAGES

- Very intuitive (easy to develop tricks to solve problems)
- Generally stable

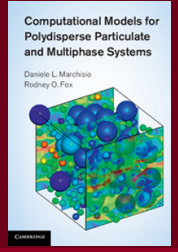
DISADVANTAGES

- Method is consistent with respect to only TWO MOMENTS of NDF: particle mass and number
- One has to define a priori lower and upper limits of discretization
- Impossible to study for example: gelation or shattering
- Many (too many) classes to be used (e.g. 20–100): problematic especially with CFD codes where CPU time scales linearly with number of additional transport equations
- Very inefficient (each class does not know what others are doing)



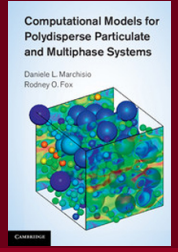
Monte Carlo Methods

- Monte Carlo methods or Direct Simulation Monte Carlo (DSMC) are based on *artificial realizations* of behavior of multiphase system under investigation
- In MC methods evolution of each particle is tracked; continuous events are represented deterministically whereas discontinuous events are represented by random events occurring with specific and prescribed probabilities
- Connection with deterministic approach used in class methods: underlying multi-particle joint PDF is not averaged but is directly solved
- Solution is obtained by adopting methods for stochastic differential equations, leading to numerous realizations, representing different *sample paths*, that must be averaged out to eliminate low probability events



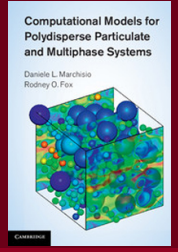
Monte Carlo Methods

- Monte Carlo methods or Direct Simulation Monte Carlo (DSMC) are based on *artificial realizations* of behavior of multiphase system under investigation
- In MC methods evolution of each particle is tracked; continuous events are represented deterministically whereas discontinuous events are represented by random events occurring with specific and prescribed probabilities
- Connection with deterministic approach used in class methods: underlying multi-particle joint PDF is not averaged but is directly solved
- Solution is obtained by adopting methods for stochastic differential equations, leading to numerous realizations, representing different *sample paths*, that must be averaged out to eliminate low probability events



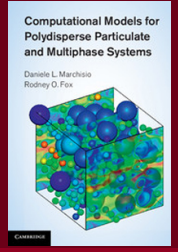
Monte Carlo Methods

- Monte Carlo methods or Direct Simulation Monte Carlo (DSMC) are based on *artificial realizations* of behavior of multiphase system under investigation
- In MC methods evolution of each particle is tracked; continuous events are represented deterministically whereas discontinuous events are represented by random events occurring with specific and prescribed probabilities
- Connection with deterministic approach used in class methods: underlying multi-particle joint PDF is not averaged but is directly solved
- Solution is obtained by adopting methods for stochastic differential equations, leading to numerous realizations, representing different *sample paths*, that must be averaged out to eliminate low probability events



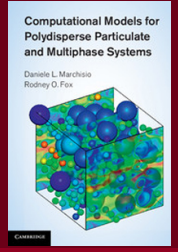
Monte Carlo Methods

- Monte Carlo methods or Direct Simulation Monte Carlo (DSMC) are based on *artificial realizations* of behavior of multiphase system under investigation
- In MC methods evolution of each particle is tracked; continuous events are represented deterministically whereas discontinuous events are represented by random events occurring with specific and prescribed probabilities
- Connection with deterministic approach used in class methods: underlying multi-particle joint PDF is not averaged but is directly solved
- Solution is obtained by adopting methods for stochastic differential equations, leading to numerous realizations, representing different *sample paths*, that must be averaged out to eliminate low probability events



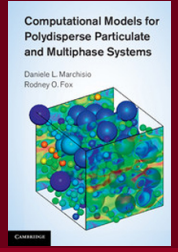
Monte Carlo Methods

- Monte Carlo methods or Direct Simulation Monte Carlo (DSMC) are based on *artificial realizations* of behavior of multiphase system under investigation
- In MC methods evolution of each particle is tracked; continuous events are represented deterministically whereas discontinuous events are represented by random events occurring with specific and prescribed probabilities
- Connection with deterministic approach used in class methods: underlying multi-particle joint PDF is not averaged but is directly solved
- Solution is obtained by adopting methods for stochastic differential equations, leading to numerous realizations, representing different *sample paths*, that must be averaged out to eliminate low probability events



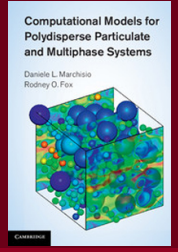
Monte Carlo Methods

- Group of N particles is randomly created (representing the given NDF) in computational domain or box and its evolution is directly tracked: time-driven simulations versus event-driven simulations
- After aggregation event for example $N - 1$ particles remain in domain, whereas after breakage event $N + 1$ particles have to be tracked
- As simulation proceeds total number of particles changes, whereas total volume (or total mass) of particles in computational domain remains constant
- For this reason these methods are called **constant-volume Monte Carlo**
- Problems if N becomes to small or too large: **constant-number Monte Carlo**
- Coupling with CFD is done via **compartment models**



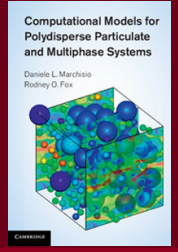
Monte Carlo Methods

- Group of N particles is randomly created (representing the given NDF) in computational domain or box and its evolution is directly tracked: time-driven simulations versus event-driven simulations
- After aggregation event for example $N-1$ particles remain in domain, whereas after breakage event $N+1$ particles have to be tracked
- As simulation proceeds total number of particles changes, whereas total volume (or total mass) of particles in computational domain remains constant
- For this reason these methods are called **constant-volume Monte Carlo**
- Problems if N becomes to small or too large: **constant-number Monte Carlo**
- Coupling with CFD is done via **compartment models**



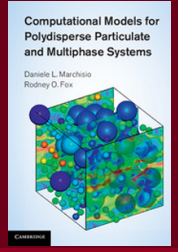
Monte Carlo Methods

- Group of N particles is randomly created (representing the given NDF) in computational domain or box and its evolution is directly tracked: time-driven simulations versus event-driven simulations
- After aggregation event for example $N - 1$ particles remain in domain, whereas after breakage event $N + 1$ particles have to be tracked
- As simulation proceeds total number of particles changes, whereas total volume (or total mass) of particles in computational domain remains constant
- For this reason these methods are called **constant-volume Monte Carlo**
- Problems if N becomes to small or too large: **constant-number Monte Carlo**
- Coupling with CFD is done via **compartment models**



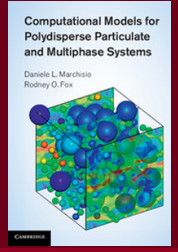
Monte Carlo Methods

- Group of N particles is randomly created (representing the given NDF) in computational domain or box and its evolution is directly tracked: time-driven simulations versus event-driven simulations
- After aggregation event for example $N - 1$ particles remain in domain, whereas after breakage event $N + 1$ particles have to be tracked
- As simulation proceeds total number of particles changes, whereas total volume (or total mass) of particles in computational domain remains constant
- For this reason these methods are called **constant-volume Monte Carlo**
- Problems if N becomes to small or too large: **constant-number Monte Carlo**
- Coupling with CFD is done via **compartment models**



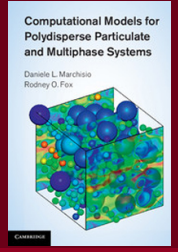
Monte Carlo Methods

- Group of N particles is randomly created (representing the given NDF) in computational domain or box and its evolution is directly tracked: time-driven simulations versus event-driven simulations
- After aggregation event for example $N - 1$ particles remain in domain, whereas after breakage event $N + 1$ particles have to be tracked
- As simulation proceeds total number of particles changes, whereas total volume (or total mass) of particles in computational domain remains constant
- For this reason these methods are called **constant-volume Monte Carlo**
- Problems if N becomes to small or too large: **constant-number Monte Carlo**
- Coupling with CFD is done via **compartment models**



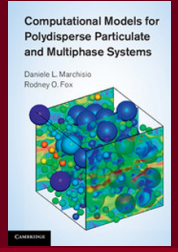
Monte Carlo Methods

- Group of N particles is randomly created (representing the given NDF) in computational domain or box and its evolution is directly tracked: time-driven simulations versus event-driven simulations
- After aggregation event for example $N - 1$ particles remain in domain, whereas after breakage event $N + 1$ particles have to be tracked
- As simulation proceeds total number of particles changes, whereas total volume (or total mass) of particles in computational domain remains constant
- For this reason these methods are called **constant-volume Monte Carlo**
- Problems if N becomes to small or too large: **constant-number Monte Carlo**
- Coupling with CFD is done via **compartment models**



Monte Carlo Methods

- Group of N particles is randomly created (representing the given NDF) in computational domain or box and its evolution is directly tracked: time-driven simulations versus event-driven simulations
- After aggregation event for example $N - 1$ particles remain in domain, whereas after breakage event $N + 1$ particles have to be tracked
- As simulation proceeds total number of particles changes, whereas total volume (or total mass) of particles in computational domain remains constant
- For this reason these methods are called **constant-volume Monte Carlo**
- Problems if N becomes to small or too large: **constant-number Monte Carlo**
- Coupling with CFD is done via **compartment models**



Method of Moments

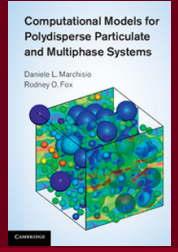
- Basic idea behind **Method of Moments** is to solve evolution equations for moments of NDF
- Generic moment of NDF is defined as

$$m_{k_1,k_2,\ldots,k_M} = m(\mathbf{k}) = \left\langle \xi_1^{k_1} \xi_2^{k_2} \cdots \xi_M^{k_M} \right\rangle := \int_{\Omega_\xi} n(\boldsymbol{\xi}) \xi_1^{k_1} \xi_2^{k_2} \cdots \xi_M^{k_M} \, \mathrm{d}\boldsymbol{\xi}$$

  where three different notations ($m_{k_1,k_2,\ldots,k_M}$, $m(\mathbf{k})$, $\langle \xi_1^{k_1} \xi_2^{k_2} \cdots \xi_M^{k_M} \rangle$) will be used interchangeably
- $\mathbf{k} = (k_1, k_2, \ldots, k_M)$ is called **exponent vector**
- General evolution equation for $\mathbf{k}^{\text{th}}$ moment

$$\frac{\mathrm{d}m_{\mathbf{k}}}{\mathrm{d}t} = \overline{S}_{\mathbf{k}} = \int \boldsymbol{\xi}^{\mathbf{k}} \left(-\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S} \right) \mathrm{d}\boldsymbol{\xi}$$

- By solving equations for moments, internal coordinates are integrated out, at expense of a crucial loss of information that must be retrieved (by reconstructing NDF for example)



Method of Moments

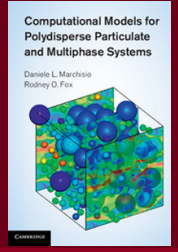
- Basic idea behind **Method of Moments** is to solve evolution equations for moments of NDF
- Generic moment of NDF is defined as

$$m_{k_1,k_2,\ldots,k_M} = m(\mathbf{k}) = \left\langle \xi_1^{k_1} \xi_2^{k_2} \cdots \xi_M^{k_M} \right\rangle := \int_{\Omega_\xi} n(\boldsymbol{\xi}) \xi_1^{k_1} \xi_2^{k_2} \cdots \xi_M^{k_M} \, \mathrm{d}\boldsymbol{\xi}$$

  where three different notations ($m_{k_1,k_2,\ldots,k_M}$, $m(\mathbf{k})$, $\langle \xi_1^{k_1} \xi_2^{k_2} \cdots \xi_M^{k_M} \rangle$) will be used interchangeably
- $\mathbf{k} = (k_1, k_2, \ldots, k_M)$ is called **exponent vector**
- General evolution equation for $\mathbf{k}^{\text{th}}$ moment

$$\frac{\mathrm{d}m_{\mathbf{k}}}{\mathrm{d}t} = \overline{S}_{\mathbf{k}} = \int \boldsymbol{\xi}^{\mathbf{k}} \left(-\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S} \right) \mathrm{d}\boldsymbol{\xi}$$

- By solving equations for moments, internal coordinates are integrated out, at expense of a crucial loss of information that must be retrieved (by reconstructing NDF for example)



Method of Moments

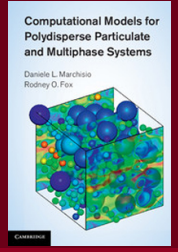
- Basic idea behind **Method of Moments** is to solve evolution equations for moments of NDF
- Generic moment of NDF is defined as

$$m_{k_1,k_2,\ldots,k_M} = m(\mathbf{k}) = \left\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \right\rangle := \int_{\Omega_\xi} n(\boldsymbol{\xi}) \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \, \mathrm{d}\boldsymbol{\xi}$$

  where three different notations ($m_{k_1,k_2,\ldots,k_M}$, $m(\mathbf{k})$, $\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \rangle$) will be used interchangeably
- $\mathbf{k} = (k_1, k_2, \ldots, k_M)$ is called **exponent vector**
- General evolution equation for $\mathbf{k}^{\text{th}}$ moment

$$\frac{\mathrm{d}m_{\mathbf{k}}}{\mathrm{d}t} = \overline{S}_{\mathbf{k}} = \int \boldsymbol{\xi}^{\mathbf{k}} \left(-\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S} \right) \mathrm{d}\boldsymbol{\xi}$$

- By solving equations for moments, internal coordinates are integrated out, at expense of a crucial loss of information that must be retrieved (by reconstructing NDF for example)



Method of Moments

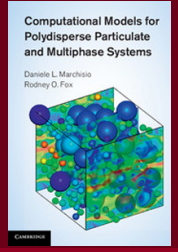
- Basic idea behind **Method of Moments** is to solve evolution equations for moments of NDF
- Generic moment of NDF is defined as

$$m_{k_1,k_2,\ldots,k_M} = m(\mathbf{k}) = \left\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \right\rangle := \int_{\Omega_\xi} n(\boldsymbol{\xi}) \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \, \mathrm{d}\boldsymbol{\xi}$$

  where three different notations ($m_{k_1,k_2,\ldots,k_M}$, $m(\mathbf{k})$, $\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \rangle$) will be used interchangeably
- $\mathbf{k} = (k_1, k_2, \ldots, k_M)$ is called **exponent vector**
- General evolution equation for $\mathbf{k}^{\text{th}}$ moment

$$\frac{\mathrm{d}m_{\mathbf{k}}}{\mathrm{d}t} = \overline{S}_{\mathbf{k}} = \int \boldsymbol{\xi}^{\mathbf{k}} \left(-\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S} \right) \mathrm{d}\boldsymbol{\xi}$$

- By solving equations for moments, internal coordinates are integrated out, at expense of a crucial loss of information that must be retrieved (by reconstructing NDF for example)



Method of Moments

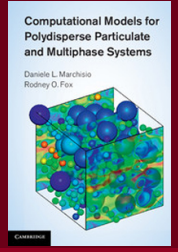
- Basic idea behind **Method of Moments** is to solve evolution equations for moments of NDF
- Generic moment of NDF is defined as

$$m_{k_1,k_2,\ldots,k_M} = m(\mathbf{k}) = \left\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \right\rangle := \int_{\Omega_\xi} n(\boldsymbol{\xi}) \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \, \mathrm{d}\boldsymbol{\xi}$$

  where three different notations ($m_{k_1,k_2,\ldots,k_M}$, $m(\mathbf{k})$, $\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \rangle$) will be used interchangeably
- $\mathbf{k} = (k_1, k_2, \ldots, k_M)$ is called **exponent vector**
- General evolution equation for $\mathbf{k}^{\text{th}}$ moment

$$\frac{\mathrm{d}m_{\mathbf{k}}}{\mathrm{d}t} = \overline{S}_{\mathbf{k}} = \int \boldsymbol{\xi}^{\mathbf{k}} \left(-\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S} \right) \mathrm{d}\boldsymbol{\xi}$$

- By solving equations for moments, internal coordinates are integrated out, at expense of a crucial loss of information that must be retrieved (by reconstructing NDF for example)



Method of Moments

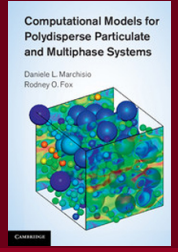
- Basic idea behind **Method of Moments** is to solve evolution equations for moments of NDF
- Generic moment of NDF is defined as

$$m_{k_1,k_2,\ldots,k_M} = m(\mathbf{k}) = \left\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \right\rangle := \int_{\Omega_\xi} n(\boldsymbol{\xi}) \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \, \mathrm{d}\boldsymbol{\xi}$$

  where three different notations ($m_{k_1,k_2,\ldots,k_M}$, $m(\mathbf{k})$, $\langle \xi_1^{k_1} \xi_2^{k_2} \ldots \xi_M^{k_M} \rangle$) will be used interchangeably
- $\mathbf{k} = (k_1, k_2, \ldots, k_M)$ is called **exponent vector**
- General evolution equation for $\mathbf{k}^{\text{th}}$ moment

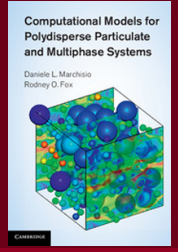
$$\frac{\mathrm{d}m_{\mathbf{k}}}{\mathrm{d}t} = \overline{S}_{\mathbf{k}} = \int \boldsymbol{\xi}^{\mathbf{k}} \left(-\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + \frac{\partial^2}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}} : \mathbf{D} n + \mathcal{S} \right) \mathrm{d}\boldsymbol{\xi}$$

- By solving equations for moments, internal coordinates are integrated out, at expense of a crucial loss of information that must be retrieved (by reconstructing NDF for example)



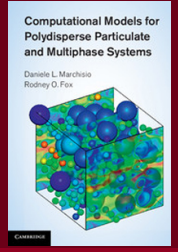
Method of Moments

- Important advantage is that moments correspond to quantities that have meaningful physical interpretations and are therefore directly measurable
- This is a crucial point, since in many applications the NDF is not directly measured, but is inferred from measurements of integral quantities
- Two main issues arise when using MOM: **number of moments** to be tracked and **closure problem**
- Closure problem is impossibility of writing source term as function of moments only
- Issues of number of moments to be tracked and closure adopted are typically addressed together



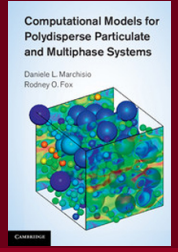
Method of Moments

- Important advantage is that moments correspond to quantities that have meaningful physical interpretations and are therefore directly measurable
- This is a crucial point, since in many applications the NDF is not directly measured, but is inferred from measurements of integral quantities
- Two main issues arise when using MOM: **number of moments** to be tracked and **closure problem**
- Closure problem is impossibility of writing source term as function of moments only
- Issues of number of moments to be tracked and closure adopted are typically addressed together



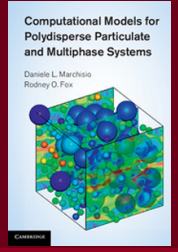
Method of Moments

- Important advantage is that moments correspond to quantities that have meaningful physical interpretations and are therefore directly measurable
- This is a crucial point, since in many applications the NDF is not directly measured, but is inferred from measurements of integral quantities
- Two main issues arise when using MOM: **number of moments** to be tracked and **closure problem**
- Closure problem is impossibility of writing source term as function of moments only
- Issues of number of moments to be tracked and closure adopted are typically addressed together



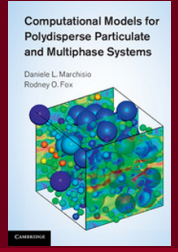
Method of Moments

- Important advantage is that moments correspond to quantities that have meaningful physical interpretations and are therefore directly measurable
- This is a crucial point, since in many applications the NDF is not directly measured, but is inferred from measurements of integral quantities
- Two main issues arise when using MOM: **number of moments** to be tracked and **closure problem**
- Closure problem is impossibility of writing source term as function of moments only
- Issues of number of moments to be tracked and closure adopted are typically addressed together



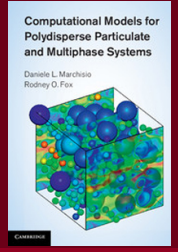
Method of Moments

- Important advantage is that moments correspond to quantities that have meaningful physical interpretations and are therefore directly measurable
- This is a crucial point, since in many applications the NDF is not directly measured, but is inferred from measurements of integral quantities
- Two main issues arise when using MOM: **number of moments** to be tracked and **closure problem**
- Closure problem is impossibility of writing source term as function of moments only
- Issues of number of moments to be tracked and closure adopted are typically addressed together



Method of Moments

- Important advantage is that moments correspond to quantities that have meaningful physical interpretations and are therefore directly measurable
- This is a crucial point, since in many applications the NDF is not directly measured, but is inferred from measurements of integral quantities
- Two main issues arise when using MOM: **number of moments** to be tracked and **closure problem**
- Closure problem is impossibility of writing source term as function of moments only
- Issues of number of moments to be tracked and closure adopted are typically addressed together



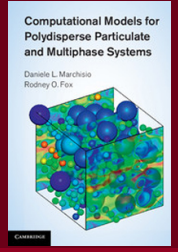
Method of Moments (Univariate)

Univariate PBE

- Evolution equation for moment of order k ($m_k := \int \xi^k n \, \mathrm{d}\xi$)

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \int \xi^k \left(-\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S} \right) \mathrm{d}\xi$$

- In case of univariate distributions moment of order zero m_0 is total particle number density
- If internal coordinate is particle size, then m_1 is total particle length density (number-average mean particle size: $d_{10} = m_1/m_0$)
- Moment of order two m_2 is proportional to total particle surface area per unit volume
- Moment of order three m_3 is related to total particle volume per unit volume (Sauter mean diameter: $d_{32} = m_3/m_2$)
- Mean particle size weighted on particle volume: $d_{43} = m_4/m_3$



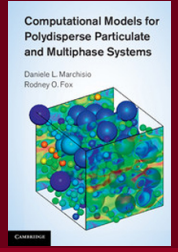
Method of Moments (Univariate)

Univariate PBE

- Evolution equation for moment of order k ($m_k := \int \xi^k n \, \mathrm{d}\xi$)

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \int \xi^k \left(-\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S} \right) \mathrm{d}\xi$$

- In case of univariate distributions moment of order zero m_0 is total particle number density
- If internal coordinate is particle size, then m_1 is total particle length density (number-average mean particle size: $d_{10} = m_1/m_0$)
- Moment of order two m_2 is proportional to total particle surface area per unit volume
- Moment of order three m_3 is related to total particle volume per unit volume (Sauter mean diameter: $d_{32} = m_3/m_2$)
- Mean particle size weighted on particle volume: $d_{43} = m_4/m_3$



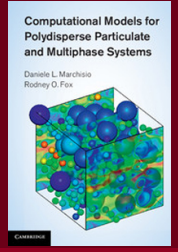
Method of Moments (Univariate)

Univariate PBE

- Evolution equation for moment of order k ($m_k := \int \xi^k n \, \mathrm{d}\xi$)

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \int \xi^k \left(-\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S} \right) \mathrm{d}\xi$$

- In case of univariate distributions moment of order zero m_0 is total particle number density
- If internal coordinate is particle size, then m_1 is total particle length density (number-average mean particle size: $d_{10} = m_1/m_0$)
- Moment of order two m_2 is proportional to total particle surface area per unit volume
- Moment of order three m_3 is related to total particle volume per unit volume (Sauter mean diameter: $d_{32} = m_3/m_2$)
- Mean particle size weighted on particle volume: $d_{43} = m_4/m_3$



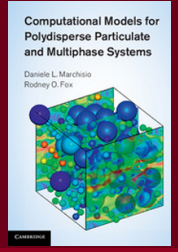
Method of Moments (Univariate)

Univariate PBE

- Evolution equation for moment of order k ($m_k := \int \xi^k n \, \mathrm{d}\xi$)

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \int \xi^k \left(-\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S} \right) \mathrm{d}\xi$$

- In case of univariate distributions moment of order zero m_0 is total particle number density
- If internal coordinate is particle size, then m_1 is total particle length density (number-average mean particle size: $d_{10} = m_1/m_0$)
- Moment of order two m_2 is proportional to total particle surface area per unit volume
- Moment of order three m_3 is related to total particle volume per unit volume (Sauter mean diameter: $d_{32} = m_3/m_2$)
- Mean particle size weighted on particle volume: $d_{43} = m_4/m_3$



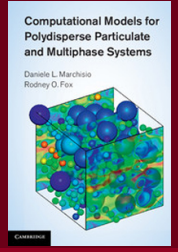
Method of Moments (Univariate)

Univariate PBE

- Evolution equation for moment of order k ($m_k := \int \xi^k n \, \mathrm{d}\xi$)

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \int \xi^k \left(-\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S} \right) \mathrm{d}\xi$$

- In case of univariate distributions moment of order zero m_0 is total particle number density
- If internal coordinate is particle size, then m_1 is total particle length density (number-average mean particle size: $d_{10} = m_1/m_0$)
- Moment of order two m_2 is proportional to total particle surface area per unit volume
- Moment of order three m_3 is related to total particle volume per unit volume (Sauter mean diameter: $d_{32} = m_3/m_2$)
- Mean particle size weighted on particle volume: $d_{43} = m_4/m_3$



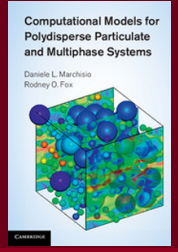
Method of Moments (Univariate)

Univariate PBE

- Evolution equation for moment of order k ($m_k := \int \xi^k n \, \mathrm{d}\xi$)

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \int \xi^k \left(-\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S} \right) \mathrm{d}\xi$$

- In case of univariate distributions moment of order zero m_0 is total particle number density
- If internal coordinate is particle size, then m_1 is total particle length density (number-average mean particle size: $d_{10} = m_1/m_0$)
- Moment of order two m_2 is proportional to total particle surface area per unit volume
- Moment of order three m_3 is related to total particle volume per unit volume (Sauter mean diameter: $d_{32} = m_3/m_2$)
- Mean particle size weighted on particle volume: $d_{43} = m_4/m_3$



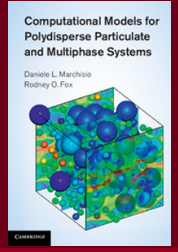
Method of Moments (Univariate)

Univariate PBE

- Evolution equation for moment of order k ($m_k := \int \xi^k n \, \mathrm{d}\xi$)

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \int \xi^k \left(-\frac{\partial \dot{\xi} n}{\partial \xi} + \frac{\partial^2 D n}{\partial \xi^2} + \mathcal{S} \right) \mathrm{d}\xi$$

- In case of univariate distributions moment of order zero m_0 is total particle number density
- If internal coordinate is particle size, then m_1 is total particle length density (number-average mean particle size: $d_{10} = m_1/m_0$)
- Moment of order two m_2 is proportional to total particle surface area per unit volume
- Moment of order three m_3 is related to total particle volume per unit volume (Sauter mean diameter: $d_{32} = m_3/m_2$)
- Mean particle size weighted on particle volume: $d_{43} = m_4/m_3$



Method of Moments (Univariate)

Univariate PBE: closure problem for aggregation

- In case of aggregation (mass as internal coordinate ξ)

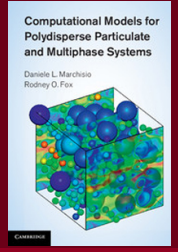
$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\int\!\!\int_0^\infty \beta(\xi,\xi')\,(\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \int_0^\infty \xi^k n(\xi)\int_0^\infty \beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi, \quad k\in 0,\ldots,N$$

- In general it is impossible to close this set of equations
- Constant aggregation kernel $\beta(\xi,\xi') = \beta_0$

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{\beta_0}{2}\int\!\!\int_0^\infty (\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \beta_0 m_0 m_k$$

- General solution (for const. aggregation kernel)

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta_0 m_0^2, \quad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\beta_0 \sum_{i=1}^{k-1}\binom{k}{i} m_i m_{k-i}$$



Method of Moments (Univariate)

Univariate PBE: closure problem for aggregation

- In case of aggregation (mass as internal coordinate ξ)

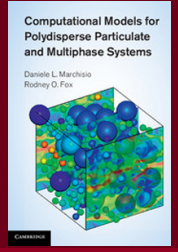
$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\int\!\!\int_0^\infty \beta(\xi,\xi')\,(\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \int_0^\infty \xi^k n(\xi)\int_0^\infty \beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi, \quad k\in 0,\ldots,N$$

- In general it is impossible to close this set of equations
- Constant aggregation kernel $\beta(\xi,\xi') = \beta_0$

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{\beta_0}{2}\int\!\!\int_0^\infty (\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \beta_0 m_0 m_k$$

- General solution (for const. aggregation kernel)

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta_0 m_0^2, \quad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\beta_0 \sum_{i=1}^{k-1}\binom{k}{i} m_i m_{k-i}$$



Method of Moments (Univariate)

Univariate PBE: closure problem for aggregation

- In case of aggregation (mass as internal coordinate ξ)

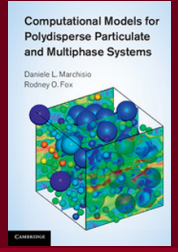
$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\int\!\!\int_0^\infty \beta(\xi,\xi')\,(\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \int_0^\infty \xi^k n(\xi)\int_0^\infty \beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi, \quad k\in 0,\ldots,N$$

- In general it is impossible to close this set of equations
- Constant aggregation kernel $\beta(\xi,\xi') = \beta_0$

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{\beta_0}{2}\int\!\!\int_0^\infty (\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \beta_0 m_0 m_k$$

- General solution (for const. aggregation kernel)

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta_0 m_0^2, \quad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\beta_0 \sum_{i=1}^{k-1}\binom{k}{i} m_i m_{k-i}$$



Method of Moments (Univariate)

Univariate PBE: closure problem for aggregation

- In case of aggregation (mass as internal coordinate ξ)

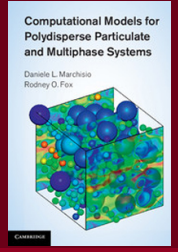
$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\int\!\!\int_0^\infty \beta(\xi,\xi')\,(\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \int_0^\infty \xi^k n(\xi)\int_0^\infty \beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi, \quad k\in 0,\ldots,N$$

- In general it is impossible to close this set of equations
- Constant aggregation kernel $\beta(\xi,\xi') = \beta_0$

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{\beta_0}{2}\int\!\!\int_0^\infty (\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \beta_0 m_0 m_k$$

- General solution (for const. aggregation kernel)

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta_0 m_0^2, \quad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\beta_0 \sum_{i=1}^{k-1}\binom{k}{i} m_i m_{k-i}$$



Method of Moments (Univariate)

Univariate PBE: closure problem for aggregation

- In case of aggregation (mass as internal coordinate ξ)

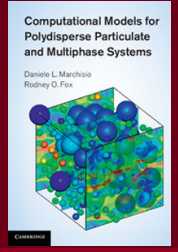
$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\int\!\!\int_0^\infty \beta(\xi,\xi')\,(\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \int_0^\infty \xi^k n(\xi)\int_0^\infty \beta(\xi,\xi')n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi, \quad k\in 0,\ldots,N$$

- In general it is impossible to close this set of equations
- Constant aggregation kernel $\beta(\xi,\xi') = \beta_0$

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{\beta_0}{2}\int\!\!\int_0^\infty (\xi+\xi')^k\, n(\xi)n(\xi')\,\mathrm{d}\xi\,\mathrm{d}\xi' - \beta_0 m_0 m_k$$

- General solution (for const. aggregation kernel)

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta_0 m_0^2, \quad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0, \quad \frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{1}{2}\beta_0 \sum_{i=1}^{k-1}\binom{k}{i} m_i m_{k-i}$$



Method of Moments (Univariate)

Univariate PBE: closure problem for breakage

- In case of pure breakage (particle mass as internal coordinate)

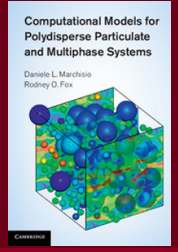
$$\frac{\partial n}{\partial t} = \int_{\xi}^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi' - a(\xi)n(\xi)$$

- Evolution equation for moment of order k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \int_0^{\infty} \xi^k \int_0^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi - \int_0^{\infty} \xi^k a(\xi)n(\xi)\,\mathrm{d}\xi$$

- Analytical solutions available for constant breakage kernel: $a(\xi) = a_0$ and symmetric fragmentation $b(\xi'|\xi) = 2\delta\left(\xi - \xi'/2\right)$
- Evolution equation for m_k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = a_0 2^{1-k} m_k - a_0 m_k$$



Method of Moments (Univariate)

Univariate PBE: closure problem for breakage

- In case of pure breakage (particle mass as internal coordinate)

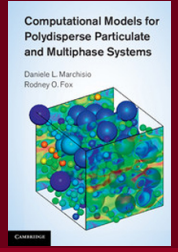
$$\frac{\partial n}{\partial t} = \int_{\xi}^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi' - a(\xi)n(\xi)$$

- Evolution equation for moment of order k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \int_0^{\infty} \xi^k \int_0^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi'\,\mathrm{d}\xi - \int_0^{\infty} \xi^k a(\xi)n(\xi)\,\mathrm{d}\xi$$

- Analytical solutions available for constant breakage kernel: $a(\xi) = a_0$ and symmetric fragmentation $b(\xi'|\xi) = 2\delta\left(\xi - \xi'/2\right)$
- Evolution equation for m_k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = a_0 2^{1-k} m_k - a_0 m_k$$



Method of Moments (Univariate)

Univariate PBE: closure problem for breakage

- In case of pure breakage (particle mass as internal coordinate)

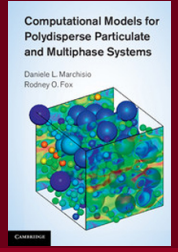
$$\frac{\partial n}{\partial t} = \int_{\xi}^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi' - a(\xi)n(\xi)$$

- Evolution equation for moment of order k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \int_0^{\infty} \xi^k \int_0^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi'\mathrm{d}\xi - \int_0^{\infty} \xi^k a(\xi)n(\xi)\,\mathrm{d}\xi$$

- Analytical solutions available for constant breakage kernel: $a(\xi) = a_0$ and symmetric fragmentation $b(\xi'|\xi) = 2\delta\left(\xi - \xi'/2\right)$
- Evolution equation for m_k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = a_0 2^{1-k} m_k - a_0 m_k$$



Method of Moments (Univariate)

Univariate PBE: closure problem for breakage

- In case of pure breakage (particle mass as internal coordinate)

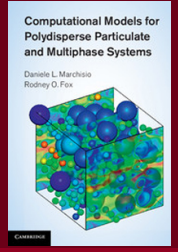
$$\frac{\partial n}{\partial t} = \int_{\xi}^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi' - a(\xi)n(\xi)$$

- Evolution equation for moment of order k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \int_0^{\infty} \xi^k \int_0^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi'\mathrm{d}\xi - \int_0^{\infty} \xi^k a(\xi)n(\xi)\,\mathrm{d}\xi$$

- Analytical solutions available for constant breakage kernel: $a(\xi) = a_0$ and symmetric fragmentation $b(\xi'|\xi) = 2\delta\left(\xi - \xi'/2\right)$
- Evolution equation for m_k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = a_0 2^{1-k} m_k - a_0 m_k$$



Method of Moments (Univariate)

Univariate PBE: closure problem for breakage

- In case of pure breakage (particle mass as internal coordinate)

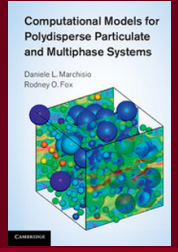
$$\frac{\partial n}{\partial t} = \int_{\xi}^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi' - a(\xi)n(\xi)$$

- Evolution equation for moment of order k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \int_0^{\infty} \xi^k \int_0^{\infty} a(\xi')b(\xi'|\xi)n(\xi')\,\mathrm{d}\xi'\mathrm{d}\xi - \int_0^{\infty} \xi^k a(\xi)n(\xi)\,\mathrm{d}\xi$$

- Analytical solutions available for constant breakage kernel: $a(\xi) = a_0$ and symmetric fragmentation $b(\xi'|\xi) = 2\delta\left(\xi - \xi'/2\right)$
- Evolution equation for m_k is

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = a_0 2^{1-k} m_k - a_0 m_k$$



Method of Moments (Univariate)

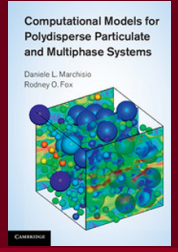
Particle number density model

- If particle mass is internal coordinate and two moments of NDF are tracked: $m_0 = N$ and $m_1 = \rho_\mathrm{p}\alpha_\mathrm{p} = \varrho_\mathrm{p}$
- Mean particle size can be calculated: $d_\mathrm{p} = \left(\frac{m_1}{\rho_\mathrm{p} k_\mathrm{V} m_0}\right)^{1/3}$, where k_V is particle volume shape factor and ρ_p is particle density
- Following reconstruction is done for NDF: $n(t,\xi) = m_0 \delta\left(\xi - \frac{m_1}{m_0}\right)$
- Final equations to be solved for pure aggregation are

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})m_0^2, \qquad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0$$

- With spatial transport (easy implementation in CFD codes)

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot N\mathbf{U}_\mathrm{N} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})N^2, \qquad \frac{\partial \varrho_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{p}\mathbf{U}_\mathrm{M} = 0$$



Method of Moments (Univariate)

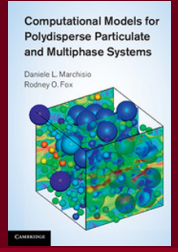
Particle number density model

- If particle mass is internal coordinate and two moments of NDF are tracked: $m_0 = N$ and $m_1 = \rho_\mathrm{p}\alpha_\mathrm{p} = \varrho_\mathrm{p}$
- Mean particle size can be calculated: $d_\mathrm{p} = \left(\frac{m_1}{\rho_\mathrm{p} k_\mathrm{V} m_0}\right)^{1/3}$, where k_V is particle volume shape factor and ρ_p is particle density
- Following reconstruction is done for NDF: $n(t,\xi) = m_0 \delta\left(\xi - \frac{m_1}{m_0}\right)$
- Final equations to be solved for pure aggregation are

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})m_0^2, \qquad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0$$

- With spatial transport (easy implementation in CFD codes)

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot N\mathbf{U}_\mathrm{N} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})N^2, \qquad \frac{\partial \varrho_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{p}\mathbf{U}_\mathrm{M} = 0$$



Method of Moments (Univariate)

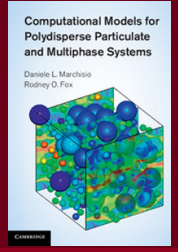
Particle number density model

- If particle mass is internal coordinate and two moments of NDF are tracked: $m_0 = N$ and $m_1 = \rho_\mathrm{p}\alpha_\mathrm{p} = \varrho_\mathrm{p}$
- Mean particle size can be calculated: $d_\mathrm{p} = \left(\frac{m_1}{\rho_\mathrm{p} k_\mathrm{V} m_0}\right)^{1/3}$, where k_V is particle volume shape factor and ρ_p is particle density
- Following reconstruction is done for NDF: $n(t,\xi) = m_0 \delta\left(\xi - \frac{m_1}{m_0}\right)$
- Final equations to be solved for pure aggregation are

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})m_0^2, \qquad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0$$

- With spatial transport (easy implementation in CFD codes)

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot N\mathbf{U}_\mathrm{N} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})N^2, \qquad \frac{\partial \varrho_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{p}\mathbf{U}_\mathrm{M} = 0$$



Method of Moments (Univariate)

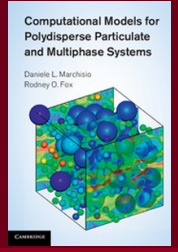
Particle number density model

- If particle mass is internal coordinate and two moments of NDF are tracked: $m_0 = N$ and $m_1 = \rho_\mathrm{p}\alpha_\mathrm{p} = \varrho_\mathrm{p}$
- Mean particle size can be calculated: $d_\mathrm{p} = \left(\frac{m_1}{\rho_\mathrm{p} k_\mathrm{V} m_0}\right)^{1/3}$, where k_V is particle volume shape factor and ρ_p is particle density
- Following reconstruction is done for NDF: $n(t,\xi) = m_0 \delta\left(\xi - \frac{m_1}{m_0}\right)$
- Final equations to be solved for pure aggregation are

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})m_0^2, \qquad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0$$

- With spatial transport (easy implementation in CFD codes)

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot N\mathbf{U}_\mathrm{N} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})N^2, \qquad \frac{\partial \varrho_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{p}\mathbf{U}_\mathrm{M} = 0$$



Method of Moments (Univariate)

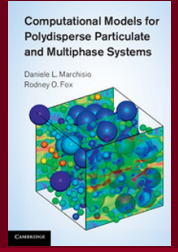
Particle number density model

- If particle mass is internal coordinate and two moments of NDF are tracked: $m_0 = N$ and $m_1 = \rho_\mathrm{p}\alpha_\mathrm{p} = \varrho_\mathrm{p}$
- Mean particle size can be calculated: $d_\mathrm{p} = \left(\frac{m_1}{\rho_\mathrm{p} k_\mathrm{V} m_0}\right)^{1/3}$, where k_V is particle volume shape factor and ρ_p is particle density
- Following reconstruction is done for NDF: $n(t,\xi) = m_0 \delta\left(\xi - \frac{m_1}{m_0}\right)$
- Final equations to be solved for pure aggregation are

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})m_0^2, \qquad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0$$

- With spatial transport (easy implementation in CFD codes)

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot N\mathbf{U}_\mathrm{N} = -\frac{1}{2}\beta(d_\mathrm{p}, d_\mathrm{p})N^2, \qquad \frac{\partial \varrho_\mathrm{p}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_\mathrm{p}\mathbf{U}_\mathrm{M} = 0$$



Method of Moments (Univariate)

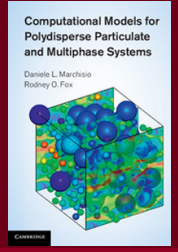
Particle number density model

- If particle mass is internal coordinate and two moments of NDF are tracked: $m_0 = N$ and $m_1 = \rho_p \alpha_p = \varrho_p$
- Mean particle size can be calculated: $d_p = \left(\frac{m_1}{\rho_p k_v m_0} \right)^{1/3}$, where k_v is particle volume shape factor and ρ_p is particle density
- Following reconstruction is done for NDF: $n(t, \xi) = m_0 \delta \left(\xi - \frac{m_1}{m_0} \right)$
- Final equations to be solved for pure aggregation are

$$\frac{\mathrm{d}m_0}{\mathrm{d}t} = -\frac{1}{2}\beta(d_p, d_p)m_0^2, \qquad \frac{\mathrm{d}m_1}{\mathrm{d}t} = 0$$

- With spatial transport (easy implementation in CFD codes)

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot N\mathbf{U}_N = -\frac{1}{2}\beta(d_p, d_p)N^2, \qquad \frac{\partial \varrho_p}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \varrho_p \mathbf{U}_M = 0$$



Quadrature-Based Moment Methods

- Our original (univariate) problem was

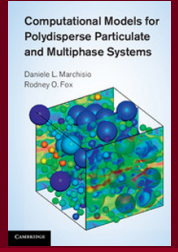
$$\frac{\partial n}{\partial t} = S = -\frac{\partial}{\partial \xi}\left(\dot{\xi} n\right) + h$$

- Instead we solve transport equations for moments

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k$$

with $k = 0, 1, 2, \ldots, 2N-1$ and with specific initial condition $m_k(0) = \int_{\Omega_\xi} n(0,\xi)\xi^k \,\mathrm{d}\xi$

- Integration of ODE requires the evaluation of source term through quadrature approximation (Wheeler algorithm)



Quadrature-Based Moment Methods

- Our original (univariate) problem was

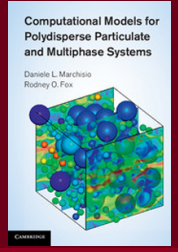
$$\frac{\partial n}{\partial t} = S = -\frac{\partial}{\partial \xi}\left(\dot{\xi} n\right) + h$$

- Instead we solve transport equations for moments

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k$$

with $k = 0, 1, 2, \ldots, 2N-1$ and with specific initial condition $m_k(0) = \int_{\Omega_\xi} n(0,\xi)\xi^k \,\mathrm{d}\xi$

- Integration of ODE requires the evaluation of source term through quadrature approximation (Wheeler algorithm)



Quadrature-Based Moment Methods

- Our original (univariate) problem was

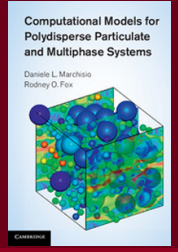
$$\frac{\partial n}{\partial t} = S = -\frac{\partial}{\partial \xi}\left(\dot{\xi} n\right) + h$$

- Instead we solve transport equations for moments

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k$$

with $k = 0, 1, 2, \ldots, 2N-1$ and with specific initial condition $m_k(0) = \int_{\Omega_\xi} n(0,\xi)\xi^k \,\mathrm{d}\xi$

- Integration of ODE requires the evaluation of source term through quadrature approximation (Wheeler algorithm)



Quadrature-Based Moment Methods

- Our original (univariate) problem was

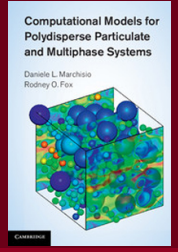
$$\frac{\partial n}{\partial t} = S = -\frac{\partial}{\partial \xi}\left(\dot{\xi} n\right) + h$$

- Instead we solve transport equations for moments

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k$$

with $k = 0, 1, 2, \ldots, 2N-1$ and with specific initial condition $m_k(0) = \int_{\Omega_\xi} n(0,\xi)\xi^k \,\mathrm{d}\xi$

- Integration of ODE requires the evaluation of source term through quadrature approximation (Wheeler algorithm)



Quadrature Method of Moments

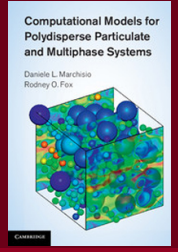
- For example, in case of continuous rate of change of internal coordinate

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \int_0^\infty \dot{\xi}(\xi)\xi^{k-1} n(\xi)\,\mathrm{d}\xi$$

- Closure problem is overcome as

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \sum_{\alpha=1}^{N} \dot{\xi}(\xi_\alpha)(\xi_\alpha)^{k-1} w_\alpha$$

- where weights w_α and nodes ξ_α are calculated from PD or Wheeler algorithm from moments
- This method is called Quadrature Method of Moments



Quadrature Method of Moments

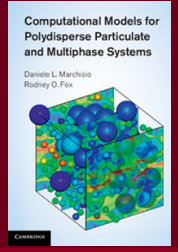
- For example, in case of continuous rate of change of internal coordinate

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \int_0^\infty \dot{\xi}(\xi)\xi^{k-1} n(\xi)\,\mathrm{d}\xi$$

- Closure problem is overcome as

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \sum_{\alpha=1}^{N} \dot{\xi}(\xi_\alpha)(\xi_\alpha)^{k-1} w_\alpha$$

- where weights w_α and nodes ξ_α are calculated from PD or Wheeler algorithm from moments
- This method is called Quadrature Method of Moments



Quadrature Method of Moments

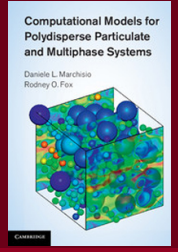
- For example, in case of continuous rate of change of internal coordinate

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \int_0^\infty \dot{\xi}(\xi)\xi^{k-1} n(\xi)\,\mathrm{d}\xi$$

- Closure problem is overcome as

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \sum_{\alpha=1}^{N} \dot{\xi}(\xi_\alpha)(\xi_\alpha)^{k-1} w_\alpha$$

- where weights w_α and nodes ξ_α are calculated from PD or Wheeler algorithm from moments
- This method is called Quadrature Method of Moments



Quadrature Method of Moments

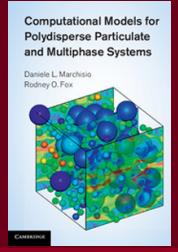
- For example, in case of continuous rate of change of internal coordinate

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \int_0^\infty \dot{\xi}(\xi)\xi^{k-1} n(\xi)\,\mathrm{d}\xi$$

- Closure problem is overcome as

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \sum_{\alpha=1}^{N} \dot{\xi}(\xi_\alpha)(\xi_\alpha)^{k-1} w_\alpha$$

- where weights w_α and nodes ξ_α are calculated from PD or Wheeler algorithm from moments
- This method is called Quadrature Method of Moments



Quadrature Method of Moments

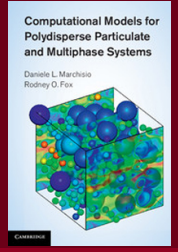
- For example, in case of continuous rate of change of internal coordinate

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \int_0^\infty \dot{\xi}(\xi)\xi^{k-1} n(\xi)\,\mathrm{d}\xi$$

- Closure problem is overcome as

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = k \sum_{\alpha=1}^{N} \dot{\xi}(\xi_\alpha)(\xi_\alpha)^{k-1} w_\alpha$$

- where weights w_α and nodes ξ_α are calculated from PD or Wheeler algorithm from moments
- This method is called Quadrature Method of Moments



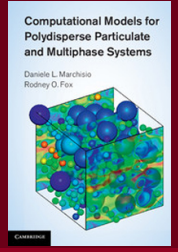
Quadrature Method of Moments

In case of standard nucleation, (positive) growth, growth dispersion, aggregation and breakage, application of QBMM to source term yields

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \overline{S}_k = \overline{J}_k + k\sum_{\alpha=1}^{N} \xi_\alpha^{k-1}\dot{\xi}_\alpha w_\alpha + k(k-1)\sum_{\alpha=1}^{N} \xi_\alpha^{k-2} D_\alpha w_\alpha$$

$$+ \frac{1}{2}\sum_{\alpha=1}^{N}\sum_{\gamma=1}^{N}\left[(\xi_\alpha+\xi_\gamma)^k - \xi_\alpha^k - \xi_\gamma^k\right]\beta_{\alpha,\gamma} w_\alpha w_\gamma + \sum_{\alpha=1}^{N} a_\alpha \bar{b}_\alpha^k w_\alpha - \sum_{\alpha=1}^{N} \xi_\alpha^k a_\alpha w_\alpha$$

where $\dot{\xi}_\alpha = \dot{\xi}(\xi_\alpha)$, $D_\alpha = D(\xi_\alpha)$, $\beta_{\alpha,\gamma} = \beta(\xi_\alpha, \xi_\gamma)$ and $a_\alpha = a(\xi_\alpha)$ and moments of daughter distribution function are $\bar{b}_\alpha^k = \int \xi^k b(\xi|\xi_\alpha)\,\mathrm{d}\xi$



Direct Quadrature Method of Moments

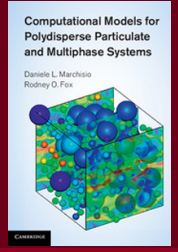
- Fact that closure problem is overcome with quadrature approximation

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

- is equivalent to assumption that NDF is

$$n(\xi) = \sum_{\alpha=1}^{N} w_\alpha \delta\left(\xi - \xi_\alpha\right)$$

- Instead of tracking evolution for moments, evolution of weights and nodes in quadrature approximation could be directly tracked: Direct quadrature method of moments



Direct Quadrature Method of Moments

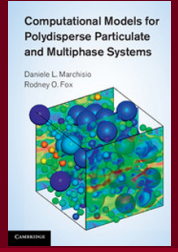
- Fact that closure problem is overcome with quadrature approximation

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

- is equivalent to assumption that NDF is

$$n(\xi) = \sum_{\alpha=1}^{N} w_\alpha \delta\left(\xi - \xi_\alpha\right)$$

- Instead of tracking evolution for moments, evolution of weights and nodes in quadrature approximation could be directly tracked: Direct quadrature method of moments



Direct Quadrature Method of Moments

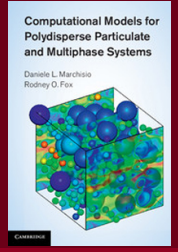
- Fact that closure problem is overcome with quadrature approximation

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

- is equivalent to assumption that NDF is

$$n(\xi) = \sum_{\alpha=1}^{N} w_\alpha \delta\left(\xi - \xi_\alpha\right)$$

- Instead of tracking evolution for moments, evolution of weights and nodes in quadrature approximation could be directly tracked: Direct quadrature method of moments



Direct Quadrature Method of Moments

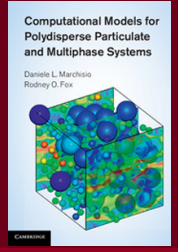
- Fact that closure problem is overcome with quadrature approximation

$$\int_{\Omega_\xi} n(\xi) g(\xi)\, \mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha g(\xi_\alpha)$$

- is equivalent to assumption that NDF is

$$n(\xi) = \sum_{\alpha=1}^{N} w_\alpha \delta\left(\xi - \xi_\alpha\right)$$

- Instead of tracking evolution for moments, evolution of weights and nodes in quadrature approximation could be directly tracked: Direct quadrature method of moments



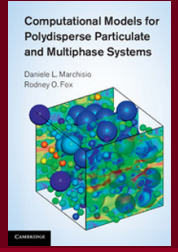
Direct Quadrature Method of Moments

- If weights and nodes are continuous functions we obtain

$$\sum_{\alpha=1}^{N} \delta(\xi - \xi_\alpha)\left(\frac{\mathrm{d}w_\alpha}{\mathrm{d}t}\right) - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha)\left(w_\alpha \frac{\mathrm{d}\xi_\alpha}{\mathrm{d}t}\right) = S(\xi)$$

- If weighted nodes (or weighted abscissas) $\varsigma_\alpha = w_\alpha \xi_\alpha$ are introduced

$$\sum_{\alpha=1}^{N} \delta(\xi - \xi_\alpha)\left(\frac{\mathrm{d}w_\alpha}{\mathrm{d}t}\right) - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha)\left(-\xi_\alpha \frac{\mathrm{d}w_\alpha}{\mathrm{d}t} + \frac{\mathrm{d}\varsigma_\alpha}{\mathrm{d}t}\right) = S(\xi)$$



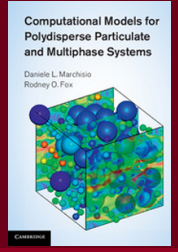
Direct Quadrature Method of Moments

- If weights and nodes are continuous functions we obtain

$$\sum_{\alpha=1}^{N} \delta(\xi - \xi_\alpha)\left(\frac{\mathrm{d}w_\alpha}{\mathrm{d}t}\right) - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha)\left(w_\alpha \frac{\mathrm{d}\xi_\alpha}{\mathrm{d}t}\right) = S(\xi)$$

- If weighted nodes (or weighted abscissas) $\varsigma_\alpha = w_\alpha \xi_\alpha$ are introduced

$$\sum_{\alpha=1}^{N} \delta(\xi - \xi_\alpha)\left(\frac{\mathrm{d}w_\alpha}{\mathrm{d}t}\right) - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha)\left(-\xi_\alpha \frac{\mathrm{d}w_\alpha}{\mathrm{d}t} + \frac{\mathrm{d}\varsigma_\alpha}{\mathrm{d}t}\right) = S(\xi)$$



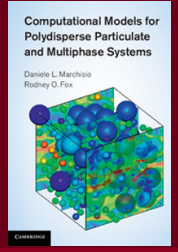
Direct Quadrature Method of Moments

- If weights and nodes are continuous functions we obtain

$$\sum_{\alpha=1}^{N} \delta(\xi - \xi_\alpha)\left(\frac{\mathrm{d}w_\alpha}{\mathrm{d}t}\right) - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha)\left(w_\alpha \frac{\mathrm{d}\xi_\alpha}{\mathrm{d}t}\right) = S(\xi)$$

- If weighted nodes (or weighted abscissas) $\varsigma_\alpha = w_\alpha \xi_\alpha$ are introduced

$$\sum_{\alpha=1}^{N} \delta(\xi - \xi_\alpha)\left(\frac{\mathrm{d}w_\alpha}{\mathrm{d}t}\right) - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha)\left(-\xi_\alpha \frac{\mathrm{d}w_\alpha}{\mathrm{d}t} + \frac{\mathrm{d}\varsigma_\alpha}{\mathrm{d}t}\right) = S(\xi)$$



Direct Quadrature Method of Moments

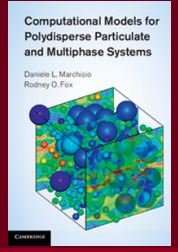
- We now define a_α and b_α to be source terms

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}\varsigma_\alpha}{\mathrm{d}t} = b_\alpha$$

- Using these definitions we write a simpler form

$$\sum_{\alpha=1}^{N} \left[\delta(\xi-\xi_\alpha) + \delta'(\xi-\xi_\alpha)\xi_\alpha\right] a_\alpha - \sum_{\alpha=1}^{N} \delta'(\xi-\xi_\alpha) b_\alpha = S(\xi)$$

- This equation can now be used to determine the unknown functions a_α and b_α by applying the moment transformation



Direct Quadrature Method of Moments

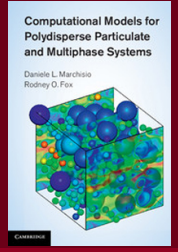
- We now define a_α and b_α to be source terms

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}\varsigma_\alpha}{\mathrm{d}t} = b_\alpha$$

- Using these definitions we write a simpler form

$$\sum_{\alpha=1}^{N} \left[\delta(\xi-\xi_\alpha) + \delta'(\xi-\xi_\alpha)\xi_\alpha\right] a_\alpha - \sum_{\alpha=1}^{N} \delta'(\xi-\xi_\alpha) b_\alpha = S(\xi)$$

- This equation can now be used to determine the unknown functions a_α and b_α by applying the moment transformation



Direct Quadrature Method of Moments

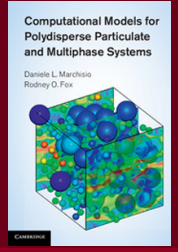
- We now define a_α and b_α to be source terms

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}\varsigma_\alpha}{\mathrm{d}t} = b_\alpha$$

- Using these definitions we write a simpler form

$$\sum_{\alpha=1}^{N} \left[\delta(\xi - \xi_\alpha) + \delta'(\xi - \xi_\alpha)\xi_\alpha\right] a_\alpha - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha) b_\alpha = S(\xi)$$

- This equation can now be used to determine the unknown functions a_α and b_α by applying the moment transformation



Direct Quadrature Method of Moments

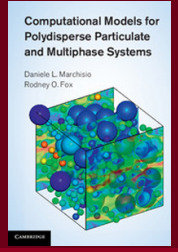
- We now define a_α and b_α to be source terms

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}\varsigma_\alpha}{\mathrm{d}t} = b_\alpha$$

- Using these definitions we write a simpler form

$$\sum_{\alpha=1}^{N} \left[\delta(\xi - \xi_\alpha) + \delta'(\xi - \xi_\alpha)\xi_\alpha\right] a_\alpha - \sum_{\alpha=1}^{N} \delta'(\xi - \xi_\alpha) b_\alpha = S(\xi)$$

- This equation can now be used to determine the unknown functions a_α and b_α by applying the moment transformation



Direct Quadrature Method of Moments

- DQMOM can be applied for any independent set of moments (number of moments MUST be equal the number of unknown functions)
- Knowing that

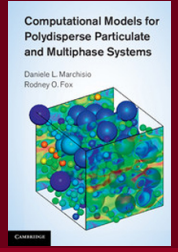
$$\int_{-\infty}^{+\infty} \xi^k \delta(\xi - \xi_\alpha)\, \mathrm{d}\xi = (\xi_\alpha)^k$$

$$\int_{-\infty}^{+\infty} \xi^k \delta'(\xi - \xi_\alpha)\, \mathrm{d}\xi = -k(\xi_\alpha)^{k-1}$$

- Moment transform yields

$$(1-k)\sum_{\alpha=1}^{N} \xi_\alpha^k a_\alpha + k\sum_{\alpha=1}^{N} \xi_\alpha^{k-1} b_\alpha = \overline{S}_k$$

with $k = k_1, k_2, \ldots, k_{2N}$



Direct Quadrature Method of Moments

- DQMOM can be applied for any independent set of moments (number of moments MUST be equal the number of unknown functions)
- Knowing that

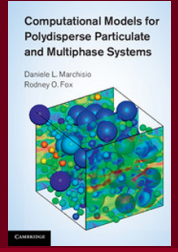
$$\int_{-\infty}^{+\infty} \xi^k \delta(\xi - \xi_\alpha)\, \mathrm{d}\xi = (\xi_\alpha)^k$$

$$\int_{-\infty}^{+\infty} \xi^k \delta'(\xi - \xi_\alpha)\, \mathrm{d}\xi = -k(\xi_\alpha)^{k-1}$$

- Moment transform yields

$$(1-k)\sum_{\alpha=1}^{N} \xi_\alpha^k a_\alpha + k\sum_{\alpha=1}^{N} \xi_\alpha^{k-1} b_\alpha = \overline{S}_k$$

with $k = k_1, k_2, \ldots, k_{2N}$



Direct Quadrature Method of Moments

- DQMOM can be applied for any independent set of moments (number of moments MUST be equal the number of unknown functions)
- Knowing that

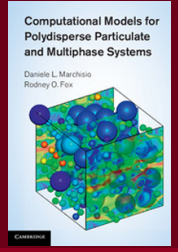
$$\int_{-\infty}^{+\infty} \xi^k \delta(\xi - \xi_\alpha)\, \mathrm{d}\xi = (\xi_\alpha)^k$$

$$\int_{-\infty}^{+\infty} \xi^k \delta'(\xi - \xi_\alpha)\, \mathrm{d}\xi = -k(\xi_\alpha)^{k-1}$$

- Moment transform yields

$$(1-k)\sum_{\alpha=1}^{N} \xi_\alpha^k a_\alpha + k\sum_{\alpha=1}^{N} \xi_\alpha^{k-1} b_\alpha = \overline{S}_k$$

with $k = k_1, k_2, \ldots, k_{2N}$



Direct Quadrature Method of Moments

- DQMOM can be applied for any independent set of moments (number of moments MUST be equal the number of unknown functions)
- Knowing that

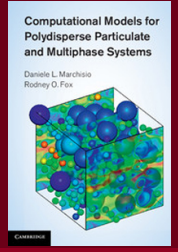
$$\int_{-\infty}^{+\infty} \xi^k \delta(\xi - \xi_\alpha)\, \mathrm{d}\xi = (\xi_\alpha)^k$$

$$\int_{-\infty}^{+\infty} \xi^k \delta'(\xi - \xi_\alpha)\, \mathrm{d}\xi = -k(\xi_\alpha)^{k-1}$$

- Moment transform yields

$$(1-k)\sum_{\alpha=1}^{N} \xi_\alpha^k a_\alpha + k\sum_{\alpha=1}^{N} \xi_\alpha^{k-1} b_\alpha = \overline{S}_k$$

with $k = k_1, k_2, \ldots, k_{2N}$



Direct Quadrature Method of Moments

- Linear system can be written in matrix form:

$$\mathbf{A}\alpha = \mathbf{d}$$

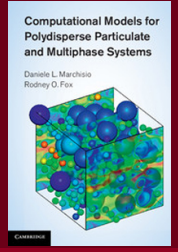
where

$$\alpha = \begin{bmatrix} a_1 & \cdots & a_N & b_1 & \cdots & b_N \end{bmatrix}^T = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} \overline{S}_{k_1} & \cdots & \overline{S}_{k_{2N-1}} \end{bmatrix}^T$$

- Components of matrix $\mathbf{A}$ are

$$a_{ij} = \begin{cases} (1-k_i)\,\xi_j^{k_i} & \text{if } 1 \le j \le N \\ k_i \xi_j^{k_i-1} & \text{if } N+1 \le j \le 2N \end{cases}$$



Direct Quadrature Method of Moments

- Linear system can be written in matrix form:

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{d}$$

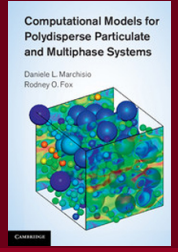
where

$$\boldsymbol{\alpha} = \begin{bmatrix} a_1 & \cdots & a_N & b_1 & \cdots & b_N \end{bmatrix}^T = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} \overline{S}_{k_1} & \cdots & \overline{S}_{k_{2N-1}} \end{bmatrix}^T$$

- Components of matrix $\mathbf{A}$ are

$$a_{ij} = \begin{cases} (1-k_i)\,\xi_j^{k_i} & \text{if } 1 \le j \le N \\ k_i \xi_j^{k_i-1} & \text{if } N+1 \le j \le 2N \end{cases}$$



Direct Quadrature Method of Moments

- Linear system can be written in matrix form:

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{d}$$

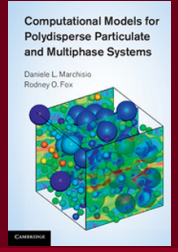
where

$$\boldsymbol{\alpha} = \begin{bmatrix} a_1 & \cdots & a_N & b_1 & \cdots & b_N \end{bmatrix}^T = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} \overline{S}_{k_1} & \cdots & \overline{S}_{k_{2N-1}} \end{bmatrix}^T$$

- Components of matrix $\mathbf{A}$ are

$$a_{ij} = \begin{cases} (1-k_i)\,\xi_j^{k_i} & \text{if } 1 \le j \le N \\ k_i \xi_j^{k_i - 1} & \text{if } N+1 \le j \le 2N \end{cases}$$

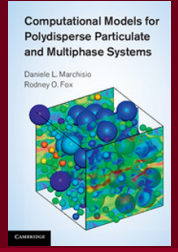


Direct Quadrature Method of Moments

- If (as in QMOM) first $2N$ integer moments are chosen (i.e., $k = 0, \ldots, 2N-1$), matrix of linear system is

$$\mathbf{A} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 \\ -\xi_1^2 & \cdots & -\xi_N^2 & 2\xi_1 & \cdots & 2\xi_N \\ \vdots & & \vdots & \vdots & & \vdots \\ 2(1-N)\xi_1^{2N-1} & \cdots & 2(1-N)\xi_N^{2N-1} & (2N-1)\xi_1^{2N-2} & \cdots & (2N-1)\xi_N^{2N-2} \end{bmatrix}$$

- $\mathbf{A}$ does not depend on weights w_α and if abscissas ξ_α are unique, then $\mathbf{A}$ will be full rank

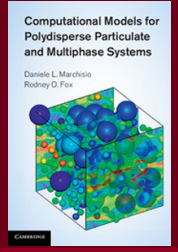


Direct Quadrature Method of Moments

- If (as in QMOM) first $2N$ integer moments are chosen (i.e., $k = 0, \ldots, 2N-1$), matrix of linear system is

$$\mathbf{A} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 \\ -\xi_1^2 & \cdots & -\xi_N^2 & 2\xi_1 & \cdots & 2\xi_N \\ \vdots & & \vdots & \vdots & & \vdots \\ 2(1-N)\xi_1^{2N-1} & \cdots & 2(1-N)\xi_N^{2N-1} & (2N-1)\xi_1^{2N-2} & \cdots & (2N-1)\xi_N^{2N-2} \end{bmatrix}$$

- $\mathbf{A}$ does not depend on weights w_α and if abscissas ξ_α are unique, then $\mathbf{A}$ will be full rank

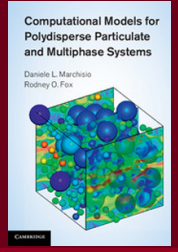


Direct Quadrature Method of Moments

- If (as in QMOM) first $2N$ integer moments are chosen (i.e., $k = 0, \ldots, 2N-1$), matrix of linear system is

$$\mathbf{A} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 \\ -\xi_1^2 & \cdots & -\xi_N^2 & 2\xi_1 & \cdots & 2\xi_N \\ \vdots & & \vdots & \vdots & & \vdots \\ 2(1-N)\xi_1^{2N-1} & \cdots & 2(1-N)\xi_N^{2N-1} & (2N-1)\xi_1^{2N-2} & \cdots & (2N-1)\xi_N^{2N-2} \end{bmatrix}$$

- $\mathbf{A}$ does not depend on weights w_α and if abscissas ξ_α are unique, then $\mathbf{A}$ will be full rank



Direct Quadrature Method of Moments

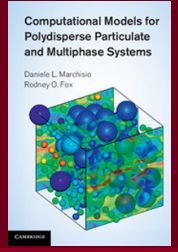
- Method is called Direct quadrature method of moments
- Evolution equations for weights and nodes of quadrature approximation are solved:

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}w_\alpha \xi_\alpha}{\mathrm{d}t} = b_\alpha$$

- Source terms are calculated by inverting linear system and by using following initial condition:

$$w_\alpha(0) = w_\alpha^0, \qquad \varsigma_\alpha(0) = w_\alpha^0 \xi_\alpha^0 \quad \text{for } k = 1, \ldots, N$$

in turn calculated from initial moments



Direct Quadrature Method of Moments

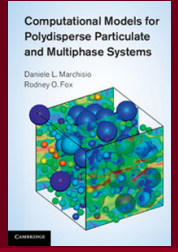
- Method is called Direct quadrature method of moments
- Evolution equations for weights and nodes of quadrature approximation are solved:

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}w_\alpha\xi_\alpha}{\mathrm{d}t} = b_\alpha$$

- Source terms are calculated by inverting linear system and by using following initial condition:

$$w_\alpha(0) = w_\alpha^0, \qquad \varsigma_\alpha(0) = w_\alpha^0\xi_\alpha^0 \quad \text{for } k = 1, \ldots, N$$

in turn calculated from initial moments



Direct Quadrature Method of Moments

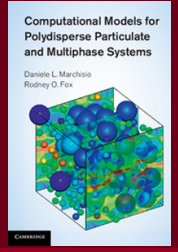
- Method is called Direct quadrature method of moments
- Evolution equations for weights and nodes of quadrature approximation are solved:

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}w_\alpha\xi_\alpha}{\mathrm{d}t} = b_\alpha$$

- Source terms are calculated by inverting linear system and by using following initial condition:

$$w_\alpha(0) = w_\alpha^0, \qquad \varsigma_\alpha(0) = w_\alpha^0\xi_\alpha^0 \quad \text{for } k = 1, \ldots, N$$

in turn calculated from initial moments



Direct Quadrature Method of Moments

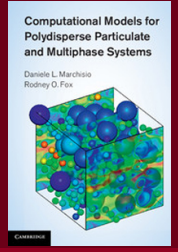
- Method is called Direct quadrature method of moments
- Evolution equations for weights and nodes of quadrature approximation are solved:

$$\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} = a_\alpha, \qquad \frac{\mathrm{d}w_\alpha\xi_\alpha}{\mathrm{d}t} = b_\alpha$$

- Source terms are calculated by inverting linear system and by using following initial condition:

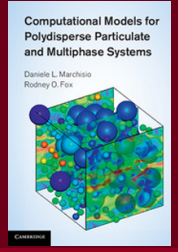
$$w_\alpha(0) = w_\alpha^0, \qquad \varsigma_\alpha(0) = w_\alpha^0\xi_\alpha^0 \quad \text{for } k = 1, \ldots, N$$

in turn calculated from initial moments



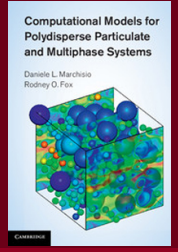
Quadrature-Based Moment Methods

- QBMM are very accurate in tracking evolution of moments of NDF: 4–8 moments do same job of many (e.g. 100) classes (see for example work of Marchisio et al. (2003) and Vanni (2000))
- PD algorithm is very stable (if the moments are realizable) and for particular cases Wheeler algorithm is successful when PD fails
- Important differences in QBMM arise when treating spatially inhomogeneous systems (discussed next)
- In general increasing number of nodes of quadrature approximation and of moments to be tracked increases accuracy, unless problem is localized in phase space (e.g. fines dissolution)



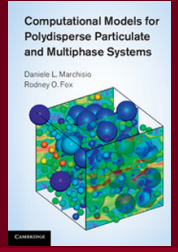
Quadrature-Based Moment Methods

- QBMM are very accurate in tracking evolution of moments of NDF: 4–8 moments do same job of many (e.g. 100) classes (see for example work of Marchisio et al. (2003) and Vanni (2000))
- PD algorithm is very stable (if the moments are realizable) and for particular cases Wheeler algorithm is successful when PD fails
- Important differences in QBMM arise when treating spatially inhomogeneous systems (discussed next)
- In general increasing number of nodes of quadrature approximation and of moments to be tracked increases accuracy, unless problem is localized in phase space (e.g. fines dissolution)



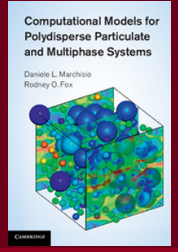
Quadrature-Based Moment Methods

- QBMM are very accurate in tracking evolution of moments of NDF: 4–8 moments do same job of many (e.g. 100) classes (see for example work of Marchisio et al. (2003) and Vanni (2000))
- PD algorithm is very stable (if the moments are realizable) and for particular cases Wheeler algorithm is successful when PD fails
- Important differences in QBMM arise when treating spatially inhomogeneous systems (discussed next)
- In general increasing number of nodes of quadrature approximation and of moments to be tracked increases accuracy, unless problem is localized in phase space (e.g. fines dissolution)



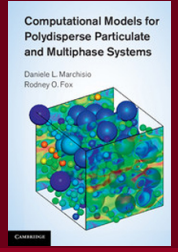
Quadrature-Based Moment Methods

- QBMM are very accurate in tracking evolution of moments of NDF: 4–8 moments do same job of many (e.g. 100) classes (see for example work of Marchisio et al. (2003) and Vanni (2000))
- PD algorithm is very stable (if the moments are realizable) and for particular cases Wheeler algorithm is successful when PD fails
- Important differences in QBMM arise when treating spatially inhomogeneous systems (discussed next)
- In general increasing number of nodes of quadrature approximation and of moments to be tracked increases accuracy, unless problem is localized in phase space (e.g. fines dissolution)



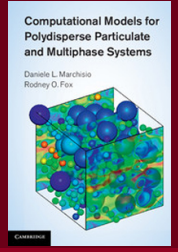
Quadrature-Based Moment Methods

- QBMM are very accurate in tracking evolution of moments of NDF: 4–8 moments do same job of many (e.g. 100) classes (see for example work of Marchisio et al. (2003) and Vanni (2000))
- PD algorithm is very stable (if the moments are realizable) and for particular cases Wheeler algorithm is successful when PD fails
- Important differences in QBMM arise when treating spatially inhomogeneous systems (discussed next)
- In general increasing number of nodes of quadrature approximation and of moments to be tracked increases accuracy, unless problem is localized in phase space (e.g. fines dissolution)



Part 4

QBMM FOR SPATIALLY INHOMOGENEOUS SYSTEMS



Spatially Inhomogeneous PBE

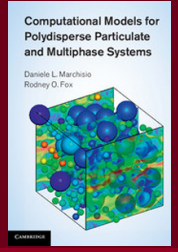
- Spatially inhomogeneous PBE operating on $n(\xi)$ is:

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_p | \xi \rangle n = \frac{\partial}{\partial \mathbf{x}} \cdot D(\xi) \frac{\partial n}{\partial \mathbf{x}} - \frac{\partial}{\partial \xi} \dot{\xi}(\xi) n + h$$

  where $\langle \mathbf{U}_p | \xi \rangle$ and $D(\xi)$ are known functions of ξ
- Application of moment transform, $M_k = \int n(\xi) \xi^k \, \mathrm{d}\xi$, generates several closure problems:

$$\frac{\partial M_k}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_k M_k = \frac{\partial}{\partial \mathbf{x}} \cdot D_k \frac{\partial M_k}{\partial \mathbf{x}} + \overline{S}_k$$

  where $\mathbf{U}_k = \frac{\int \langle \mathbf{U}_p | \xi \rangle n(\xi) \xi^k \, \mathrm{d}\xi}{M_k}$ (similarly D_k) and $\overline{h}_k = \int h(\xi) \xi^k \, \mathrm{d}\xi$
- Solution with QMOM is as usual ...



Spatially Inhomogeneous PBE

- Spatially inhomogeneous PBE operating on $n(\xi)$ is:

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_p | \xi \rangle n = \frac{\partial}{\partial \mathbf{x}} \cdot D(\xi) \frac{\partial n}{\partial \mathbf{x}} - \frac{\partial}{\partial \xi} \dot{\xi}(\xi) n + h$$

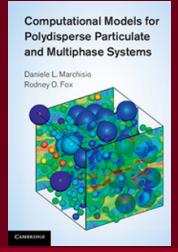
  where $\langle \mathbf{U}_p | \xi \rangle$ and $D(\xi)$ are known functions of ξ

- Application of moment transform, $M_k = \int n(\xi) \xi^k \, \mathrm{d}\xi$, generates several closure problems:

$$\frac{\partial M_k}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_k M_k = \frac{\partial}{\partial \mathbf{x}} \cdot D_k \frac{\partial M_k}{\partial \mathbf{x}} + \overline{S}_k$$

  where $\mathbf{U}_k = \frac{\int \langle \mathbf{U}_p | \xi \rangle n(\xi) \xi^k \, \mathrm{d}\xi}{M_k}$ (similarly D_k) and $\overline{h}_k = \int h(\xi) \xi^k \, \mathrm{d}\xi$

- Solution with QMOM is as usual ...



Spatially Inhomogeneous PBE

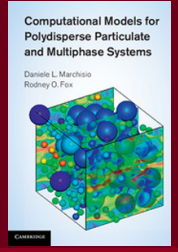
- Spatially inhomogeneous PBE operating on $n(\xi)$ is:

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_p | \xi \rangle n = \frac{\partial}{\partial \mathbf{x}} \cdot D(\xi) \frac{\partial n}{\partial \mathbf{x}} - \frac{\partial}{\partial \xi} \dot{\xi}(\xi) n + h$$

  where $\langle \mathbf{U}_p | \xi \rangle$ and $D(\xi)$ are known functions of ξ
- Application of moment transform, $M_k = \int n(\xi) \xi^k \, \mathrm{d}\xi$, generates several closure problems:

$$\frac{\partial M_k}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_k M_k = \frac{\partial}{\partial \mathbf{x}} \cdot D_k \frac{\partial M_k}{\partial \mathbf{x}} + \overline{S}_k$$

  where $\mathbf{U}_k = \frac{\int \langle \mathbf{U}_p | \xi \rangle n(\xi) \xi^k \, \mathrm{d}\xi}{M_k}$ (similarly D_k) and $\overline{h}_k = \int h(\xi) \xi^k \, \mathrm{d}\xi$
- Solution with QMOM is as usual ...



Spatially Inhomogeneous PBE

- Spatially inhomogeneous PBE operating on $n(\xi)$ is:

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \langle \mathbf{U}_p | \xi \rangle n = \frac{\partial}{\partial \mathbf{x}} \cdot D(\xi) \frac{\partial n}{\partial \mathbf{x}} - \frac{\partial}{\partial \xi} \dot{\xi}(\xi) n + h$$

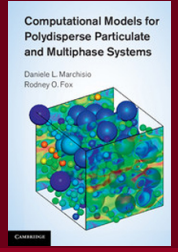
  where $\langle \mathbf{U}_p | \xi \rangle$ and $D(\xi)$ are known functions of ξ

- Application of moment transform, $M_k = \int n(\xi) \xi^k \, \mathrm{d}\xi$, generates several closure problems:

$$\frac{\partial M_k}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_k M_k = \frac{\partial}{\partial \mathbf{x}} \cdot D_k \frac{\partial M_k}{\partial \mathbf{x}} + \overline{S}_k$$

  where $\mathbf{U}_k = \frac{\int \langle \mathbf{U}_p | \xi \rangle n(\xi) \xi^k \, \mathrm{d}\xi}{M_k}$ (similarly D_k) and $\overline{h}_k = \int h(\xi) \xi^k \, \mathrm{d}\xi$

- Solution with QMOM is as usual ...



Spatially Inhomogeneous PBE

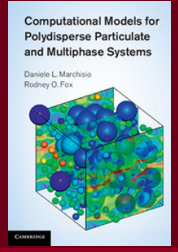
- From $\{M_0, M_1, \ldots, M_{2N-1}\}$ quadrature formula of order N is constructed resulting in following approximation

$$M_k(t,\mathbf{x}) := \int_{\Omega_\xi} n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha(t,\mathbf{x})\xi_\alpha(t,\mathbf{x})^k$$

- that can be used to overcome different closure problems, for example:

$$\mathbf{U}_k(t,\mathbf{x}) := \frac{\int_{\Omega_\xi} \langle \mathbf{U}_p|\xi\rangle n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi}{M_k} \approx \frac{\sum_{\alpha=1}^{N}\langle \mathbf{U}_p|\xi_\alpha\rangle w_\alpha \xi_\alpha^k}{M_k}$$

- By using different velocities, $\mathbf{U}_k(t,\mathbf{x})$, for different moments, we are capable of describing mixing and segregation patterns



Spatially Inhomogeneous PBE

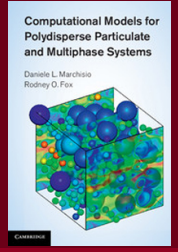
- From $\{M_0, M_1, \ldots, M_{2N-1}\}$ quadrature formula of order N is constructed resulting in following approximation

$$M_k(t,\mathbf{x}) := \int_{\Omega_\xi} n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha(t,\mathbf{x})\xi_\alpha(t,\mathbf{x})^k$$

- that can be used to overcome different closure problems, for example:

$$\mathbf{U}_k(t,\mathbf{x}) := \frac{\int_{\Omega_\xi} \langle \mathbf{U}_p|\xi\rangle n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi}{M_k} \approx \frac{\sum_{\alpha=1}^{N} \langle \mathbf{U}_p|\xi_\alpha\rangle w_\alpha \xi_\alpha^k}{M_k}$$

- By using different velocities, $\mathbf{U}_k(t,\mathbf{x})$, for different moments, we are capable of describing mixing and segregation patterns



Spatially Inhomogeneous PBE

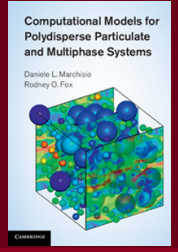
- From $\{M_0, M_1, \ldots, M_{2N-1}\}$ quadrature formula of order N is constructed resulting in following approximation

$$M_k(t,\mathbf{x}) := \int_{\Omega_\xi} n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha(t,\mathbf{x})\xi_\alpha(t,\mathbf{x})^k$$

- that can be used to overcome different closure problems, for example:

$$\mathbf{U}_k(t,\mathbf{x}) := \frac{\int_{\Omega_\xi} \langle \mathbf{U}_p|\xi\rangle n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi}{M_k} \approx \frac{\sum_{\alpha=1}^{N}\langle \mathbf{U}_p|\xi_\alpha\rangle w_\alpha \xi_\alpha^k}{M_k}$$

- By using different velocities, $\mathbf{U}_k(t,\mathbf{x})$, for different moments, we are capable of describing mixing and segregation patterns



Spatially Inhomogeneous PBE

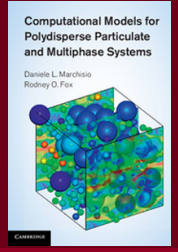
- From $\{M_0, M_1, \ldots, M_{2N-1}\}$ quadrature formula of order N is constructed resulting in following approximation

$$M_k(t,\mathbf{x}) := \int_{\Omega_\xi} n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi \approx \sum_{\alpha=1}^{N} w_\alpha(t,\mathbf{x})\xi_\alpha(t,\mathbf{x})^k$$

- that can be used to overcome different closure problems, for example:

$$\mathbf{U}_k(t,\mathbf{x}) := \frac{\int_{\Omega_\xi} \langle \mathbf{U}_p|\xi\rangle n(t,\mathbf{x},\xi)\xi^k \,\mathrm{d}\xi}{M_k} \approx \frac{\sum_{\alpha=1}^{N} \langle \mathbf{U}_p|\xi_\alpha\rangle w_\alpha \xi_\alpha^k}{M_k}$$

- By using different velocities, $\mathbf{U}_k(t,\mathbf{x})$, for different moments, we are capable of describing mixing and segregation patterns



Spatial Discretization in QMOM (1-D)

Solve transport equation for generic moment m_k with discretization schemes that PRESERVE realizable moments (constant $\mathrm{U_p}$):

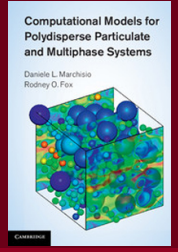
$$\frac{\partial m_k}{\partial t} + \mathrm{U_p}\frac{\partial m_k}{\partial x} = \overline{S}_k$$

After spatial discretization (finite-volume method):

$$\frac{\mathrm{d}m_k^{\mathrm{P}}}{\mathrm{d}t} = \overline{S}_k^{\mathrm{P}} - \frac{\mathrm{U_p}}{\Delta x}\left(m_k^{\mathrm{e}} - m_k^{\mathrm{w}}\right)$$

With first-order upwind $m_k^{\mathrm{e}} = m_k^{\mathrm{P}}$ and $m_k^{\mathrm{w}} = m_k^{\mathrm{W}}$

Spatial discretization schemes based on first-order upwind always result in REALIZABLE moments
Higher-order schemes (CDS, second-order upwind, QUICK, MUSCL) always result in UNREALIZABLE moments



Spatial Discretization in QMOM (1-D)

Solve transport equation for generic moment m_k with discretization schemes that PRESERVE realizable moments (constant $\mathrm{U_p}$):

$$\frac{\partial m_k}{\partial t} + \mathrm{U_p}\frac{\partial m_k}{\partial x} = \overline{S}_k - \mathrm{U_p}\frac{\partial m_k}{\partial x}$$

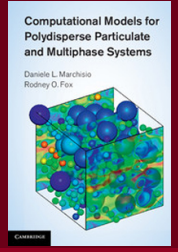
After spatial discretization (finite-volume method):

$$\frac{\mathrm{d}m_k^{\mathrm{P}}}{\mathrm{d}t} = \overline{S}_k^{\mathrm{P}} - \frac{\mathrm{U_p}}{\Delta x}\left(m_k^{\mathrm{e}} - m_k^{\mathrm{w}}\right)$$

With first-order upwind $m_k^{\mathrm{e}} = m_k^{\mathrm{P}}$ and $m_k^{\mathrm{w}} = m_k^{\mathrm{W}}$

Spatial discretization schemes based on first-order upwind always result in REALIZABLE moments

Higher-order schemes (CDS, second-order upwind, QUICK, MUSCL) always result in UNREALIZABLE moments



Spatial Discretization in QMOM (1-D)

Solve transport equation for generic moment m_k with discretization schemes that PRESERVE realizable moments (constant $\mathrm{U_p}$):

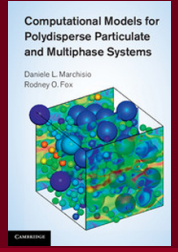
$$\frac{\partial m_k}{\partial t} = \overline{S}_k - \mathrm{U_p}\frac{\partial m_k}{\partial x}$$

After spatial discretization (finite-volume method):

$$\frac{\mathrm{d}m_k^{\mathrm{P}}}{\mathrm{d}t} = \overline{S}_k^{\mathrm{P}} - \frac{\mathrm{U_p}}{\Delta x}\left(m_k^{\mathrm{e}} - m_k^{\mathrm{w}}\right)$$

With first-order upwind $m_k^{\mathrm{e}} = m_k^{\mathrm{P}}$ and $m_k^{\mathrm{w}} = m_k^{\mathrm{W}}$

Spatial discretization schemes based on first-order upwind always result in REALIZABLE moments
Higher-order schemes (CDS, second-order upwind, QUICK, MUSCL) always result in UNREALIZABLE moments

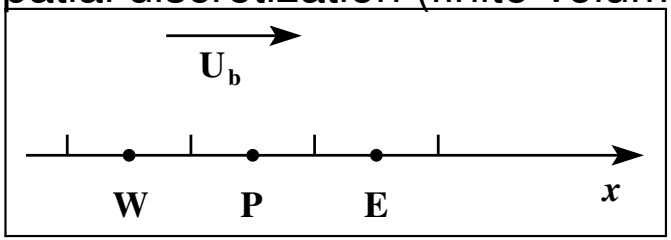


Spatial Discretization in QMOM (1-D)

Solve transport equation for generic moment m_k with discretization schemes that PRESERVE realizable moments (constant $\mathrm{U_p}$):

$$\frac{\partial m_k}{\partial t} = \overline{S}_k - \mathrm{U_p}\frac{\partial m_k}{\partial x}$$

After spatial discretization (finite-volume method):

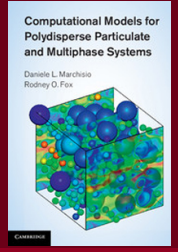


$$\frac{\mathrm{d}m_k^{\mathbf{P}}}{\mathrm{d}t} = \overline{S}_k^{\mathbf{P}} - \frac{\mathrm{U_p}}{\Delta x}\left(m_k^{\mathbf{e}} - m_k^{\mathbf{w}}\right)$$

With first-order upwind $m_k^{\mathbf{e}} = m_k^{\mathbf{P}}$ and $m_k^{\mathbf{w}} = m_k^{\mathbf{W}}$

Spatial discretization schemes based on first-order upwind always result in REALIZABLE moments

Higher-order schemes (CDS, second-order upwind, QUICK, MUSCL) always result in UNREALIZABLE moments

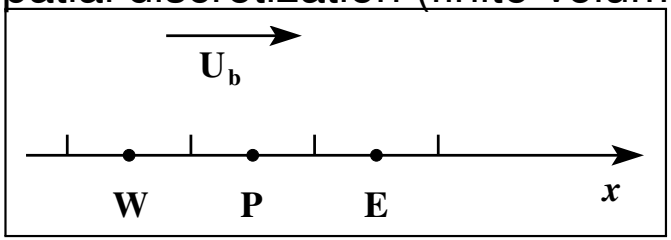


Spatial Discretization in QMOM (1-D)

Solve transport equation for generic moment m_k with discretization schemes that PRESERVE realizable moments (constant $\mathrm{U_p}$):

$$\frac{\partial m_k}{\partial t} = \overline{S}_k - \mathrm{U_p}\frac{\partial m_k}{\partial x}$$

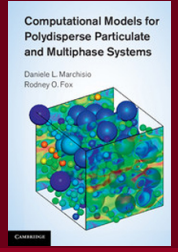
After spatial discretization (finite-volume method):



$$\frac{\mathrm{d}m_k^{\mathbf{P}}}{\mathrm{d}t} = \overline{S}_k^{\mathbf{P}} - \frac{\mathrm{U_p}}{\Delta x}\left(m_k^{\mathbf{e}} - m_k^{\mathbf{w}}\right)$$

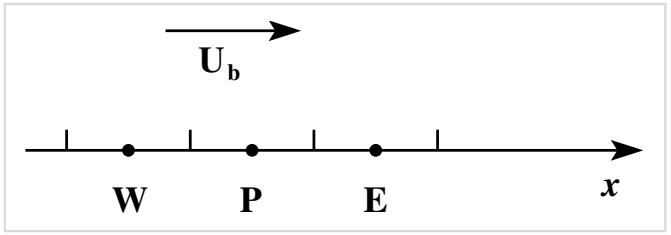
With first-order upwind $m_k^{\mathbf{e}} = m_k^{\mathbf{P}}$ and $m_k^{\mathbf{w}} = m_k^{\mathbf{W}}$

Spatial discretization schemes based on first-order upwind always result in REALIZABLE moments
Higher-order schemes (CDS, second-order upwind, QUICK, MUSCL) always result in UNREALIZABLE moments

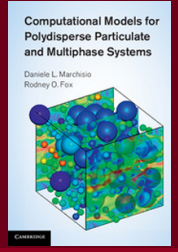


Spatial Discretization in QMOM (1-D)

- One solution would be to evaluate moments at faces m_k^{e} and m_k^{w} through quadrature approximation
- We know value of cell-average moments m_k^{W}, m_k^{P}, m_k^{E}

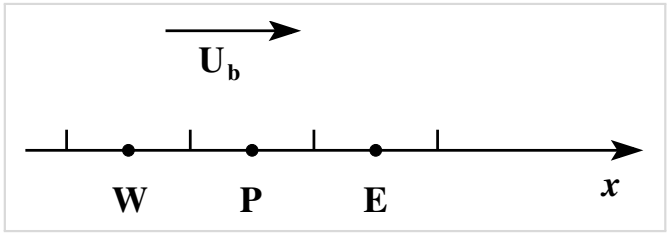


- From these moments we can evaluate corresponding weights w_α^{P} and abscissas ξ_α^{P}
- If weights at center of face are interpolated with p^{th}-order spatial reconstruction and abscissas are interpolated 1^{st}-order spatial resulting moments will be valid if time step is well chosen
- This allows to improve numerical accuracy preserving realizable moments!

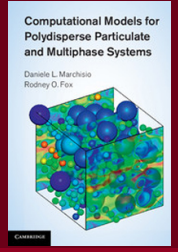


Spatial Discretization in QMOM (1-D)

- One solution would be to evaluate moments at faces m_k^{e} and m_k^{w} through quadrature approximation
- We know value of cell-average moments m_k^{W}, m_k^{P}, m_k^{E}

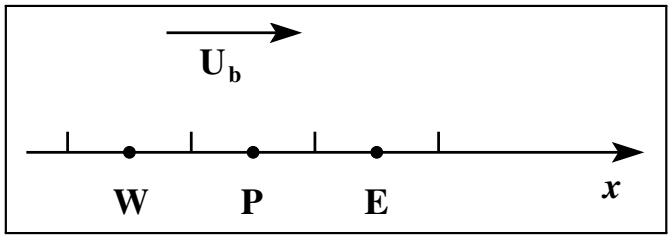


- From these moments we can evaluate corresponding weights w_α^{P} and abscissas ξ_α^{P}
- If weights at center of face are interpolated with p^{th}-order spatial reconstruction and abscissas are interpolated 1^{st}-order spatial resulting moments will be valid if time step is well chosen
- This allows to improve numerical accuracy preserving realizable moments!

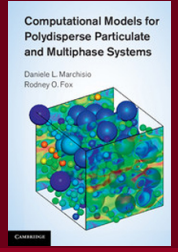


Spatial Discretization in QMOM (1-D)

- One solution would be to evaluate moments at faces $m_k^{\mathbf{e}}$ and $m_k^{\mathbf{w}}$ through quadrature approximation
- We know value of cell-average moments $m_k^{\mathbf{W}}$, $m_k^{\mathbf{P}}$, $m_k^{\mathbf{E}}$

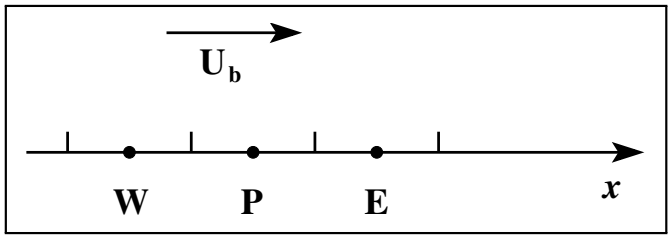


- From these moments we can evaluate corresponding weights $w_\alpha^{\mathbf{P}}$ and abscissas $\xi_\alpha^{\mathbf{P}}$
- If weights at center of face are interpolated with p^{th}-order spatial reconstruction and abscissas are interpolated 1^{st}-order spatial resulting moments will be valid if time step is well chosen
- This allows to improve numerical accuracy preserving realizable moments!

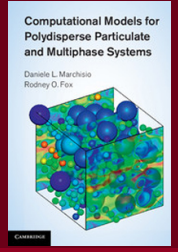


Spatial Discretization in QMOM (1-D)

- One solution would be to evaluate moments at faces $m_k^{\mathbf{e}}$ and $m_k^{\mathbf{w}}$ through quadrature approximation
- We know value of cell-average moments $m_k^{\mathbf{W}}$, $m_k^{\mathbf{P}}$, $m_k^{\mathbf{E}}$

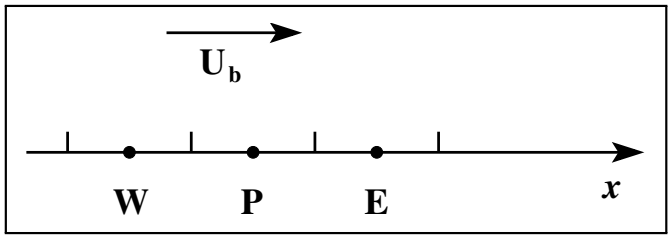


- From these moments we can evaluate corresponding weights $w_\alpha^{\mathbf{P}}$ and abscissas $\xi_\alpha^{\mathbf{P}}$
- If weights at center of face are interpolated with p^{th}-order spatial reconstruction and abscissas are interpolated 1^{st}-order spatial resulting moments will be valid if time step is well chosen
- This allows to improve numerical accuracy preserving realizable moments!

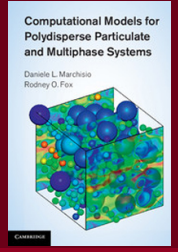


Spatial Discretization in QMOM (1-D)

- One solution would be to evaluate moments at faces $m_k^{\mathbf{e}}$ and $m_k^{\mathbf{w}}$ through quadrature approximation
- We know value of cell-average moments $m_k^{\mathbf{W}}$, $m_k^{\mathbf{P}}$, $m_k^{\mathbf{E}}$

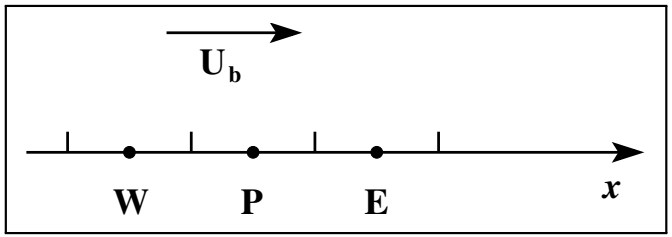


- From these moments we can evaluate corresponding weights $w_\alpha^{\mathbf{P}}$ and abscissas $\xi_\alpha^{\mathbf{P}}$
- If weights at center of face are interpolated with p^{th}-order spatial reconstruction and abscissas are interpolated 1^{st}-order spatial resulting moments will be valid if time step is well chosen
- This allows to improve numerical accuracy preserving realizable moments!

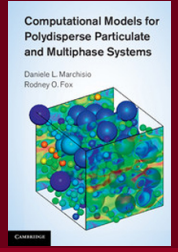


Spatial Discretization in QMOM (1-D)

- One solution would be to evaluate moments at faces $m_k^{\mathbf{e}}$ and $m_k^{\mathbf{w}}$ through quadrature approximation
- We know value of cell-average moments $m_k^{\mathbf{W}}$, $m_k^{\mathbf{P}}$, $m_k^{\mathbf{E}}$

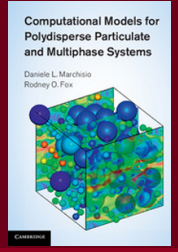


- From these moments we can evaluate corresponding weights $w_\alpha^{\mathbf{P}}$ and abscissas $\xi_\alpha^{\mathbf{P}}$
- If weights at center of face are interpolated with p^{th}-order spatial reconstruction and abscissas are interpolated 1^{st}-order spatial resulting moments will be valid if time step is well chosen
- This allows to improve numerical accuracy preserving realizable moments!



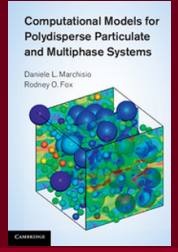
Multivariate QBMM

- Also for multivariate cases QMOM can be used
- With QMOM transport equations for moments $m_{\mathbf{k}}$ are solved, closure problem is overcome with quadrature approximation, and N weights w_α and N vector abscissas (or nodes) of length M, $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are calculated using inversion (multidimensional) algorithms
- With DQMOM transport equations for weights w_α and for weighted vector abscissas (or nodes) $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are solved, closure problem is overcome with quadrature approximation, and source terms are calculated by solving linear system
- All issues related to spatially inhomogeneous systems are still valid, so QMOM is preferred to conserve moments during transport



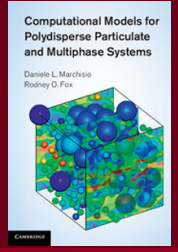
Multivariate QBMM

- Also for multivariate cases QMOM can be used
- With QMOM transport equations for moments $m_{\mathbf{k}}$ are solved, closure problem is overcome with quadrature approximation, and N weights w_α and N vector abscissas (or nodes) of length M, $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are calculated using inversion (multidimensional) algorithms
- With DQMOM transport equations for weights w_α and for weighted vector abscissas (or nodes) $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are solved, closure problem is overcome with quadrature approximation, and source terms are calculated by solving linear system
- All issues related to spatially inhomogeneous systems are still valid, so QMOM is preferred to conserve moments during transport



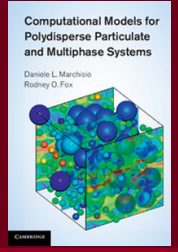
Multivariate QBMM

- Also for multivariate cases QMOM can be used
- With QMOM transport equations for moments $m_{\mathbf{k}}$ are solved, closure problem is overcome with quadrature approximation, and N weights w_α and N vector abscissas (or nodes) of length M, $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are calculated using inversion (multidimensional) algorithms
- With DQMOM transport equations for weights w_α and for weighted vector abscissas (or nodes) $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are solved, closure problem is overcome with quadrature approximation, and source terms are calculated by solving linear system
- All issues related to spatially inhomogeneous systems are still valid, so QMOM is preferred to conserve moments during transport



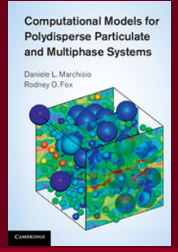
Multivariate QBMM

- Also for multivariate cases QMOM can be used
- With QMOM transport equations for moments $m_{\mathbf{k}}$ are solved, closure problem is overcome with quadrature approximation, and N weights w_α and N vector abscissas (or nodes) of length M, $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are calculated using inversion (multidimensional) algorithms
- With DQMOM transport equations for weights w_α and for weighted vector abscissas (or nodes) $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are solved, closure problem is overcome with quadrature approximation, and source terms are calculated by solving linear system
- All issues related to spatially inhomogeneous systems are still valid, so QMOM is preferred to conserve moments during transport



Multivariate QBMM

- Also for multivariate cases QMOM can be used
- With QMOM transport equations for moments $m_{\mathbf{k}}$ are solved, closure problem is overcome with quadrature approximation, and N weights w_α and N vector abscissas (or nodes) of length M, $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are calculated using inversion (multidimensional) algorithms
- With DQMOM transport equations for weights w_α and for weighted vector abscissas (or nodes) $\boldsymbol{\xi}_\alpha = (\xi_{1,\alpha}, \xi_{2,\alpha}, \ldots, \xi_{M,\alpha})$ are solved, closure problem is overcome with quadrature approximation, and source terms are calculated by solving linear system
- All issues related to spatially inhomogeneous systems are still valid, so QMOM is preferred to conserve moments during transport



Multivariate QBMM

Consider

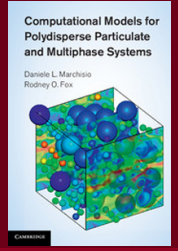
$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_p(\boldsymbol{\xi}) n = \frac{\partial}{\partial \mathbf{x}} \cdot D(\boldsymbol{\xi}) \frac{\partial n}{\partial \mathbf{x}} - \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + h$$

and define generic integer moment:

$$m_{\mathbf{k}} := \int_{\Omega_\xi} \xi_1^{k_1} \cdots \xi_M^{k_M} n(t, \boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}$$

equivalent notations $m_{k_1,\ldots,k_M} = m_{\mathbf{k}} = m(k_1, \ldots, k_M) = m(\mathbf{k})$
Solve resulting moment transport equations (**QMOM**):

$$\frac{\partial m_{\mathbf{k}}}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{U}_{\mathbf{k}} m_{\mathbf{k}} - \frac{\partial}{\partial \mathbf{x}} \cdot D_{\mathbf{k}} \frac{\partial m_{\mathbf{k}}}{\partial \mathbf{x}} = \overline{S}_{\mathbf{k}} = \int_{\Omega_\xi} \boldsymbol{\xi}^{\mathbf{k}} \left[-\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}} n + h \right] \mathrm{d}\boldsymbol{\xi}$$



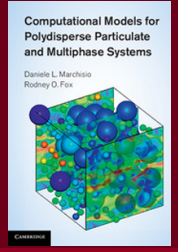
Multidimensional Inversion Algorithms

Brute-force QMOM uses $N(M+1)$ moments to determine quadrature approximation by solving non-linear system:

$$m_{k_{i1},k_{i2},\ldots,k_{iM}} = m(\mathbf{k}_i) = \sum_{\alpha=1}^{N} w_\alpha \prod_{\beta=1}^{M} \xi_{\beta,\alpha}^{k_{i\beta}} \quad 1 \le i \le N(M+1)$$

obtained with $N(M+1)$ different values of exponent vector $\mathbf{k}_i = (k_{i1}, k_{i2}, \ldots, k_{iM})$; for $M = 2$ and $N = 2$

$$\mathbf{K} = \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \\ \mathbf{k}_4 \\ \mathbf{k}_5 \\ \mathbf{k}_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 2 & 0 \\ 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{matrix} \leftarrow \gamma_1 = 0 \\ \leftarrow \gamma_2 = 1 \\ \leftarrow \gamma_3 = 1 \\ \leftarrow \gamma_4 = 2 \\ \leftarrow \gamma_5 = 2 \\ \leftarrow \gamma_6 = 2 \end{matrix}$$



Multidimensional Inversion Algorithms

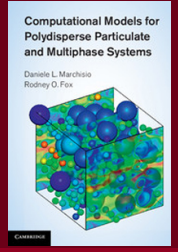
Let us write problem in terms of weights and weighted abscissas (instead of abscissas). The final non-linear system of $N(M+1)$ equations can be solved by employing Newton-Raphson iterative scheme:

$$\mathbf{Z}_{n+1} = \mathbf{Z}_n - \mathbf{A}^{-1}(\mathbf{K}, \mathbf{X}_n)\mathbf{F}(\mathbf{Z}_n)$$

Matrix $\mathbf{A}$ is the Jacobian of non-linear system:

$$a_{ij} = \begin{cases} (1-\gamma_i)\prod_{\alpha=1}^{M} \xi_{\alpha,p}^{k_{i\alpha}} & \text{for } p = j, \text{ if } 1 \le j \le N \\ \left(\frac{k_{i1}}{\xi_{1,p}}\right)\prod_{\alpha=1}^{M} \xi_{\alpha,p}^{k_{i\alpha}} & \text{for } p = j-N, \text{ if } N+1 \le j \le 2N \\ \vdots & \vdots \\ \left(\frac{k_{iM}}{\xi_{M,p}}\right)\prod_{\alpha=1}^{M} \xi_{\alpha,M}^{k_{i\alpha}} & \text{for } p = j-MN, \text{ if } MN+1 \le j \le (1+M)N \end{cases}$$

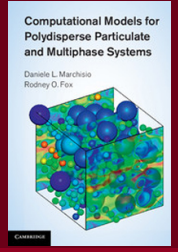
and is identical to the matrix of previous linear system



Optimal Moment Set

- To perform calculations with both multivariate QMOM[1] matrix $\mathbf{A}$ must be non-singular (full rank)
- For $M = 1$ (univariate) problems this requirement is satisfied if nodes are distinct
- For multivariate cases, having distinct abscissas does not guarantee that $\mathbf{A}$ will be full rank
- It can be shown that for fixed N and M, certain distinct moments are linearly dependent when $M \geq 1$ for all possible sets of abscissas
- It is therefore necessary to identify moment set for which $\mathbf{A}$ is always non-singular for all non-degenerate points in phase space for given values of M and N

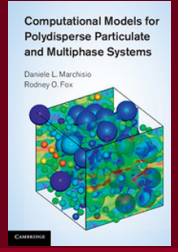
[1] If Brute-Force inversion algorithm is used



Optimal Moment Set

- To perform calculations with both multivariate QMOM[1] matrix $\mathbf{A}$ must be non-singular (full rank)
- For $M = 1$ (univariate) problems this requirement is satisfied if nodes are distinct
- For multivariate cases, having distinct abscissas does not guarantee that $\mathbf{A}$ will be full rank
- It can be shown that for fixed N and M, certain distinct moments are linearly dependent when $M \geq 1$ for all possible sets of abscissas
- It is therefore necessary to identify moment set for which $\mathbf{A}$ is always non-singular for all non-degenerate points in phase space for given values of M and N

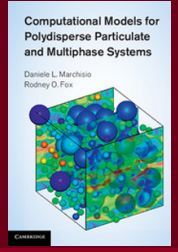
[1] If Brute-Force inversion algorithm is used



Optimal Moment Set

- To perform calculations with both multivariate QMOM[1] matrix $\mathbf{A}$ must be non-singular (full rank)
- For $M = 1$ (univariate) problems this requirement is satisfied if nodes are distinct
- For multivariate cases, having distinct abscissas does not guarantee that $\mathbf{A}$ will be full rank
- It can be shown that for fixed N and M, certain distinct moments are linearly dependent when $M \geq 1$ for all possible sets of abscissas
- It is therefore necessary to identify moment set for which $\mathbf{A}$ is always non-singular for all non-degenerate points in phase space for given values of M and N

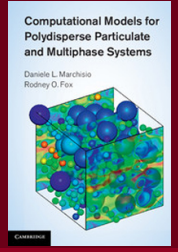
[1] If Brute-Force inversion algorithm is used



Optimal Moment Set

- To perform calculations with both multivariate QMOM[1] matrix $\mathbf{A}$ must be non-singular (full rank)
- For $M = 1$ (univariate) problems this requirement is satisfied if nodes are distinct
- For multivariate cases, having distinct abscissas does not guarantee that $\mathbf{A}$ will be full rank
- It can be shown that for fixed N and M, certain distinct moments are linearly dependent when $M \geq 1$ for all possible sets of abscissas
- It is therefore necessary to identify moment set for which $\mathbf{A}$ is always non-singular for all non-degenerate points in phase space for given values of M and N

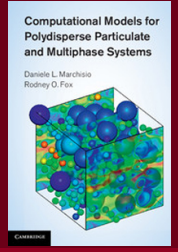
[1] If Brute-Force inversion algorithm is used



Optimal Moment Set

- To perform calculations with both multivariate QMOM[1] matrix $\mathbf{A}$ must be non-singular (full rank)
- For $M = 1$ (univariate) problems this requirement is satisfied if nodes are distinct
- For multivariate cases, having distinct abscissas does not guarantee that $\mathbf{A}$ will be full rank
- It can be shown that for fixed N and M, certain distinct moments are linearly dependent when $M \geq 1$ for all possible sets of abscissas
- It is therefore necessary to identify moment set for which $\mathbf{A}$ is always non-singular for all non-degenerate points in phase space for given values of M and N

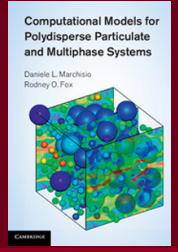
[1] If Brute-Force inversion algorithm is used



Optimal Moment Set

- To perform calculations with both multivariate QMOM[1] matrix $\mathbf{A}$ must be non-singular (full rank)
- For $M = 1$ (univariate) problems this requirement is satisfied if nodes are distinct
- For multivariate cases, having distinct abscissas does not guarantee that $\mathbf{A}$ will be full rank
- It can be shown that for fixed N and M, certain distinct moments are linearly dependent when $M \geq 1$ for all possible sets of abscissas
- It is therefore necessary to identify moment set for which $\mathbf{A}$ is always non-singular for all non-degenerate points in phase space for given values of M and N

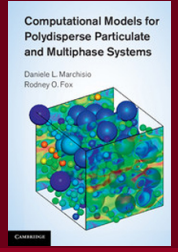
[1] If Brute-Force inversion algorithm is used



Optimal Moment Set

OPTIMAL MOMENT SET

1. Optimal moment set consists of $N(M+1)$ distinct moments
2. Optimal moment set will result in full-rank square matrix $\mathbf{A}$ for all possible sets of N distinct, non-degenerate abscissas
3. Optimal moment set includes all linearly independent moments of a particular order γ_i before adding moments of higher order in order to result in perfectly symmetric treatment of internal coordinates



Optimal Moment Set

Moments used for bivariate quadrature approximation ($M = 2$) for $N = 2$

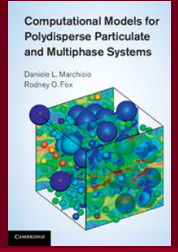
m(2,0)			
m(1,0)			
m(0,0)	m(0,1)	m(0,2)	m(0,3)

In this case $m_{0,3}$ is chosen as the third-order moment to saturate degrees of freedom

Moments used for bivariate quadrature approximation ($M = 2$) for $N = 3$

m(2,0)	m(2,1)		
m(1,0)	m(1,1)	m(1,2)	
m(0,0)	m(0,1)	m(0,2)	m(0,3)

In this case $m_{2,1}$, $m_{1,2}$ and $m_{0,3}$ are chosen among third-order moments to saturate the degrees of freedom



Optimal Moment Set

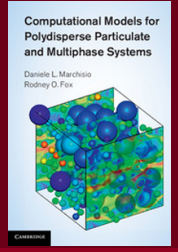
Optimal moment set for bivariate quadrature approximation ($M = 2$) for $N = 4$

m(3,0)	m(3,1)		
m(2,0)	m(2,1)		
m(1,0)	m(1,1)	m(1,2)	m(1,3)
m(0,0)	m(0,1)	m(0,2)	m(0,3)

When $N^{1/M}$ is an integer, there exists an optimal moment set that is also symmetric

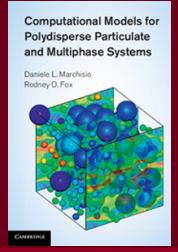
Optimal moment for bivariate quadrature approximation ($M = 2$) for $N = 9$

m(5,0)	m(5,1)	m(5,2)			
m(4,0)	m(4,1)	m(4,2)			
m(3,0)	m(3,1)	m(3,2)			
m(2,0)	m(2,1)	m(2,2)	m(2,3)	m(2,4)	m(2,5)
m(1,0)	m(1,1)	m(1,2)	m(1,3)	m(1,4)	m(1,5)
m(0,0)	m(0,1)	m(0,2)	m(0,3)	m(0,4)	m(0,5)



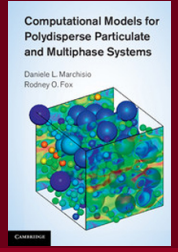
Other Inversion Algorithms

- Brute-force methods need an initial guess very (very (very)) close to final solution
- Alternative approach is **Tensor-product QMOM**
- First M univariate quadratures are calculated from 'pure' moments in different directions
- These M quadratures nodes are used to define final nodes by tensor product
- Weights are calculated subsequently by forcing quadrature to reproduce moment set (by solving linear system)
- Example for $M = 2$



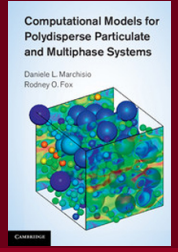
Other Inversion Algorithms

- Brute-force methods need an initial guess very (very (very)) close to final solution
- Alternative approach is **Tensor-product QMOM**
- First M univariate quadratures are calculated from 'pure' moments in different directions
- These M quadratures nodes are used to define final nodes by tensor product
- Weights are calculated subsequently by forcing quadrature to reproduce moment set (by solving linear system)
- Example for $M = 2$



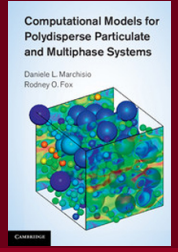
Other Inversion Algorithms

- Brute-force methods need an initial guess very (very (very)) close to final solution
- Alternative approach is **Tensor-product QMOM**
- First M univariate quadratures are calculated from 'pure' moments in different directions
- These M quadratures nodes are used to define final nodes by tensor product
- Weights are calculated subsequently by forcing quadrature to reproduce moment set (by solving linear system)
- Example for $M = 2$



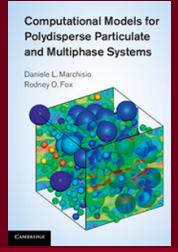
Other Inversion Algorithms

- Brute-force methods need an initial guess very (very (very)) close to final solution
- Alternative approach is **Tensor-product QMOM**
- First M univariate quadratures are calculated from 'pure' moments in different directions
- These M quadratures nodes are used to define final nodes by tensor product
- Weights are calculated subsequently by forcing quadrature to reproduce moment set (by solving linear system)
- Example for $M = 2$



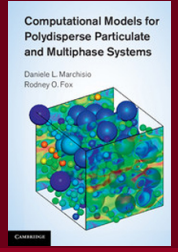
Other Inversion Algorithms

- Brute-force methods need an initial guess very (very (very)) close to final solution
- Alternative approach is **Tensor-product QMOM**
- First M univariate quadratures are calculated from 'pure' moments in different directions
- These M quadratures nodes are used to define final nodes by tensor product
- Weights are calculated subsequently by forcing quadrature to reproduce moment set (by solving linear system)
- Example for $M = 2$



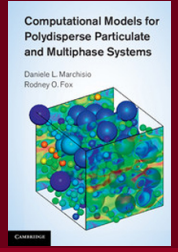
Other Inversion Algorithms

- Brute-force methods need an initial guess very (very (very)) close to final solution
- Alternative approach is **Tensor-product QMOM**
- First M univariate quadratures are calculated from 'pure' moments in different directions
- These M quadratures nodes are used to define final nodes by tensor product
- Weights are calculated subsequently by forcing quadrature to reproduce moment set (by solving linear system)
- Example for $M = 2$



Other Inversion Algorithms

- Brute-force methods need an initial guess very (very (very)) close to final solution
- Alternative approach is **Tensor-product QMOM**
- First M univariate quadratures are calculated from 'pure' moments in different directions
- These M quadratures nodes are used to define final nodes by tensor product
- Weights are calculated subsequently by forcing quadrature to reproduce moment set (by solving linear system)
- Example for $M = 2$



Tensor-Product Inversion Algorithm

4-point quadrature approximation for bivariate distribution

Using PD (or Wheeler) algorithm:

$$(m_{0,0}, m_{1,0}, m_{2,0}, m_{3,0}) \rightarrow \left(w_1^1, w_2^1; \xi_{1,1}^*, \xi_{1,2}^*\right)$$

$$(m_{0,0}, m_{0,1}, m_{0,2}, m_{0,3}) \rightarrow \left(w_1^2, w_2^2; \xi_{2,1}^*, \xi_{2,2}^*\right)$$

Using tensor product, final 4-point quadrature approximation is centered on bivariate nodes:

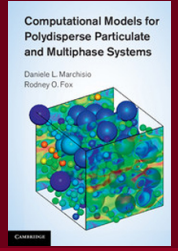
$$\boldsymbol{\xi}_1 = \left[\xi_{1,1}^*, \xi_{2,1}^*\right]$$

$$\boldsymbol{\xi}_2 = \left[\xi_{1,1}^*, \xi_{2,2}^*\right]$$

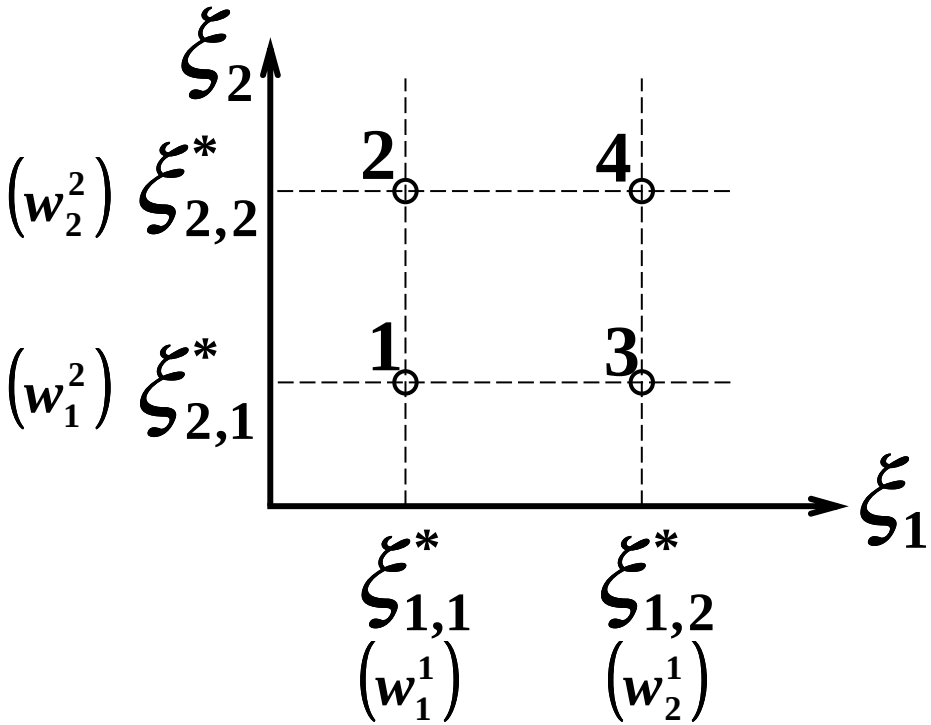
$$\boldsymbol{\xi}_3 = \left[\xi_{1,2}^*, \xi_{2,1}^*\right]$$

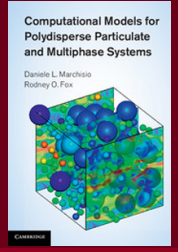
$$\boldsymbol{\xi}_4 = \left[\xi_{1,2}^*, \xi_{2,2}^*\right]$$

Four weights w_1, w_2, w_3, w_4 are calculated by forcing moment subset to be reproduced



Tensor-Product Inversion Algorithm



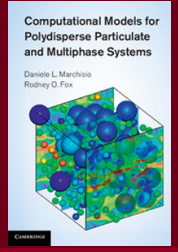


Tensor-Product Inversion Algorithm

Moment set used to build a bivariate quadrature approximation ($M = 2$) for $N = N_1 N_2 = 4$ with tensor-product method

m(3,0)			
m(2,0)			
m(1,0)	m(1,1)		
m(0,0)	m(0,1)	m(0,2)	m(0,3)

- Moments accommodated by quadrature approximation are: $N_1 N_2 \cdots N_M + N_1 + N_2 + \cdots + N_M$, less than maximum: $(M+1)N$
- For example for $M = 2$ for $N = 4$ instead of 12 moments used and correctly represented are 8!
- Depending on shape of distribution negative weights are sometimes obtained (particularly for large N)

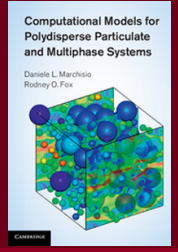


Tensor-Product Inversion Algorithm

Moment set used to build a bivariate quadrature approximation ($M = 2$) for $N = N_1 N_2 = 4$ with tensor-product method

m(3,0)			
m(2,0)			
m(1,0)	m(1,1)		
m(0,0)	m(0,1)	m(0,2)	m(0,3)

- Moments accommodated by quadrature approximation are: $N_1 N_2 \cdots N_M + N_1 + N_2 + \cdots + N_M$, less than maximum: $(M+1)N$
- For example for $M = 2$ for $N = 4$ instead of 12 moments used and correctly represented are 8!
- Depending on shape of distribution negative weights are sometimes obtained (particularly for large N)

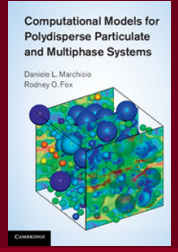


Tensor-Product Inversion Algorithm

Moment set used to build a bivariate quadrature approximation ($M = 2$) for $N = N_1 N_2 = 4$ with tensor-product method

m(3,0)			
m(2,0)			
m(1,0)	m(1,1)		
m(0,0)	m(0,1)	m(0,2)	m(0,3)

- Moments accommodated by quadrature approximation are: $N_1 N_2 \cdots N_M + N_1 + N_2 + \cdots + N_M$, less than maximum: $(M+1)N$
- For example for $M = 2$ for $N = 4$ instead of 12 moments used and correctly represented are 8!
- Depending on shape of distribution negative weights are sometimes obtained (particularly for large N)

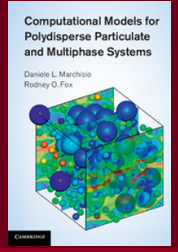


Tensor-Product Inversion Algorithm

Moment set used to build a bivariate quadrature approximation ($M = 2$) for $N = N_1 N_2 = 4$ with tensor-product method

m(3,0)			
m(2,0)			
m(1,0)	m(1,1)		
m(0,0)	m(0,1)	m(0,2)	m(0,3)

- Moments accommodated by quadrature approximation are: $N_1 N_2 \cdots N_M + N_1 + N_2 + \cdots + N_M$, less than maximum: $(M+1)N$
- For example for $M = 2$ for $N = 4$ instead of 12 moments used and correctly represented are 8!
- Depending on shape of distribution negative weights are sometimes obtained (particularly for large N)

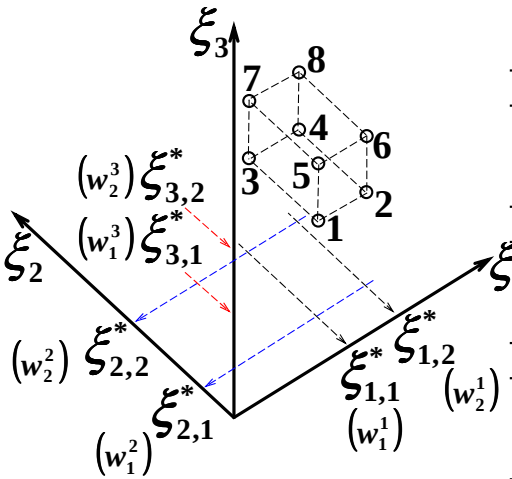


Tensor-Product Inversion Algorithm

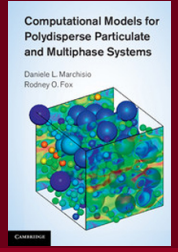
8-point trivariate quadrature approximation

$(M = 3, N_1 = N_2 = N_3 = 2, N = N_1 N_2 N_3 = 8)$

Optimal set $(N(M+1))$: 32 moments
Moments actually used: 14



i	1^*	2^*	3^*	4^*	5^*	6^*	7^*	8^*	9^*	10^*
k_{i1}	0	1	0	0	2	1	1	0	0	0
k_{i2}	0	0	1	0	0	1	0	2	1	0
k_{i3}	0	0	0	1	0	0	1	0	1	2
i	11^*	12	13	14	15^*	16	17^*	18	19	20^*
k_{i1}	3	2	2	1	1	1	0	0	0	0
k_{i2}	0	1	0	2	1	0	3	2	1	0
k_{i3}	0	0	1	0	1	2	0	1	2	3
i	21	22	23	24	25	26	27	28	29	
k_{i1}	3	3	1	1	0	0	2	1	1	
k_{i2}	0	1	3	0	3	1	1	2	1	
k_{i3}	1	0	0	3	1	3	1	1	2	
i	30	31	32							
k_{i1}	3	1	1							
k_{i2}	1	3	1							
k_{i3}	1	1	3							



Inversion Algorithms Based on Conditional Moments

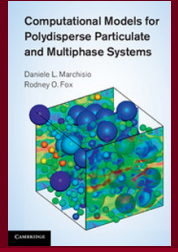
- Methods based on **conditional density functions**:
  $n(\xi_1,\xi_2) = n_1(\xi_1) f_{21}(\xi_2|\xi_1) = n_2(\xi_2) f_{12}(\xi_1|\xi_2)$
- Univariate quadrature (N_1) calculated from the first $2N_1 - 1$:

$$\begin{pmatrix} m_{0,0,\ldots,0,0} \\ \vdots \\ m_{2N_1-1,0,\ldots,0,0} \end{pmatrix} \xrightarrow{\text{PD/Wheeler}} \begin{pmatrix} w_1 \\ \vdots \\ w_{N_1} \end{pmatrix} \begin{pmatrix} \xi_{1;1} \\ \vdots \\ \xi_{1;N_1} \end{pmatrix}$$

  resulting for example in: $n(\xi_1,\xi_2) = \sum_{\alpha_1=1}^{N_1} \delta(\xi_1 - \xi_{1;\alpha}) f_{21}(\xi_2|\xi_1)$
- Generic moment becomes:

$$\begin{aligned} m_{k_1,k_2} &= \iint n(\xi_1,\xi_2)\xi_1^{k_1}\xi_2^{k_2}\,\mathrm{d}\xi_1\,\mathrm{d}\xi_2 \\ &= \sum_{\alpha_1=1}^{N_1} \xi_{1;\alpha}^{k_1} \int f_{21}(\xi_2|\xi_{1;\alpha})\xi_2^{k_2}\,\mathrm{d}\xi_2 \end{aligned}$$

- **Conditional moment**: $\left\langle \xi_2^{k_2} \right\rangle_{\alpha_1} := \int f(\xi_2|\xi_{1;\alpha_1})\xi_2^{k_2}\,\mathrm{d}\xi_2$



Inversion Algorithms Based on Conditional Moments

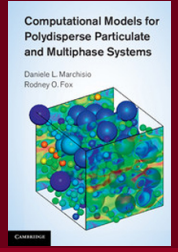
- Methods based on **conditional density functions**:
  $n(\xi_1,\xi_2) = n_1(\xi_1) f_{21}(\xi_2|\xi_1) = n_2(\xi_2) f_{12}(\xi_1|\xi_2)$
- Univariate quadrature (N_1) calculated from the first $2N_1 - 1$:

$$\begin{pmatrix} m_{0,0,\ldots,0,0} \\ \vdots \\ m_{2N_1-1,0,\ldots,0,0} \end{pmatrix} \xrightarrow{\text{PD/Wheeler}} \begin{pmatrix} w_1 \\ \vdots \\ w_{N_1} \end{pmatrix} \begin{pmatrix} \xi_{1;1} \\ \vdots \\ \xi_{1;N_1} \end{pmatrix}$$

  resulting for example in: $n(\xi_1,\xi_2) = \sum_{\alpha_1=1}^{N_1} \delta(\xi_1 - \xi_{1;\alpha}) f_{21}(\xi_2|\xi_1)$
- Generic moment becomes:

$$\begin{aligned} m_{k_1,k_2} &= \iint n(\xi_1,\xi_2)\xi_1^{k_1}\xi_2^{k_2}\,\mathrm{d}\xi_1\,\mathrm{d}\xi_2 \\ &= \sum_{\alpha_1=1}^{N_1} \xi_{1;\alpha}^{k_1} \int f_{21}(\xi_2|\xi_{1;\alpha})\xi_2^{k_2}\,\mathrm{d}\xi_2 \end{aligned}$$

- **Conditional moment**: $\left\langle \xi_2^{k_2} \right\rangle_{\alpha_1} := \int f(\xi_2|\xi_{1;\alpha_1})\xi_2^{k_2}\,\mathrm{d}\xi_2$



Inversion Algorithms Based on Conditional Moments

- Methods based on **conditional density functions**:
  $n(\xi_1,\xi_2) = n_1(\xi_1) f_{21}(\xi_2|\xi_1) = n_2(\xi_2) f_{12}(\xi_1|\xi_2)$
- Univariate quadrature (N_1) calculated from the first $2N_1 - 1$:

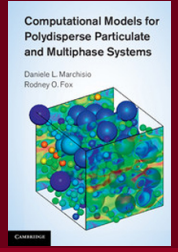
$$\begin{pmatrix} m_{0,0,\ldots,0,0} \\ \vdots \\ m_{2N_1-1,0,\ldots,0,0} \end{pmatrix} \xrightarrow{\text{PD/Wheeler}} \begin{pmatrix} w_1 \\ \vdots \\ w_{N_1} \end{pmatrix} \begin{pmatrix} \xi_{1;1} \\ \vdots \\ \xi_{1;N_1} \end{pmatrix}$$

  resulting for example in: $n(\xi_1,\xi_2) = \sum_{\alpha_1=1}^{N_1} \delta(\xi_1 - \xi_{1;\alpha}) f_{21}(\xi_2|\xi_1)$
- Generic moment becomes:

$$m_{k_1,k_2} = \iint n(\xi_1,\xi_2)\xi_1^{k_1}\xi_2^{k_2}\,\mathrm{d}\xi_1\,\mathrm{d}\xi_2$$

$$= \sum_{\alpha_1=1}^{N_1} \xi_{1;\alpha}^{k_1} \int f_{21}(\xi_2|\xi_{1;\alpha})\xi_2^{k_2}\,\mathrm{d}\xi_2$$

- **Conditional moment**: $\left\langle \xi_2^{k_2} \right\rangle_{\alpha_1} := \int f(\xi_2|\xi_{1;\alpha_1})\xi_2^{k_2}\,\mathrm{d}\xi_2$



Inversion Algorithms Based on Conditional Moments

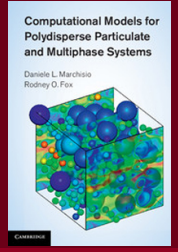
- Methods based on **conditional density functions**:
  $n(\xi_1,\xi_2) = n_1(\xi_1) f_{21}(\xi_2|\xi_1) = n_2(\xi_2) f_{12}(\xi_1|\xi_2)$
- Univariate quadrature (N_1) calculated from the first $2N_1 - 1$:

$$\begin{pmatrix} m_{0,0,\ldots,0,0} \\ \vdots \\ m_{2N_1-1,0,\ldots,0,0} \end{pmatrix} \xrightarrow{\text{PD/Wheeler}} \begin{pmatrix} w_1 \\ \vdots \\ w_{N_1} \end{pmatrix} \begin{pmatrix} \xi_{1;1} \\ \vdots \\ \xi_{1;N_1} \end{pmatrix}$$

  resulting for example in: $n(\xi_1,\xi_2) = \sum_{\alpha_1=1}^{N_1} \delta(\xi_1 - \xi_{1;\alpha}) f_{21}(\xi_2|\xi_1)$
- Generic moment becomes:

$$\begin{aligned} m_{k_1,k_2} &= \iint n(\xi_1,\xi_2)\xi_1^{k_1}\xi_2^{k_2}\,\mathrm{d}\xi_1\,\mathrm{d}\xi_2 \\ &= \sum_{\alpha_1=1}^{N_1} \xi_{1;\alpha}^{k_1} \int f_{21}(\xi_2|\xi_{1;\alpha})\xi_2^{k_2}\,\mathrm{d}\xi_2 \end{aligned}$$

- **Conditional moment**: $\left\langle \xi_2^{k_2} \right\rangle_{\alpha_1} := \int f(\xi_2|\xi_{1;\alpha_1})\xi_2^{k_2}\,\mathrm{d}\xi_2$



Inversion Algorithms Based on Conditional Moments

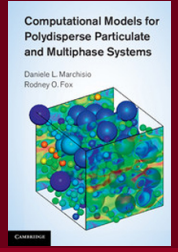
- Methods based on **conditional density functions**:
  $n(\xi_1,\xi_2) = n_1(\xi_1) f_{21}(\xi_2|\xi_1) = n_2(\xi_2) f_{12}(\xi_1|\xi_2)$
- Univariate quadrature (N_1) calculated from the first $2N_1 - 1$:

$$\begin{pmatrix} m_{0,0,\ldots,0,0} \\ \vdots \\ m_{2N_1-1,0,\ldots,0,0} \end{pmatrix} \xrightarrow{\text{PD/Wheeler}} \begin{pmatrix} w_1 \\ \vdots \\ w_{N_1} \end{pmatrix} \begin{pmatrix} \xi_{1;1} \\ \vdots \\ \xi_{1;N_1} \end{pmatrix}$$

  resulting for example in: $n(\xi_1,\xi_2) = \sum_{\alpha_1=1}^{N_1} \delta(\xi_1 - \xi_{1;\alpha}) f_{21}(\xi_2|\xi_1)$
- Generic moment becomes:

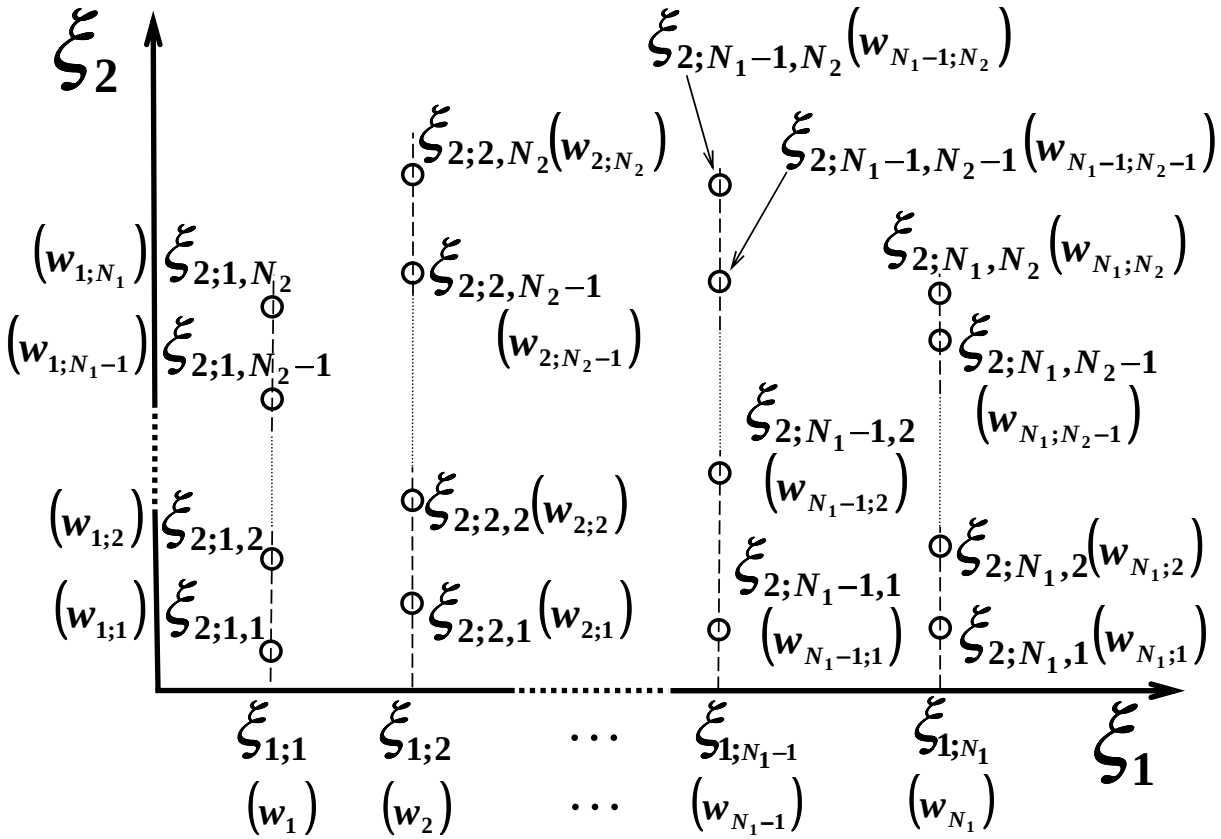
$$\begin{aligned} m_{k_1,k_2} &= \iint n(\xi_1,\xi_2)\xi_1^{k_1}\xi_2^{k_2}\,\mathrm{d}\xi_1\,\mathrm{d}\xi_2 \\ &= \sum_{\alpha_1=1}^{N_1} \xi_{1;\alpha}^{k_1} \int f_{21}(\xi_2|\xi_{1;\alpha})\xi_2^{k_2}\,\mathrm{d}\xi_2 \end{aligned}$$

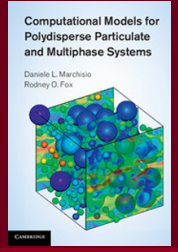
- **Conditional moment**: $\left\langle \xi_2^{k_2} \right\rangle_{\alpha_1} := \int f(\xi_2|\xi_{1;\alpha_1})\xi_2^{k_2}\,\mathrm{d}\xi_2$



Inversion Algorithms Based on Conditional Moments

For each of these N_1 nodes, $2N_2 - 1$ conditional moments are calculated, and univariate quadratures N_2 are determined (in direction ξ_2): **Conditional QMOM or CQMOM**



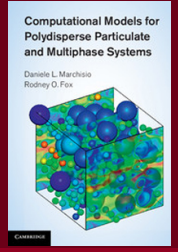


Inversion Algorithms Based on Conditional Moments

Moments used to build a bivariate quadrature approximation ($M = 2$) for $N_1 = N_2 = 3$ using CQMOM with ξ_2 conditioned on ξ_1 (top) and ξ_1 conditioned on ξ_2 (bottom)

m(5,0)					
m(4,0)					
m(3,0)					
m(2,0)	m(2,1)	m(2,2)	m(2,3)	m(2,4)	m(2,5)
m(1,0)	m(1,1)	m(1,2)	m(1,3)	m(1,4)	m(1,5)
m(0,0)	m(0,1)	m(0,2)	m(0,3)	m(0,4)	m(0,5)

m(5,0)	m(5,1)	m(5,2)			
m(4,0)	m(4,1)	m(4,2)			
m(3,0)	m(3,1)	m(3,2)			
m(2,0)	m(2,1)	m(2,2)			
m(1,0)	m(1,1)	m(1,2)			
m(0,0)	m(0,1)	m(0,2)	m(0,3)	m(0,4)	m(0,5)



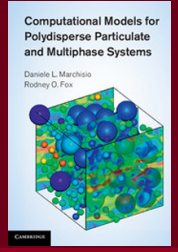
Comparison of Different Algorithms

EXAMPLE

Determine two quadrature approximations of orders 4 and 9 for bivariate Gaussian distribution:

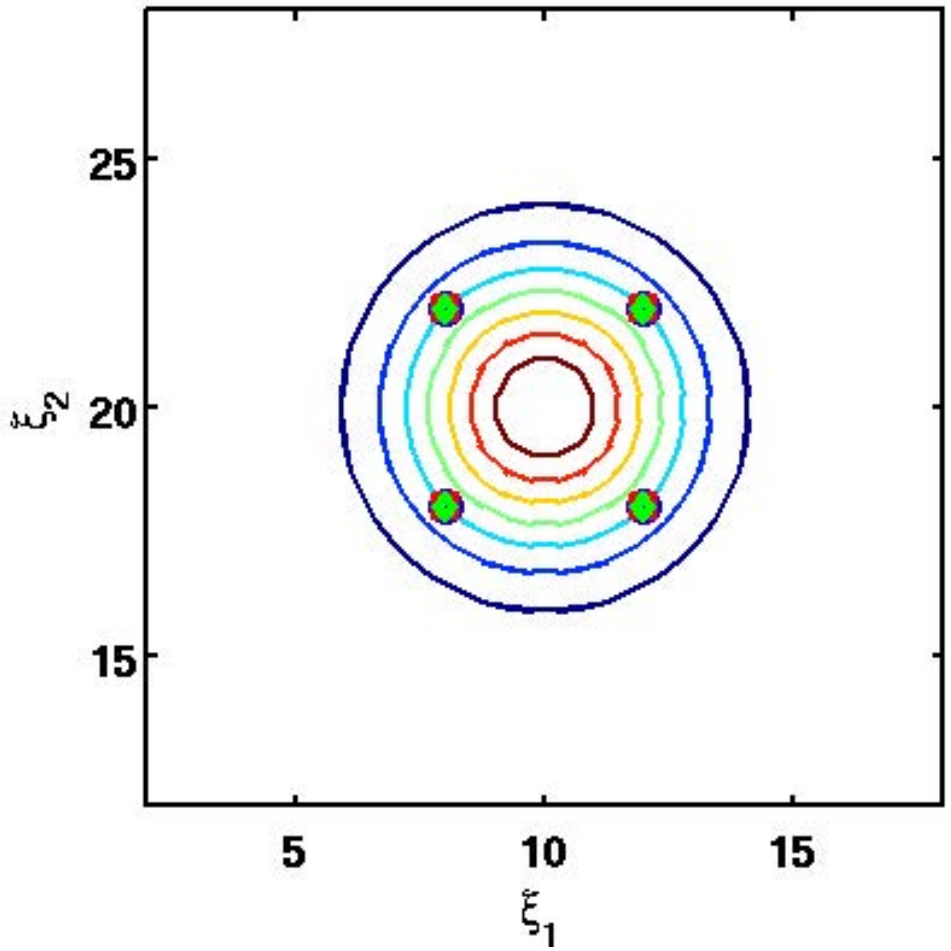
$$f(\xi_1,\xi_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}\times$$
$$\exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(\xi_1-\mu_1)^2}{\sigma_1^2}+\frac{2\rho(\xi_1-\mu_1)(\xi_2-\mu_2)}{\sigma_1\sigma_2}+\frac{(\xi_2-\mu_2)^2}{\sigma_2^2}\right]\right)$$

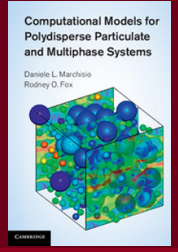
with $\mu_1 = 10$, $\mu_2 = 20$, $\sigma_1 = \sigma_2 = 2$ and different ρ values, by employing the Brute-Force **BF-QMOM**, Tensor-Product **TP-QMOM** and **CQMOM**.



Comparison of Different Algorithms

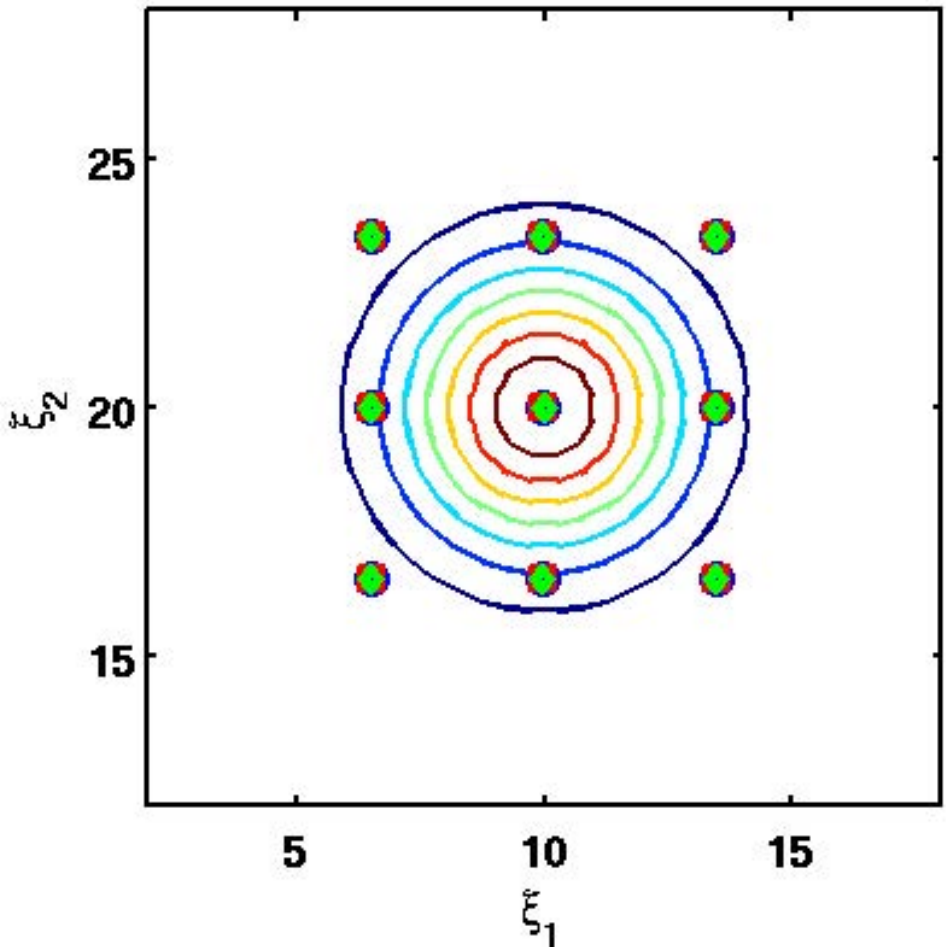
$\rho = 0$ $N = 4$; **BF-QMOM** (diamond), **TP-QMOM** (circle) and **CQMOM** (square)

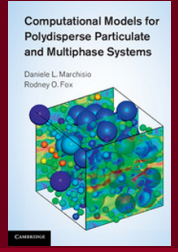




Comparison of Different Algorithms

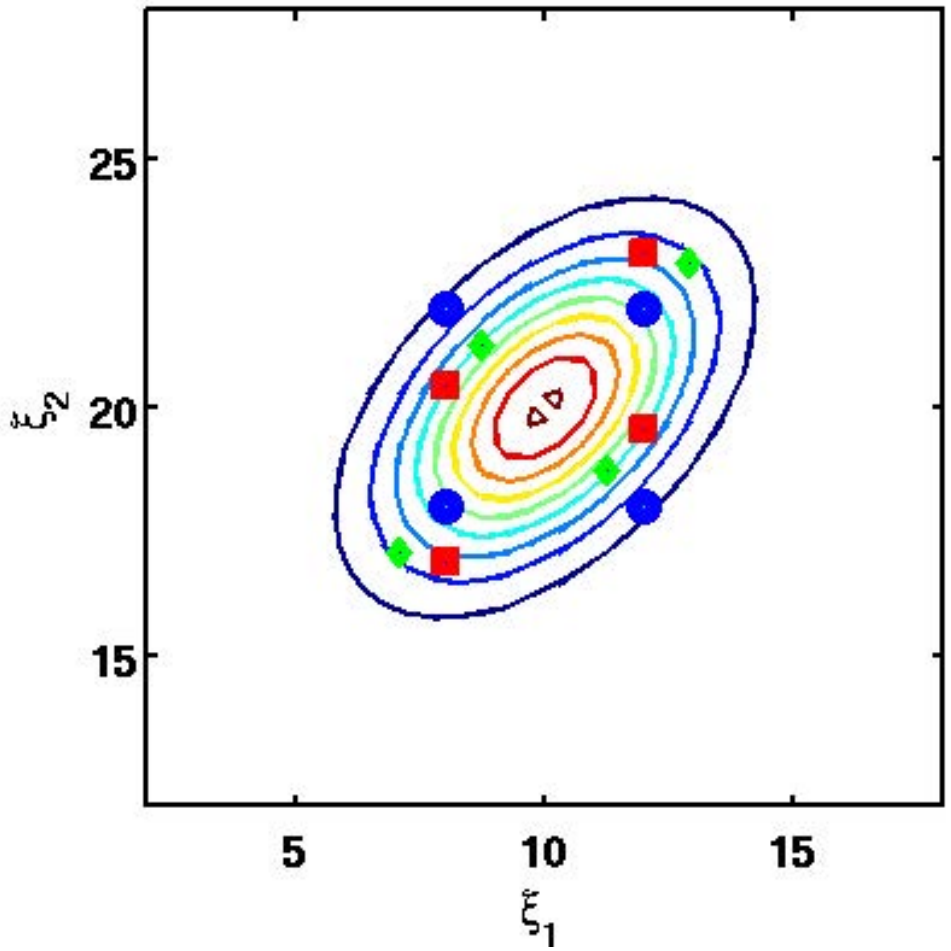
$\rho = 0$ $N = 9$; **BF-QMOM** (diamond), **TP-QMOM** (circle) and **CQMOM** (square)

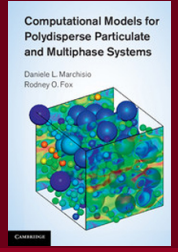




Comparison of Different Algorithms

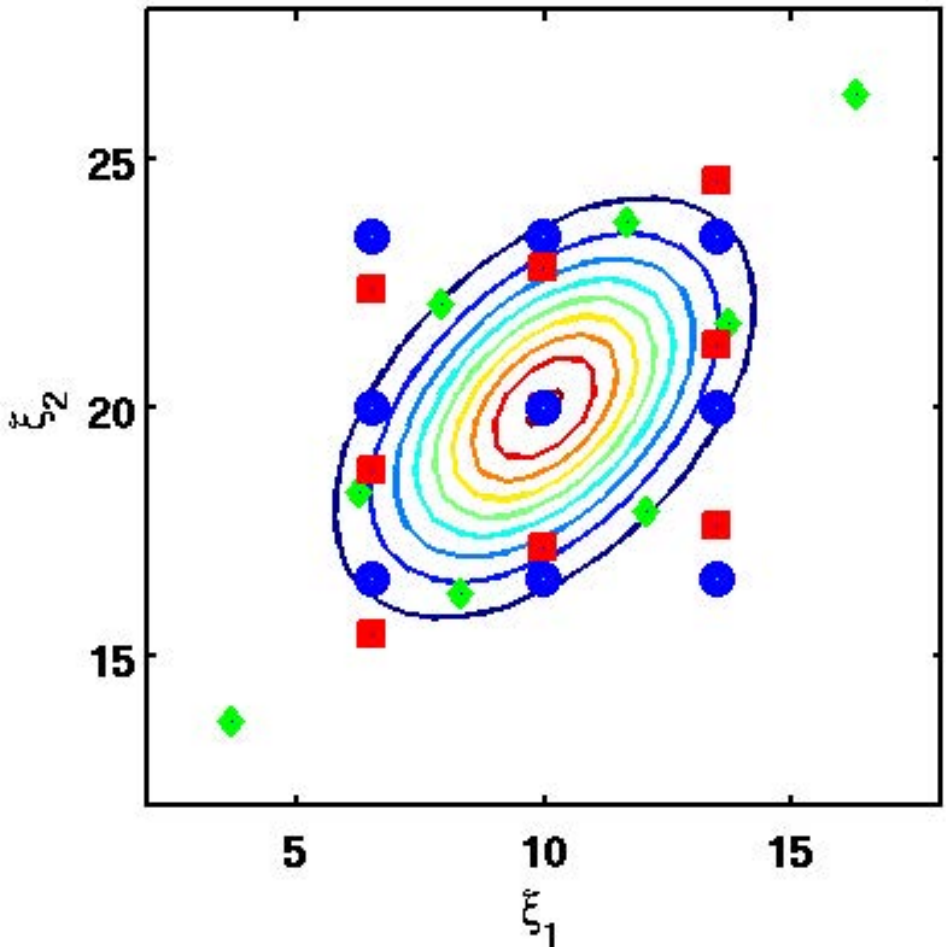
$\rho = 0.5$ $N = 4$; **BF-QMOM** (diamond), **TP-QMOM** (circle) and **CQMOM** (square)

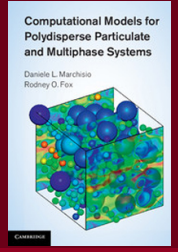




Comparison of Different Algorithms

$\rho = 0.5$ $N = 9$; **BF-QMOM** (diamond), **TP-QMOM** (circle) and **CQMOM** (square)





Hyperbolic: What Is It? Why?

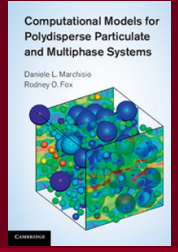
- Particles have real, finite velocities
- GPBE has real, finite velocities
- Moment system $\mathbf{M} = [M_0, M_1, M_2, M_3, M_4]$ from GPBE

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

  should be a hyberbolic equation (i.e. real eigenvalues)
- How do we check?: use Jacobian matrix of $\mathbf{F} = [M_1, M_2, M_3, M_4, M_5]$

$$\frac{\partial \mathbf{F}}{\partial \mathbf{M}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{\partial M_5}{\partial M_0} & \frac{\partial M_5}{\partial M_1} & \frac{\partial M_5}{\partial M_2} & \frac{\partial M_5}{\partial M_3} & \frac{\partial M_5}{\partial M_4} \end{bmatrix}$$

  Five eigenvalues must be real and distinct
- For HyQMOM, choose M_5 such that system is hyperbolic



Hyperbolic: What Is It? Why?

- Particles have real, finite velocities
- GPBE has real, finite velocities
- Moment system $\mathbf{M} = [M_0, M_1, M_2, M_3, M_4]$ from GPBE

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

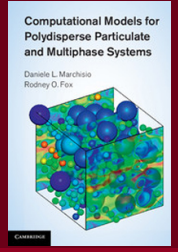
should be a hyberbolic equation (i.e. real eigenvalues)

- How do we check?: use Jacobian matrix of $\mathbf{F} = [M_1, M_2, M_3, M_4, M_5]$

$$\frac{\partial \mathbf{F}}{\partial \mathbf{M}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{\partial M_5}{\partial M_0} & \frac{\partial M_5}{\partial M_1} & \frac{\partial M_5}{\partial M_2} & \frac{\partial M_5}{\partial M_3} & \frac{\partial M_5}{\partial M_4} \end{bmatrix}$$

Five eigenvalues must be real and distinct

- For HyQMOM, choose M_5 such that system is hyperbolic



Hyperbolic: What Is It? Why?

- Particles have real, finite velocities
- GPBE has real, finite velocities
- Moment system $\mathbf{M} = [M_0, M_1, M_2, M_3, M_4]$ from GPBE

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

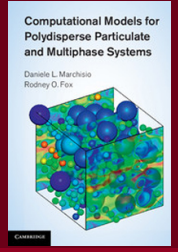
should be a hyberbolic equation (i.e. real eigenvalues)

- How do we check?: use Jacobian matrix of $\mathbf{F} = [M_1, M_2, M_3, M_4, M_5]$

$$\frac{\partial \mathbf{F}}{\partial \mathbf{M}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{\partial M_5}{\partial M_0} & \frac{\partial M_5}{\partial M_1} & \frac{\partial M_5}{\partial M_2} & \frac{\partial M_5}{\partial M_3} & \frac{\partial M_5}{\partial M_4} \end{bmatrix}$$

Five eigenvalues must be real and distinct

- For HyQMOM, choose M_5 such that system is hyperbolic



Hyperbolic: What Is It? Why?

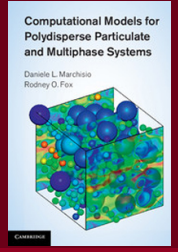
- Particles have real, finite velocities
- GPBE has real, finite velocities
- Moment system $\mathbf{M} = [M_0, M_1, M_2, M_3, M_4]$ from GPBE

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

  should be a hyberbolic equation (i.e. real eigenvalues)
- How do we check?: use Jacobian matrix of $\mathbf{F} = [M_1, M_2, M_3, M_4, M_5]$

$$\frac{\partial \mathbf{F}}{\partial \mathbf{M}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{\partial M_5}{\partial M_0} & \frac{\partial M_5}{\partial M_1} & \frac{\partial M_5}{\partial M_2} & \frac{\partial M_5}{\partial M_3} & \frac{\partial M_5}{\partial M_4} \end{bmatrix}$$

  Five eigenvalues must be real and distinct
- For HyQMOM, choose M_5 such that system is hyperbolic



Hyperbolic: What Is It? Why?

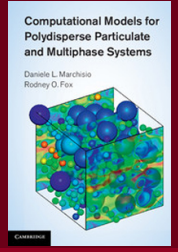
- Particles have real, finite velocities
- GPBE has real, finite velocities
- Moment system $\mathbf{M} = [M_0, M_1, M_2, M_3, M_4]$ from GPBE

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

  should be a hyberbolic equation (i.e. real eigenvalues)
- How do we check?: use Jacobian matrix of $\mathbf{F} = [M_1, M_2, M_3, M_4, M_5]$

$$\frac{\partial \mathbf{F}}{\partial \mathbf{M}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{\partial M_5}{\partial M_0} & \frac{\partial M_5}{\partial M_1} & \frac{\partial M_5}{\partial M_2} & \frac{\partial M_5}{\partial M_3} & \frac{\partial M_5}{\partial M_4} \end{bmatrix}$$

  Five eigenvalues must be real and distinct
- For HyQMOM, choose M_5 such that system is hyperbolic



Hyperbolic QMOM

- Approximate velocity NDF by $(N \geq 2)$

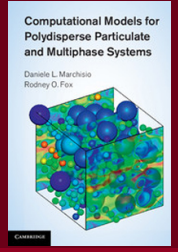
$$n(u) \approx M_0 \sum_{\alpha=1}^{N} p_\alpha \delta(u - \bar{u} - u_\alpha)$$

where $\bar{u} = M_1/M_0$, and p_α and u_α are found from central moments:

$$C_i = \frac{1}{M_0} \int_{-\infty}^{+\infty} (u - \bar{u})^i n(u) \, \mathrm{d}u$$

- Fix C_{2N-1} such that moment system is hyperbolic
- Given $\{1, 0, C_2 \ldots, C_{2N-2}\}$ and constraint on C_{2N-1}, apply QMOM

$$\{1, 0, C_2 \ldots, C_{2N-1}\} \overset{\text{QMOM}}{\Longrightarrow} \{p_1, p_2, \ldots, p_N\}, \{u_1, u_2, \ldots, u_N\}$$



Hyperbolic QMOM

- Approximate velocity NDF by $(N \geq 2)$

$$n(u) \approx M_0 \sum_{\alpha=1}^{N} p_\alpha \delta(u - \bar{u} - u_\alpha)$$

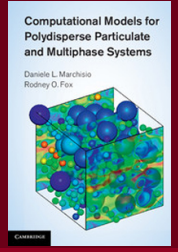
where $\bar{u} = M_1/M_0$, and p_α and u_α are found from central moments:

$$C_i = \frac{1}{M_0} \int_{-\infty}^{+\infty} (u - \bar{u})^i n(u) \, \mathrm{d}u$$

- Fix C_{2N-1} such that moment system is hyperbolic

- Given $\{1, 0, C_2 \ldots, C_{2N-2}\}$ and constraint on C_{2N-1}, apply QMOM

$$\{1, 0, C_2 \ldots, C_{2N-1}\} \overset{\text{QMOM}}{\Longrightarrow} \{p_1, p_2, \ldots, p_N\}, \{u_1, u_2, \ldots, u_N\}$$



Hyperbolic QMOM

- Approximate velocity NDF by $(N \geq 2)$

$$n(u) \approx M_0 \sum_{\alpha=1}^{N} p_\alpha \delta(u - \bar{u} - u_\alpha)$$

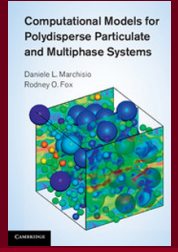
where $\bar{u} = M_1/M_0$, and p_α and u_α are found from central moments:

$$C_i = \frac{1}{M_0} \int_{-\infty}^{+\infty} (u - \bar{u})^i n(u) \, \mathrm{d}u$$

- Fix C_{2N-1} such that moment system is hyperbolic

- Given $\{1, 0, C_2 \ldots, C_{2N-2}\}$ and constraint on C_{2N-1}, apply QMOM

$$\{1, 0, C_2 \ldots, C_{2N-1}\} \overset{\text{QMOM}}{\Longrightarrow} \{p_1, p_2, \ldots, p_N\}, \{u_1, u_2, \ldots, u_N\}$$



Hyperbolic QMOM

- Moment closure for kinetic flux in 1-D:

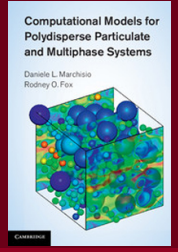
$$\partial_t M_{2N-2} + \partial_x \bar{M}_{2N-1} = 0 \quad \Longrightarrow \quad \bar{M}_{2N-1} = M_0 \sum_{\alpha=1}^{N} p_\alpha (\bar{u} + u_\alpha)^{2N-1}$$

For $N = 3$: $\bar{M}_5 = M_0 \left[p_1 (\bar{u} + u_1)^5 + p_2 \bar{u}^5 + p_3 (\bar{u} + u_3)^5 \right]$

- Theorem: Moment system for $\{M_0, M_1, M_2, M_3, M_4\}$ with kinetic flux $\bar{M}_5$ is hyperbolic with 5 distinct eigenvalues

$$\lambda_0 = \bar{u}, \; \lambda_{1,2,3,4} = \bar{u} + \frac{\sqrt{C_2}}{2} \left(q \pm \sqrt{4\eta - 3q^2 \pm 4\sqrt{(\eta - q^2)(\eta - q^2 - 1)}} \right)$$

where $q = C_3 / C_2^{3/2}$ and $\eta = C_4 / C_2^2$



Hyperbolic QMOM

- Moment closure for kinetic flux in 1-D:

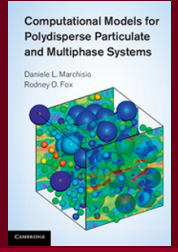
$$\partial_t M_{2N-2} + \partial_x \bar{M}_{2N-1} = 0 \quad \Longrightarrow \quad \bar{M}_{2N-1} = M_0 \sum_{\alpha=1}^{N} p_\alpha (\bar{u} + u_\alpha)^{2N-1}$$

For $N = 3$: $\bar{M}_5 = M_0 \left[p_1 (\bar{u} + u_1)^5 + p_2 \bar{u}^5 + p_3 (\bar{u} + u_3)^5 \right]$

- Theorem: Moment system for $\{M_0, M_1, M_2, M_3, M_4\}$ with kinetic flux $\bar{M}_5$ is hyperbolic with 5 distinct eigenvalues

$$\lambda_0 = \bar{u}, \; \lambda_{1,2,3,4} = \bar{u} + \frac{\sqrt{C_2}}{2} \left(q \pm \sqrt{4\eta - 3q^2 \pm 4\sqrt{(\eta - q^2)(\eta - q^2 - 1)}} \right)$$

where $q = C_3 / C_2^{3/2}$ and $\eta = C_4 / C_2^2$



Hyperbolic QMOM: Kinetic-Based Flux

- Kinetic-based flux in 1-D for $\mathbf{M} = \{M_0, M_1, M_2, M_3, M_4\}$:

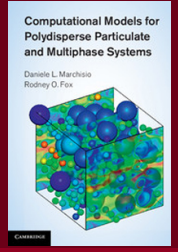
$$\partial_t \mathbf{M} + \partial_x \mathbf{F}(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} w_\alpha \lambda_\alpha^{n+1}$$

where $\mathbf{F} = \{F_0, F_1, F_2, F_3, F_4\}$

- 5 eigenvalues are known, weights found from

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\ \lambda_0^2 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \lambda_4^2 \\ \lambda_0^3 & \lambda_1^3 & \lambda_2^3 & \lambda_3^3 & \lambda_4^3 \\ \lambda_0^4 & \lambda_1^4 & \lambda_2^4 & \lambda_3^4 & \lambda_4^4 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}$$

and find that $w_0 = 0 \Longrightarrow$ system has 4 velocities



Hyperbolic QMOM: Kinetic-Based Flux

- Kinetic-based flux in 1-D for $\mathbf{M} = \{M_0, M_1, M_2, M_3, M_4\}$:

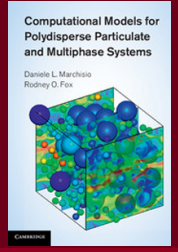
$$\partial_t \mathbf{M} + \partial_x \mathbf{F}(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} w_\alpha \lambda_\alpha^{n+1}$$

where $\mathbf{F} = \{F_0, F_1, F_2, F_3, F_4\}$

- 5 eigenvalues are known, weights found from

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\ \lambda_0^2 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \lambda_4^2 \\ \lambda_0^3 & \lambda_1^3 & \lambda_2^3 & \lambda_3^3 & \lambda_4^3 \\ \lambda_0^4 & \lambda_1^4 & \lambda_2^4 & \lambda_3^4 & \lambda_4^4 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}$$

and find that $w_0 = 0 \Longrightarrow$ system has 4 velocities



Hyperbolic QMOM: Kinetic-Based Flux

- Define normalized eigenvalues $\lambda_\alpha = \bar{u} + \sqrt{C_2}\,\mu_\alpha$ and normalized weights:

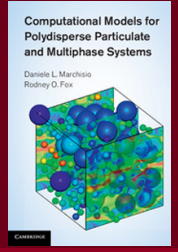
$$\begin{bmatrix} \mu_1 & \mu_2 & \mu_3 & \mu_4 \\ \mu_1^2 & \mu_2^2 & \mu_3^2 & \mu_4^2 \\ \mu_1^3 & \mu_2^3 & \mu_3^3 & \mu_4^3 \\ \mu_1^4 & \mu_2^4 & \mu_3^4 & \mu_4^4 \end{bmatrix} \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ q \\ \eta \end{bmatrix}$$

$\Longrightarrow$ positive weights depend on q and η

- Up/Down-wind flux splitting:

$$F_n = M_0 \sum_{\alpha=1}^{4} w'_\alpha \lambda_\alpha^n \min(0, \lambda_\alpha) + M_0 \sum_{\alpha=1}^{4} w'_\alpha \lambda_\alpha^n \max(0, \lambda_\alpha)$$

is used in finite-volume solver



Hyperbolic QMOM: Kinetic-Based Flux

- Define normalized eigenvalues $\lambda_\alpha = \bar{u} + \sqrt{C_2}\,\mu_\alpha$ and normalized weights:

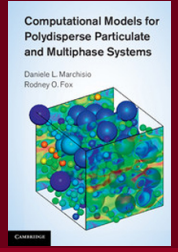
$$\begin{bmatrix} \mu_1 & \mu_2 & \mu_3 & \mu_4 \\ \mu_1^2 & \mu_2^2 & \mu_3^2 & \mu_4^2 \\ \mu_1^3 & \mu_2^3 & \mu_3^3 & \mu_4^3 \\ \mu_1^4 & \mu_2^4 & \mu_3^4 & \mu_4^4 \end{bmatrix} \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ q \\ \eta \end{bmatrix}$$

$\Longrightarrow$ positive weights depend on q and η

- Up/Down-wind flux splitting:

$$F_n = M_0 \sum_{\alpha=1}^{4} w'_\alpha \lambda_\alpha^n \min(0, \lambda_\alpha) + M_0 \sum_{\alpha=1}^{4} w'_\alpha \lambda_\alpha^n \max(0, \lambda_\alpha)$$

is used in finite-volume solver



Conditional HyQMOM for 2-D Phase Space

- Approximate 2-D velocity NDF by ($N \geq 2$)

$$n(u,v) \approx M_{0,0} \sum_{\alpha=1}^{N} \sum_{\beta=1}^{N} p_\alpha p_{\alpha\beta} \delta(u - \bar{u} - u_\alpha) \delta(v - \bar{v} - \bar{v}_\alpha - v_{\alpha\beta})$$

where $\bar{v} = M_{0,1}/M_{0,0}$ and abscissas $v_{\alpha\beta}$ and $\bar{v}_\alpha$ are found from central moments:

$$C_{i,j} = \frac{1}{M_{0,0}} \int_{\mathbb{R}^2} (u - \bar{u})^i (v - \bar{v})^j n(u,v) \, \mathrm{d}u \, \mathrm{d}v$$

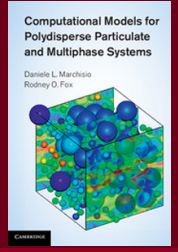
- Example $N^2 = 9$ nodes, CQMOM applied to symmetric moment sets:

10 moments

$M_{0,0}$	$M_{1,0}$	$M_{2,0}$	$M_{3,0}$	$M_{4,0}$
$M_{0,1}$	$M_{1,1}$			
$M_{0,2}$				
$M_{0,3}$				
$M_{0,4}$				

12 moments

$M_{0,0}$	$M_{1,0}$	$M_{2,0}$	$M_{3,0}$	$M_{4,0}$
$M_{0,1}$	$M_{1,1}$	$M_{2,1}$		
$M_{0,2}$	$M_{1,2}$			
$M_{0,3}$				
$M_{0,4}$				



Conditional HyQMOM for 2-D Phase Space

- Approximate 2-D velocity NDF by ($N \geq 2$)

$$n(u,v) \approx M_{0,0} \sum_{\alpha=1}^{N} \sum_{\beta=1}^{N} p_\alpha p_{\alpha\beta} \delta(u - \bar{u} - u_\alpha) \delta(v - \bar{v} - \bar{v}_\alpha - v_{\alpha\beta})$$

where $\bar{v} = M_{0,1}/M_{0,0}$ and abscissas $v_{\alpha\beta}$ and $\bar{v}_\alpha$ are found from central moments:

$$C_{i,j} = \frac{1}{M_{0,0}} \int_{\mathbb{R}^2} (u - \bar{u})^i (v - \bar{v})^j n(u,v) \, \mathrm{d}u \, \mathrm{d}v$$

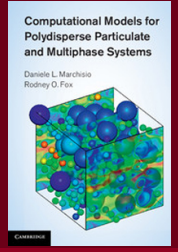
- Example $N^2 = 9$ nodes, CQMOM applied to symmetric moment sets:

10 moments

$M_{0,0}$	$M_{1,0}$	$M_{2,0}$	$M_{3,0}$	$M_{4,0}$
$M_{0,1}$	$M_{1,1}$			
$M_{0,2}$				
$M_{0,3}$				
$M_{0,4}$				

12 moments

$M_{0,0}$	$M_{1,0}$	$M_{2,0}$	$M_{3,0}$	$M_{4,0}$
$M_{0,1}$	$M_{1,1}$	$M_{2,1}$		
$M_{0,2}$	$M_{1,2}$			
$M_{0,3}$				
$M_{0,4}$				



Conditional HyQMOM for 2-D Phase Space

- With 10 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha$ is found from

$$\sum_{\alpha=1}^{3} p_\alpha \bar{v}_\alpha = C_{0,1} = 0 \qquad \sum_{\alpha=1}^{3} p_\alpha u_\alpha \bar{v}_\alpha = C_{1,1}$$

$\Longrightarrow a_0 = 0$ and $a_1 = C_{1,1}/C_{2,0}$ captures correlation between u and v

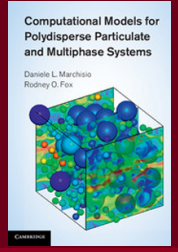
- CHyQMOM conditional moments are $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ where

$$C_{2|u_\alpha} = C_{0,2} - a_1^2 C_{2,0} \quad C_{3|u_\alpha} = C_{0,3} - a_1^3 C_{3,0} \quad C_{4|u_\alpha} = C_{0,4} - 6a_1^2 C_{2,0} C_{2|u_\alpha} - a_1^4 C_{4,0}$$

N.B. conditional moments do not depend on α

- HyQMOM applied to $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ to find $p_{\alpha\beta}$ and $v_{\alpha\beta}$

- With 12 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha + a_2 u_\alpha^2$ and $C_{2|u_\alpha} = b_0 + b_1 u_\alpha$



Conditional HyQMOM for 2-D Phase Space

- With 10 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha$ is found from

$$\sum_{\alpha=1}^{3} p_\alpha \bar{v}_\alpha = C_{0,1} = 0 \qquad \sum_{\alpha=1}^{3} p_\alpha u_\alpha \bar{v}_\alpha = C_{1,1}$$

$\Longrightarrow a_0 = 0$ and $a_1 = C_{1,1}/C_{2,0}$ captures correlation between u and v

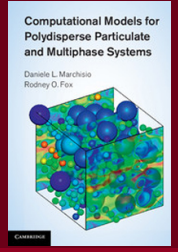
- CHyQMOM conditional moments are $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ where

$$C_{2|u_\alpha} = C_{0,2} - a_1^2 C_{2,0} \quad C_{3|u_\alpha} = C_{0,3} - a_1^3 C_{3,0} \quad C_{4|u_\alpha} = C_{0,4} - 6a_1^2 C_{2,0} C_{2|u_\alpha} - a_1^4 C_{4,0}$$

N.B. conditional moments do not depend on α

- HyQMOM applied to $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ to find $p_{\alpha\beta}$ and $v_{\alpha\beta}$

- With 12 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha + a_2 u_\alpha^2$ and $C_{2|u_\alpha} = b_0 + b_1 u_\alpha$



Conditional HyQMOM for 2-D Phase Space

- With 10 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha$ is found from

$$\sum_{\alpha=1}^{3} p_\alpha \bar{v}_\alpha = C_{0,1} = 0 \qquad \sum_{\alpha=1}^{3} p_\alpha u_\alpha \bar{v}_\alpha = C_{1,1}$$

$\Longrightarrow a_0 = 0$ and $a_1 = C_{1,1}/C_{2,0}$ captures correlation between u and v

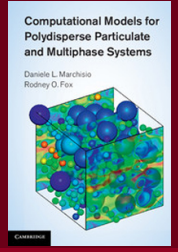
- CHyQMOM conditional moments are $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ where

$$C_{2|u_\alpha} = C_{0,2} - a_1^2 C_{2,0} \quad C_{3|u_\alpha} = C_{0,3} - a_1^3 C_{3,0} \quad C_{4|u_\alpha} = C_{0,4} - 6a_1^2 C_{2,0} C_{2|u_\alpha} - a_1^4 C_{4,0}$$

N.B. conditional moments do not depend on α

- HyQMOM applied to $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ to find $p_{\alpha\beta}$ and $v_{\alpha\beta}$

- With 12 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha + a_2 u_\alpha^2$ and $C_{2|u_\alpha} = b_0 + b_1 u_\alpha$



Conditional HyQMOM for 2-D Phase Space

- With 10 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha$ is found from

$$\sum_{\alpha=1}^{3} p_\alpha \bar{v}_\alpha = C_{0,1} = 0 \qquad \sum_{\alpha=1}^{3} p_\alpha u_\alpha \bar{v}_\alpha = C_{1,1}$$

$\Longrightarrow a_0 = 0$ and $a_1 = C_{1,1}/C_{2,0}$ captures correlation between u and v

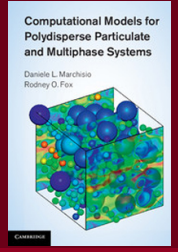
- CHyQMOM conditional moments are $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ where

$$C_{2|u_\alpha} = C_{0,2} - a_1^2 C_{2,0} \quad C_{3|u_\alpha} = C_{0,3} - a_1^3 C_{3,0} \quad C_{4|u_\alpha} = C_{0,4} - 6a_1^2 C_{2,0} C_{2|u_\alpha} - a_1^4 C_{4,0}$$

N.B. conditional moments do not depend on α

- HyQMOM applied to $\{1, 0, C_{2|u_\alpha}, C_{3|u_\alpha}, C_{4|u_\alpha}\}$ to find $p_{\alpha\beta}$ and $v_{\alpha\beta}$

- With 12 moments, $\bar{v}_\alpha = a_0 + a_1 u_\alpha + a_2 u_\alpha^2$ and $C_{2|u_\alpha} = b_0 + b_1 u_\alpha$



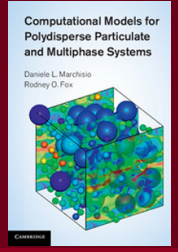
Conditional HyQMOM for 2-D Phase Space

- CHyQMOM closure for kinetic flux in 1-D spatial and 2-D velocity with 10 moments:

$$\partial_t M_{1,1} + \partial_x \bar{M}_{2,1} = 0 \quad \Longrightarrow \quad \bar{M}_{2,1} = M_{0,0} \sum_{\alpha=1}^{3} p_\alpha (\bar{u} + u_\alpha)^2 (\bar{v} + \bar{v}_\alpha)$$

- Theorem: 10 moment system with kinetic flux is hyperbolic with 10 distinct eigenvectors

- Extension to 3-D phase space follows same logic: Example $N^3 = 27$ nodes uses either 16 or 23 moments



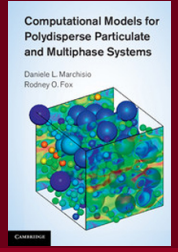
Conditional HyQMOM for 2-D Phase Space

- CHyQMOM closure for kinetic flux in 1-D spatial and 2-D velocity with 10 moments:

$$\partial_t M_{1,1} + \partial_x \bar{M}_{2,1} = 0 \quad \Longrightarrow \quad \bar{M}_{2,1} = M_{0,0} \sum_{\alpha=1}^{3} p_\alpha (\bar{u} + u_\alpha)^2 (\bar{v} + \bar{v}_\alpha)$$

- Theorem: 10 moment system with kinetic flux is hyperbolic with 10 distinct eigenvectors

- Extension to 3-D phase space follows same logic: Example $N^3 = 27$ nodes uses either 16 or 23 moments



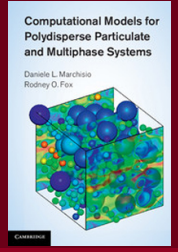
Conditional HyQMOM for 2-D Phase Space

- CHyQMOM closure for kinetic flux in 1-D spatial and 2-D velocity with 10 moments:

$$\partial_t M_{1,1} + \partial_x \bar{M}_{2,1} = 0 \quad \Longrightarrow \quad \bar{M}_{2,1} = M_{0,0} \sum_{\alpha=1}^{3} p_\alpha (\bar{u} + u_\alpha)^2 (\bar{v} + \bar{v}_\alpha)$$

- Theorem: 10 moment system with kinetic flux is hyperbolic with 10 distinct eigenvectors

- Extension to 3-D phase space follows same logic: Example $N^3 = 27$ nodes uses either 16 or 23 moments



Hyperbolic QMOM: Kinetic-Based Flux 2/3-D

- Kinetic-based flux in 2-D for 10 moments $\mathbf{M} = \{M_{0,0}, M_{1,0}, M_{0,1}, \ldots, M_{4,0}, M_{0,4}\}$:

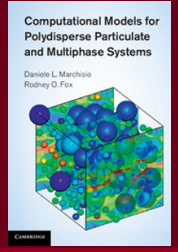
$$\partial_t \mathbf{M} + \partial_x \mathbf{F}_x(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

where the 10 flux components are $\mathbf{F}_x = \{F_{0,0}, F_{1,0}, F_{0,1}, \ldots, F_{4,0}, F_{0,4}\}$

- 5 eigenvalues λ_α are same as 1-D case
- Conditional moments $M_{n|\lambda_\alpha}$ found using CQMOM:

$$M_n = \sum_{\alpha=1}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

for $n = (0,0), (1,0), (0,1), \ldots, (4,0), (0,4)$



Hyperbolic QMOM: Kinetic-Based Flux 2/3-D

- Kinetic-based flux in 2-D for 10 moments $\mathbf{M} = \{M_{0,0}, M_{1,0}, M_{0,1}, \ldots, M_{4,0}, M_{0,4}\}$:

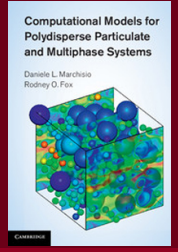
$$\partial_t \mathbf{M} + \partial_x \mathbf{F}_x(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

  where the 10 flux components are $\mathbf{F}_x = \{F_{0,0}, F_{1,0}, F_{0,1}, \ldots, F_{4,0}, F_{0,4}\}$

- 5 eigenvalues λ_α are same as 1-D case
- Conditional moments $M_{n|\lambda_\alpha}$ found using CQMOM:

$$M_n = \sum_{\alpha=1}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

  for $n = (0,0), (1,0), (0,1), \ldots, (4,0), (0,4)$



Hyperbolic QMOM: Kinetic-Based Flux 2/3-D

- Kinetic-based flux in 2-D for 10 moments $\mathbf{M} = \{M_{0,0}, M_{1,0}, M_{0,1}, \ldots, M_{4,0}, M_{0,4}\}$:

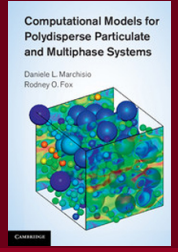
$$\partial_t \mathbf{M} + \partial_x \mathbf{F}_x(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

  where the 10 flux components are $\mathbf{F}_x = \{F_{0,0}, F_{1,0}, F_{0,1}, \ldots, F_{4,0}, F_{0,4}\}$

- 5 eigenvalues λ_α are same as 1-D case
- Conditional moments $M_{n|\lambda_\alpha}$ found using CQMOM:

$$M_n = \sum_{\alpha=1}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

for $n = (0,0), (1,0), (0,1), \ldots, (4,0), (0,4)$



Hyperbolic QMOM: Kinetic-Based Flux 2/3-D

- Kinetic-based flux for y direction:

$$\partial_t \mathbf{M} + \partial_y \mathbf{F}_y(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

  is done in same way, except 5 eigenvalues are found using moments $\{M_{0,0}, M_{0,1}, M_{0,2}, M_{0,3}, M_{0,4}\}$

- Up/Down-wind flux splitting:

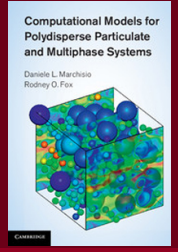
$$F_n = \sum_{\alpha=1}^{4} M_{n|\lambda_\alpha} \min(0, \lambda_\alpha) + \sum_{\alpha=1}^{4} M_{n|\lambda_\alpha} \max(0, \lambda_\alpha)$$

  is used in finite-volume solver

- In 3-D, definitions are analogous

$$\partial_t \mathbf{M} + \partial_z \mathbf{F}_z(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

for $n = (0,0,0), (1,0,0), (0,1,0), (0,0,1), \ldots, (4,0,0), (0,4,0), (0,0,4)$



Hyperbolic QMOM: Kinetic-Based Flux 2/3-D

- Kinetic-based flux for y direction:

$$\partial_t \mathbf{M} + \partial_y \mathbf{F}_y(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

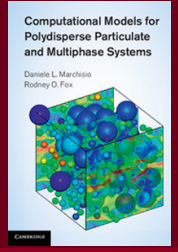
  is done in same way, except 5 eigenvalues are found using moments $\{M_{0,0}, M_{0,1}, M_{0,2}, M_{0,3}, M_{0,4}\}$
- Up/Down-wind flux splitting:

$$F_n = \sum_{\alpha=1}^{4} M_{n|\lambda_\alpha} \min(0, \lambda_\alpha) + \sum_{\alpha=1}^{4} M_{n|\lambda_\alpha} \max(0, \lambda_\alpha)$$

  is used in finite-volume solver
- In 3-D, definitions are analogous

$$\partial_t \mathbf{M} + \partial_z \mathbf{F}_z(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

for $n = (0,0,0), (1,0,0), (0,1,0), (0,0,1), \ldots, (4,0,0), (0,4,0), (0,0,4)$



Hyperbolic QMOM: Kinetic-Based Flux 2/3-D

- Kinetic-based flux for y direction:

$$\partial_t \mathbf{M} + \partial_y \mathbf{F}_y(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

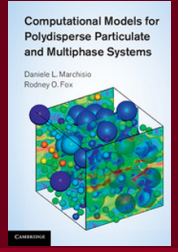
  is done in same way, except 5 eigenvalues are found using moments $\{M_{0,0}, M_{0,1}, M_{0,2}, M_{0,3}, M_{0,4}\}$
- Up/Down-wind flux splitting:

$$F_n = \sum_{\alpha=1}^{4} M_{n|\lambda_\alpha} \min(0, \lambda_\alpha) + \sum_{\alpha=1}^{4} M_{n|\lambda_\alpha} \max(0, \lambda_\alpha)$$

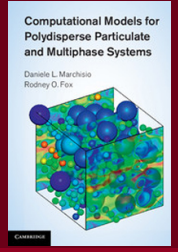
  is used in finite-volume solver
- In 3-D, definitions are analogous

$$\partial_t \mathbf{M} + \partial_z \mathbf{F}_z(\mathbf{M}) = 0 \quad \Longrightarrow \quad F_n = \sum_{\alpha=0}^{4} \lambda_\alpha M_{n|\lambda_\alpha}$$

  for $n = (0,0,0), (1,0,0), (0,1,0), (0,0,1), \ldots, (4,0,0), (0,4,0), (0,0,4)$



HIGH-ORDER, REALIZABLE, KINETIC-BASED, FINITE-VOLUME METHODS



Kinetic-Based Finite-Volume Methods

Given a set of **transported moments**, solve

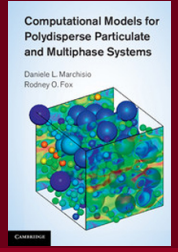
$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = k \int v^{k-1} \xi^l An \, \mathrm{d}v \, \mathrm{d}\xi + l \int v^k \xi^{l-1} Gn \, \mathrm{d}v \, \mathrm{d}\xi + \int v^k \xi^l \mathbb{C} \, \mathrm{d}v \, \mathrm{d}\xi$$

where RHS is closed using QBMM:

$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = \sum_{\alpha=1}^{N} n_\alpha \left\{ k v_\alpha^{k-1} \xi_\alpha^l A_\alpha + l v_\alpha^k \xi_\alpha^{l-1} G_\alpha + v_\alpha^k \xi_\alpha^l \mathbb{C}_\alpha \right\}$$

Things to consider:

- How do we discretize the spatial fluxes?
- How do we update the moments in time?
- How can we ensure that the moments are always **realizable**?



Kinetic-Based Finite-Volume Methods

Given a set of **transported moments**, solve

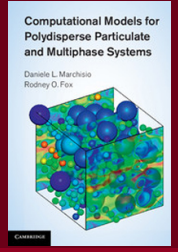
$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = k \int v^{k-1} \xi^l An \, \mathrm{d}v \, \mathrm{d}\xi + l \int v^k \xi^{l-1} Gn \, \mathrm{d}v \, \mathrm{d}\xi + \int v^k \xi^l \mathbb{C} \, \mathrm{d}v \, \mathrm{d}\xi$$

where RHS is closed using QBMM:

$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = \sum_{\alpha=1}^{\mathcal{N}} n_\alpha \left\{ k v_\alpha^{k-1} \xi_\alpha^l A_\alpha + l v_\alpha^k \xi_\alpha^{l-1} G_\alpha + v_\alpha^k \xi_\alpha^l \mathbb{C}_\alpha \right\}$$

Things to consider:

- How do we discretize the spatial fluxes?
- How do we update the moments in time?
- How can we ensure that the moments are always **realizable**?



Kinetic-Based Finite-Volume Methods

Given a set of **transported moments**, solve

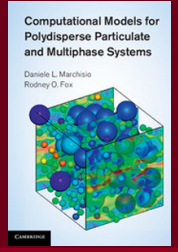
$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = k \int v^{k-1} \xi^l A n \, \mathrm{d}v \, \mathrm{d}\xi + l \int v^k \xi^{l-1} G n \, \mathrm{d}v \, \mathrm{d}\xi + \int v^k \xi^l \mathbb{C} \, \mathrm{d}v \, \mathrm{d}\xi$$

where RHS is closed using QBMM:

$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = \sum_{\alpha=1}^{\mathcal{N}} n_\alpha \left\{ k v_\alpha^{k-1} \xi_\alpha^l A_\alpha + l v_\alpha^k \xi_\alpha^{l-1} G_\alpha + v_\alpha^k \xi_\alpha^l \mathbb{C}_\alpha \right\}$$

Things to consider:

- How do we discretize the spatial fluxes?
- How do we update the moments in time?
- How can we ensure that the moments are always **realizable**?



Kinetic-Based Finite-Volume Methods

Given a set of **transported moments**, solve

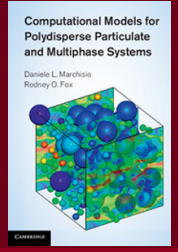
$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = k \int v^{k-1} \xi^l A n \, \mathrm{d}v \, \mathrm{d}\xi + l \int v^k \xi^{l-1} G n \, \mathrm{d}v \, \mathrm{d}\xi + \int v^k \xi^l \mathbb{C} \, \mathrm{d}v \, \mathrm{d}\xi$$

where RHS is closed using QBMM:

$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = \sum_{\alpha=1}^{\mathcal{N}} n_\alpha \left\{ k v_\alpha^{k-1} \xi_\alpha^l A_\alpha + l v_\alpha^k \xi_\alpha^{l-1} G_\alpha + v_\alpha^k \xi_\alpha^l \mathbb{C}_\alpha \right\}$$

Things to consider:

- How do we discretize the spatial fluxes?
- How do we update the moments in time?
- How can we ensure that the moments are always **realizable**?



Kinetic-Based Finite-Volume Methods

Given a set of **transported moments**, solve

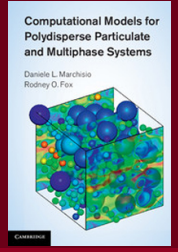
$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = k \int v^{k-1} \xi^l An \, \mathrm{d}v \, \mathrm{d}\xi + l \int v^k \xi^{l-1} Gn \, \mathrm{d}v \, \mathrm{d}\xi + \int v^k \xi^l \mathbb{C} \, \mathrm{d}v \, \mathrm{d}\xi$$

where RHS is closed using QBMM:

$$\frac{\partial M_{kl}}{\partial t} + \frac{\partial M_{k+1l}}{\partial x} = \sum_{\alpha=1}^{\mathcal{N}} n_\alpha \left\{ k v_\alpha^{k-1} \xi_\alpha^l A_\alpha + l v_\alpha^k \xi_\alpha^{l-1} G_\alpha + v_\alpha^k \xi_\alpha^l \mathbb{C}_\alpha \right\}$$

Things to consider:

- How do we discretize the spatial fluxes?
- How do we update the moments in time?
- How can we ensure that the moments are always **realizable**?



Kinetic-Based Spatial Fluxes in 1-D

Spatial fluxes can use kinetic formulation: e.g. $\partial_t M_0 + \partial_x M_1 = 0$

$$M_1 = Q_1^- + Q_1^+$$
$$= \int_{-\infty}^{0} u\, n^*(u)\, \mathrm{d}u + \int_{0}^{\infty} u\, n^*(u)\, \mathrm{d}u$$

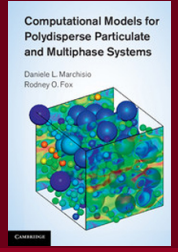
Using reconstructed n^*, downwind and upwind flux components are

$$Q_1^- = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(-\infty,0)}(u_\alpha) \qquad Q_1^+ = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(0,\infty)}(u_\alpha)$$

where $I_{\mathbb{S}}(x)$ is the indicator function for the interval $\mathbb{S}$

Kinetic-based fluxes are weakly hyperbolic with QMOM

Using hyperbolic QMOM, they are hyperbolic



Kinetic-Based Spatial Fluxes in 1-D

Spatial fluxes can use kinetic formulation: e.g. $\partial_t M_0 + \partial_x M_1 = 0$

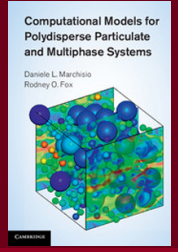
$$M_1 = Q_1^- + Q_1^+$$
$$= \int_{-\infty}^{0} u\, n^*(u)\, \mathrm{d}u + \int_{0}^{\infty} u\, n^*(u)\, \mathrm{d}u$$

Using reconstructed n^*, downwind and upwind flux components are

$$Q_1^- = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(-\infty,0)}(u_\alpha) \qquad Q_1^+ = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(0,\infty)}(u_\alpha)$$

where $I_{\mathbb{S}}(x)$ is the indicator function for the interval $\mathbb{S}$

Kinetic-based fluxes are weakly hyperbolic with QMOM
Using hyperbolic QMOM, they are hyperbolic



Kinetic-Based Spatial Fluxes in 1-D

Spatial fluxes can use kinetic formulation: e.g. $\partial_t M_0 + \partial_x M_1 = 0$

$$M_1 = Q_1^- + Q_1^+$$
$$= \int_{-\infty}^{0} u\, n^*(u)\, \mathrm{d}u + \int_{0}^{\infty} u\, n^*(u)\, \mathrm{d}u$$

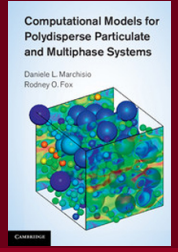
Using reconstructed n^*, downwind and upwind flux components are

$$Q_1^- = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(-\infty,0)}(u_\alpha) \qquad Q_1^+ = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(0,\infty)}(u_\alpha)$$

where $I_{\mathbb{S}}(x)$ is the indicator function for the interval $\mathbb{S}$

Kinetic-based fluxes are weakly hyperbolic with QMOM

Using hyperbolic QMOM, they are hyperbolic



Kinetic-Based Spatial Fluxes in 1-D

Spatial fluxes can use kinetic formulation: e.g. $\partial_t M_0 + \partial_x M_1 = 0$

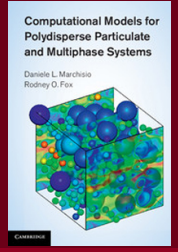
$$M_1 = Q_1^- + Q_1^+$$
$$= \int_{-\infty}^{0} u\, n^*(u)\, \mathrm{d}u + \int_{0}^{\infty} u\, n^*(u)\, \mathrm{d}u$$

Using reconstructed n^*, downwind and upwind flux components are

$$Q_1^- = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(-\infty,0)}(u_\alpha) \qquad Q_1^+ = \sum_{\alpha=1}^{N} n_\alpha u_\alpha I_{(0,\infty)}(u_\alpha)$$

where $I_{\mathbb{S}}(x)$ is the indicator function for the interval $\mathbb{S}$

Kinetic-based fluxes are weakly hyperbolic with QMOM
Using hyperbolic QMOM, they are hyperbolic



Finite-Volume Method: Definitions I

- 1-D advection problem:

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

where $\mathbf{M} = \int \mathbf{K}(v) n(v)\, \mathrm{d}v$ and $\mathbf{F}(\mathbf{M}) = \int v \mathbf{K}(v) n(v)\, \mathrm{d}v$

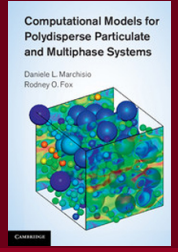
- Finite-volume representation of moment vector:

$$\mathbf{M}_i^n \equiv \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} \mathbf{M}(t_n, x)\, \mathrm{d}x$$

- Finite-volume formula: $\lambda = \Delta t / \Delta x$

$$\mathbf{M}_i^{n+1} = \mathbf{M}_i^n - \lambda \left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n \right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n \right) \right]$$

where $\mathbf{G}(\mathbf{M}_l, \mathbf{M}_r) = \int v^+ \mathbf{K}(v) n_l(v)\, \mathrm{d}v + \int v^- \mathbf{K}(v) n_r(v)\, \mathrm{d}v$



Finite-Volume Method: Definitions I

- 1-D advection problem:

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

where $\mathbf{M} = \int \mathbf{K}(v) n(v)\, \mathrm{d}v$ and $\mathbf{F}(\mathbf{M}) = \int v \mathbf{K}(v) n(v)\, \mathrm{d}v$

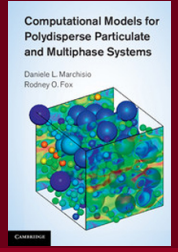
- Finite-volume representation of moment vector:

$$\mathbf{M}_i^n \equiv \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} \mathbf{M}(t_n, x)\, \mathrm{d}x$$

- Finite-volume formula: $\lambda = \Delta t / \Delta x$

$$\mathbf{M}_i^{n+1} = \mathbf{M}_i^n - \lambda \left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n \right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n \right) \right]$$

where $\mathbf{G}(\mathbf{M}_l, \mathbf{M}_r) = \int v^+ \mathbf{K}(v) n_l(v)\, \mathrm{d}v + \int v^- \mathbf{K}(v) n_r(v)\, \mathrm{d}v$



Finite-Volume Method: Definitions I

- 1-D advection problem:

$$\frac{\partial \mathbf{M}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{M})}{\partial x} = 0$$

where $\mathbf{M} = \int \mathbf{K}(v) n(v)\, \mathrm{d}v$ and $\mathbf{F}(\mathbf{M}) = \int v \mathbf{K}(v) n(v)\, \mathrm{d}v$

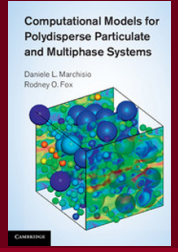
- Finite-volume representation of moment vector:

$$\mathbf{M}_i^n \equiv \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} \mathbf{M}(t_n, x)\, \mathrm{d}x$$

- Finite-volume formula: $\lambda = \Delta t / \Delta x$

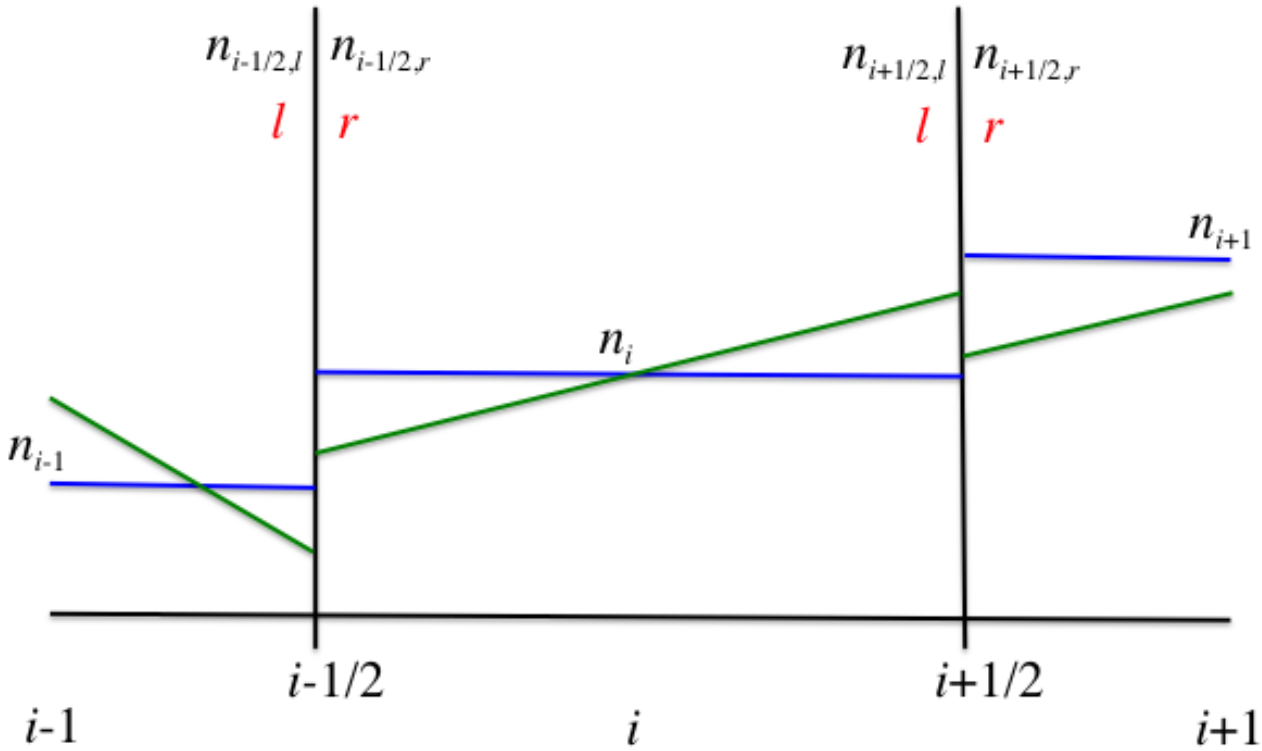
$$\mathbf{M}_i^{n+1} = \mathbf{M}_i^n - \lambda \left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n \right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n \right) \right]$$

where $\mathbf{G}(\mathbf{M}_l, \mathbf{M}_r) = \int v^+ \mathbf{K}(v) n_l(v)\, \mathrm{d}v + \int v^- \mathbf{K}(v) n_r(v)\, \mathrm{d}v$



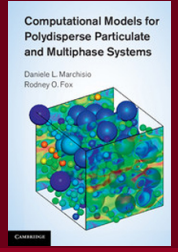
Finite-Volume Method: Definitions II

Reconstruction of $n(t_n, x, v)$ in each cell:



Blue: 1st order reconstruction

Green: 2nd order reconstruction



Realizability and Spatial Fluxes

- Flux functions: given $\mathbf{M}_i^n$ define $\mathbf{G}(\mathbf{M}_l, \mathbf{M}_r)$ to achieve high-order spatial accuracy but keep $\mathbf{M}_i^{n+1}$ realizable!
- Discrete distribution function: Define

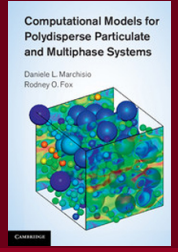
$$\mathbf{M}_i^{n+1} \equiv \int \mathbf{K}(v) h_i(v)\, \mathrm{d}v$$

and finite-volume formula can be written as

$$h_i(v) = \lambda |v^-| n^n_{i+\frac{1}{2},r} + \lambda v^+ n^n_{i-\frac{1}{2},l} + n_i^n - \lambda |v^-| n^n_{i-\frac{1}{2},r} - \lambda v^+ n^n_{i+\frac{1}{2},l}$$

(black part ≥ 0, red part can be negative)

- Sufficient condition for realizable moments: $h_i(v) \geq 0$ for all v and i



Realizability and Spatial Fluxes

- Flux functions: given $\mathbf{M}_i^n$ define $\mathbf{G}(\mathbf{M}_l, \mathbf{M}_r)$ to achieve high-order spatial accuracy but keep $\mathbf{M}_i^{n+1}$ realizable!
- Discrete distribution function: Define

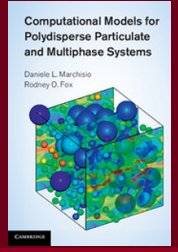
$$\mathbf{M}_i^{n+1} \equiv \int \mathbf{K}(v) h_i(v)\, \mathrm{d}v$$

and finite-volume formula can be written as

$$h_i(v) = \lambda |v^-| n^n_{i+\frac{1}{2},r} + \lambda v^+ n^n_{i-\frac{1}{2},l} + n_i^n - \lambda |v^-| n^n_{i-\frac{1}{2},r} - \lambda v^+ n^n_{i+\frac{1}{2},l}$$

(black part ≥ 0, red part can be negative)

- Sufficient condition for realizable moments: $h_i(v) \geq 0$ for all v and i



Realizability and Spatial Fluxes

- Flux functions: given $\mathbf{M}_i^n$ define $\mathbf{G}(\mathbf{M}_l, \mathbf{M}_r)$ to achieve high-order spatial accuracy but keep $\mathbf{M}_i^{n+1}$ realizable!
- Discrete distribution function: Define

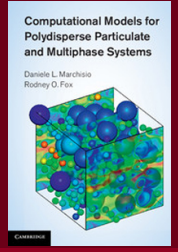
$$\mathbf{M}_i^{n+1} \equiv \int \mathbf{K}(v) h_i(v)\, \mathrm{d}v$$

and finite-volume formula can be written as

$$h_i(v) = \lambda |v^-| n^n_{i+\frac{1}{2},r} + \lambda v^+ n^n_{i-\frac{1}{2},l} + n_i^n - \lambda |v^-| n^n_{i-\frac{1}{2},r} - \lambda v^+ n^n_{i+\frac{1}{2},l}$$

(black part ≥ 0, red part can be negative)

- Sufficient condition for realizable moments: $h_i(v) \geq 0$ for all v and i



Realizable, High-Order, Spatial Fluxes

- First order: $n^n_{i-\frac{1}{2},r} = n^n_{i+\frac{1}{2},l} = n^n_i$ so that

  $h = \lambda|v^-|n^n_{i+1} + \lambda v^+ n^n_{i-1} + (1 - \lambda|v^-| - \lambda v^+)\, n^n_i \quad \Rightarrow \quad \frac{1}{|v^-|+v^+} \geq \lambda$

  Moments are realizable, but scheme is diffusive ...

- Quasi-higher order: Let $n^n_i = \sum_\alpha \rho^n_{\alpha,i}\delta(v - v^n_{\alpha,i})$ and define

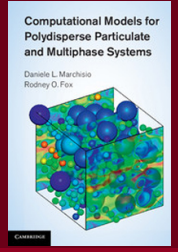
$$n^n_{i-\frac{1}{2},r} = \sum_\alpha \rho^n_{\alpha,i-\frac{1}{2},r}\,\delta(v - v^n_{\alpha,i})$$
$$n^n_{i+\frac{1}{2},l} = \sum_\alpha \rho^n_{\alpha,i+\frac{1}{2},l}\,\delta(v - v^n_{\alpha,i})$$

  so that

$$h = \lambda|v^-|n^n_{i+\frac{1}{2},r} + \lambda v^+ n^n_{i-\frac{1}{2},l} + \sum_\alpha \left(\rho^n_{\alpha,i} - \lambda|v^-|\rho^n_{\alpha,i-\frac{1}{2},r} - \lambda v^+\rho^n_{\alpha,i+\frac{1}{2},l}\right)\delta(v - v^n_{\alpha,i})$$

$$\Longrightarrow \quad \min_\alpha \left(\frac{\rho^n_{\alpha,i}}{|v^-_{\alpha,i}|\rho^n_{\alpha,i-\frac{1}{2},r} + v^+_{\alpha,i}\rho^n_{\alpha,i+\frac{1}{2},l}}\right) \geq \lambda$$

Use high-order, finite-volume schemes only for the weights



Realizable, High-Order, Spatial Fluxes

- First order: $n^n_{i-\frac{1}{2},r} = n^n_{i+\frac{1}{2},l} = n^n_i$ so that

  $h = \lambda|v^-|n^n_{i+1} + \lambda v^+ n^n_{i-1} + (1 - \lambda|v^-| - \lambda v^+)\, n^n_i \quad \Rightarrow \quad \frac{1}{|v^-|+v^+} \geq \lambda$

  Moments are realizable, but scheme is diffusive ...

- Quasi-higher order: Let $n^n_i = \sum_\alpha \rho^n_{\alpha,i}\delta(v - v^n_{\alpha,i})$ and define

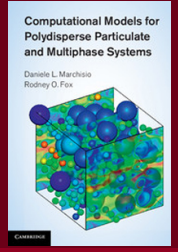
$$n^n_{i-\frac{1}{2},r} = \sum_\alpha \rho^n_{\alpha,i-\frac{1}{2},r}\,\delta(v - v^n_{\alpha,i})$$
$$n^n_{i+\frac{1}{2},l} = \sum_\alpha \rho^n_{\alpha,i+\frac{1}{2},l}\,\delta(v - v^n_{\alpha,i})$$

  so that

$$h = \lambda|v^-|n^n_{i+\frac{1}{2},r} + \lambda v^+ n^n_{i-\frac{1}{2},l} + \sum_\alpha \left(\rho^n_{\alpha,i} - \lambda|v^-|\rho^n_{\alpha,i-\frac{1}{2},r} - \lambda v^+ \rho^n_{\alpha,i+\frac{1}{2},l}\right)\delta(v - v^n_{\alpha,i})$$

$$\Longrightarrow \quad \min_\alpha \left(\frac{\rho^n_{\alpha,i}}{|v^-_{\alpha,i}|\rho^n_{\alpha,i-\frac{1}{2},r} + v^+_{\alpha,i}\rho^n_{\alpha,i+\frac{1}{2},l}}\right) \geq \lambda$$

Use high-order, finite-volume schemes only for the weights



Realizable Time-Stepping Schemes

- First-order explicit Euler:

$$\mathbf{M}_i^{n+1} = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

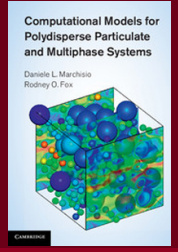
  is realizable because RHS is convex sum of $\mathbf{M}_i^n$

- Second-order Runga-Kutta (RK2):

$$\mathbf{M}_i^* = \mathbf{M}_i^n - \frac{1}{2}\lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

$$\mathbf{M}_i^{n+1} = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^*, \mathbf{M}_{i+\frac{1}{2},r}^*\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^*, \mathbf{M}_{i-\frac{1}{2},r}^*\right)\right]$$

  is not realizable because Step 2 is not convex sum



Realizable Time-Stepping Schemes

- First-order explicit Euler:

$$\mathbf{M}_i^{n+1} = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

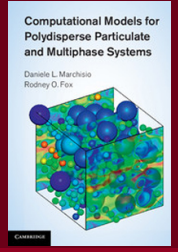
  is realizable because RHS is convex sum of $\mathbf{M}_i^n$

- Second-order Runga-Kutta (RK2):

$$\mathbf{M}_i^* = \mathbf{M}_i^n - \frac{1}{2}\lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

$$\mathbf{M}_i^{n+1} = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^*, \mathbf{M}_{i+\frac{1}{2},r}^*\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^*, \mathbf{M}_{i-\frac{1}{2},r}^*\right)\right]$$

  is not realizable because Step 2 is not convex sum



Realizable Time-Stepping Schemes

- RK2SSP:

$$\mathbf{M}_i^* = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

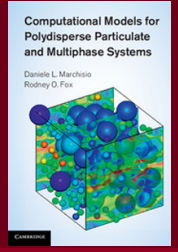
$$\mathbf{M}_i^{**} = \mathbf{M}_i^* - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^*, \mathbf{M}_{i+\frac{1}{2},r}^*\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^*, \mathbf{M}_{i-\frac{1}{2},r}^*\right)\right]$$

$$\mathbf{M}_i^{n+1} = \frac{1}{2}\left(\mathbf{M}_i^n + \mathbf{M}_i^{**}\right)$$

  is realizable because RHS are all convex sums of realizable moments

- In general, explicit, high-order, strong stability-preserving schemes are realizable

Achieve high-order in space and time on unstructured grids!



Realizable Time-Stepping Schemes

- RK2SSP:

$$\mathbf{M}_i^* = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

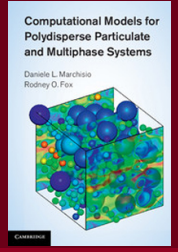
$$\mathbf{M}_i^{**} = \mathbf{M}_i^* - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^*, \mathbf{M}_{i+\frac{1}{2},r}^*\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^*, \mathbf{M}_{i-\frac{1}{2},r}^*\right)\right]$$

$$\mathbf{M}_i^{n+1} = \frac{1}{2}\left(\mathbf{M}_i^n + \mathbf{M}_i^{**}\right)$$

  is realizable because RHS are all convex sums of realizable moments

- In general, explicit, high-order, strong stability-preserving schemes are realizable

Achieve high-order in space and time on unstructured grids!



Realizable Time-Stepping Schemes

- RK2SSP:

$$\mathbf{M}_i^* = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

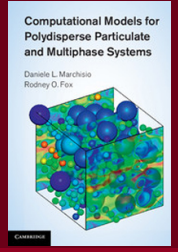
$$\mathbf{M}_i^{**} = \mathbf{M}_i^* - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^*, \mathbf{M}_{i+\frac{1}{2},r}^*\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^*, \mathbf{M}_{i-\frac{1}{2},r}^*\right)\right]$$

$$\mathbf{M}_i^{n+1} = \frac{1}{2}\left(\mathbf{M}_i^n + \mathbf{M}_i^{**}\right)$$

  is realizable because RHS are all convex sums of realizable moments

- In general, explicit, high-order, strong stability-preserving schemes are realizable

Achieve high-order in space and time on unstructured grids!



Realizable Time-Stepping Schemes

- RK2SSP:

$$\mathbf{M}_i^* = \mathbf{M}_i^n - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^n, \mathbf{M}_{i+\frac{1}{2},r}^n\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^n, \mathbf{M}_{i-\frac{1}{2},r}^n\right)\right]$$

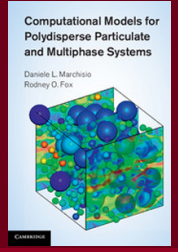
$$\mathbf{M}_i^{**} = \mathbf{M}_i^* - \lambda\left[\mathbf{G}\left(\mathbf{M}_{i+\frac{1}{2},l}^*, \mathbf{M}_{i+\frac{1}{2},r}^*\right) - \mathbf{G}\left(\mathbf{M}_{i-\frac{1}{2},l}^*, \mathbf{M}_{i-\frac{1}{2},r}^*\right)\right]$$

$$\mathbf{M}_i^{n+1} = \frac{1}{2}\left(\mathbf{M}_i^n + \mathbf{M}_i^{**}\right)$$

  is realizable because RHS are all convex sums of realizable moments

- In general, explicit, high-order, strong stability-preserving schemes are realizable

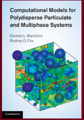
Achieve high-order in space and time on unstructured grids!



1-D Particle Trajectory Crossing: Main Program I

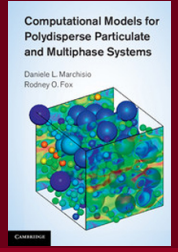
Matlab Script for 1-D PTC: $\partial_t \mathbf{M} + \partial_x \mathbf{F} = 0$

```
% Initializing numerical parameters
%
cfl = 0.9 ;
Ny = 402 ;
T0 = 1 ;
Ly = 1 ; % dimensionless width of domain
H = Ly ;  % width of domain (dimensionless units)
Tmax = 1 ; % final time
Dy = H/(Ny-2) ;  % cell size
eps = 1.d-6 ;
Ycell = -(Dy/2) : Dy : (H+Dy/2) ; % grid cell centers
%
% Initializing the cell values
M0 = zeros(5,Ny); % moments
N0 = 0*ones(4,Ny) ; % weights
U0 = zeros(4,Ny) ; % U velocity
N1 = N0 ;
U1 = U0 ;
%
% initial two waves in x direction
for i = 1:48
    N0(1,i) = 1 ;
    U0(1,i)  = 1 ;
```



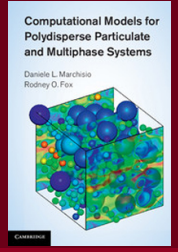
1-D Particle Trajectory Crossing: Main Program II

```
end
for i = Ny-47:Ny
    N0(1,i) = 1 ;
    U0(1,i)  = -1 ;
end
%
for i =1:Ny
    M0(:,i) = moments_4node_5mom(N0(:,i),U0(:,i)) ;
    [N1(:,i),U1(:,i)] = four_node_flux_hyqmom(M0(:,i)) ;
end
M0(:,1) = moments_4node_5mom(N0(:,Ny-1),U0(:,Ny-1)) ;
M0(:,Ny) = moments_4node_5mom(N0(:,2),U0(:,2)) ;
[N1(:,1),U1(:,1)] = four_node_flux_hyqmom(M0(:,1)) ;
[N1(:,Ny),U1(:,Ny)] = four_node_flux_hyqmom(M0(:,Ny)) ;
%
M1 = M0 ;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Time loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = 0 ;
while( t < Tmax + eps )
    Umax = 0.01 ;
    for j=1:4
```



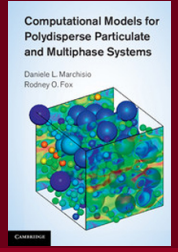
1-D Particle Trajectory Crossing: Main Program III


```
    for i=1:Ny
        Umax = max(Umax,abs(U0(j,i))) ;
    end
end
Dt = cfl*Dy/Umax ;
t = t + Dt ;
%
% Evaluation of the new moments using RK2SSP
% Step 1 du RK2SSP
%
% Step of spatial transport
for i=2:Ny-1
    % Etape 1
    Nlmoins = N0(:,i-1) ;
    Ulmoins = U0(:,i-1) ;
    Nlplus  = N0(:,i)   ;
    Ulplus  = U0(:,i)   ;
    Nrmoins = N0(:,i)   ;
    Urmoins = U0(:,i)   ;
    Nrplus  = N0(:,i+1) ;
    Urplus  = U0(:,i+1) ;
    %
    Fleft  = flux_5mom(Nlmoins,Ulmoins,Nlplus,Ulplus);
    Fright = flux_5mom(Nrmoins,Urmoins,Nrplus,Urplus);
```



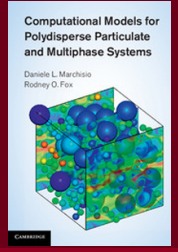
1-D Particle Trajectory Crossing: Main Program IV


```
    % update moments by time step
    M1(:,i) = M0(:,i) - (Dt/Dy)*(Fright - Fleft) ;
end
% update weights and abscissas
for i=2:Ny-1
    [N1(:,i),U1(:,i)] = four_node_flux_hyqmom(M1(:,i)) ;
    M1(:,i) = moments_4node_5mom(N1(:,i),U1(:,i)) ; % projection step
end
M1(:,1) = moments_4node_5mom(N1(:,Ny-1),U1(:,Ny-1)) ;
M1(:,Ny) = moments_4node_5mom(N1(:,2),U1(:,2)) ;
[N1(:,1),U1(:,1)] = four_node_flux_hyqmom(M1(:,1)) ;
[N1(:,Ny),U1(:,Ny)] = four_node_flux_hyqmom(M1(:,Ny)) ;
%
% Step 2 du RK2SSP:  M0 are old values, M1 are new values
% Step of spatial transport
for i=2:Ny-1
    Nlmoins = N1(:,i-1) ;
    Ulmoins = U1(:,i-1) ;
    Nlplus  = N1(:,i)   ;
    Ulplus  = U1(:,i)   ;
    Nrmoins = N1(:,i)   ;
    Urmoins = U1(:,i)   ;
    Nrplus  = N1(:,i+1) ;
    Urplus  = U1(:,i+1) ;
```



1-D Particle Trajectory Crossing: Main Program V

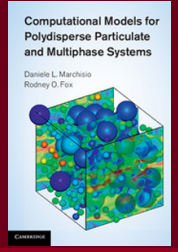

```
        %
        Fleft  = flux_5mom(Nlmoins,Ulmoins,Nlplus,Ulplus);
        Fright = flux_5mom(Nrmoins,Urmoins,Nrplus,Urplus);
        %
        % update moments by full time step
        M1(:,i) = M1(:,i) - (Dt/Dy)*(Fright - Fleft) ;
        M1(:,i) = 0.5*( M0(:,i) + M1(:,i) ) ;
    end
    % update weights and abscissas
    for i=2:Ny-1
        [N1(:,i),U1(:,i)] = four_node_flux_hyqmom(M1(:,i)) ;
        M1(:,i) = moments_4node_5mom((N1(:,i),U1(:,i)) ; % projection step
    end
    M1(:,1) = moments_4node_5mom(N1(:,Ny-1),U1(:,Ny-1)) ;
    M1(:,Ny) = moments_4node_5mom(N1(:,2),U1(:,2)) ;
    [N1(:,1),U1(:,1)] = four_node_flux_hyqmom(M1(:,1)) ;
    [N1(:,Ny),U1(:,Ny)] = four_node_flux_hyqmom(M1(:,Ny)) ;
    %
    % End of RK2SSP time step
    M0 = M1 ;
    N0 = N1 ;
    U0 = U1 ;
end
```



1-D Particle Trajectory Crossing: Kinetic-Based HyQMOM Program I

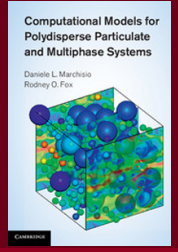
Matlab Script for 1-D PTC: $\mathbf{M} \Longrightarrow \mathbf{N}, \mathbf{U}$ kinetic based

```
function [ N, U ] = four_node_flux_hyqmom( M )
% [ N, U ] = four_node_flux_hyqmom( M )
% four-node flux-based HyQMOM with realizability checking
%
% input: 5 velocity moments (in this order)
% M = [ m0, m1, m2, m3, m4]
%
% output:
% N = [ n1, n2, n3, n4 ] weights
% U = [ u1, u2, u3, u4 ] abscissas (eigenvalues)
%
m0 = M(1); m1 = M(2); m2 = M(3); m3 = M(4); m4 = M(5);
%
%
if isnan(m0) == 1
    display(M)
    error('corrupted moments in four_node_flux_hyqmom')
end
%
etasmall = 1.d-10 ;  % smallest variance for computing eta
verysmall = 1.d-14 ; % smallest nonzero mass
qmax = 30 ; % maximum normalized skewness
% check for zero density
```



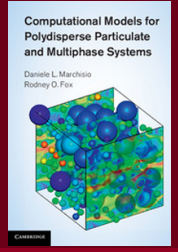
1-D Particle Trajectory Crossing: Kinetic-Based HyQMOM Program II


```
N = zeros(4,1) ; U = N ;
if m0 <= verysmall
    N(1:4) = m0/4 ;
    return
end
% mean velocities
bu = m1/m0 ;
% normalized moments
d2 = m2/m0 ;
d3 = m3/m0 ;
d4 = m4/m0 ;
% central moments
c2 = d2 - bu^2 ;
c3 = d3 - 3*bu*d2 + 2*bu^3 ;
c4 = d4 - 4*bu*d3 + 6*bu^2*d2 - 3*bu^4 ;
%
% realizability check
realizable = c2*c4 - c2^3 - c3^2 ;
if c2 < 0  % negative variance
    if c2 < - verysmall
        warning('c2 negative in four_node_flux_hyqmom')
        display(c2)
    end
    c2 = 0 ;
```



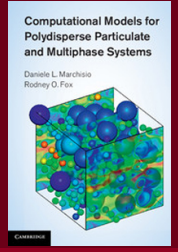
1-D Particle Trajectory Crossing: Kinetic-Based HyQMOM Program III


```
    c3 = 0 ;
    c4 = 0 ;
elseif realizable < 0 % eta < 1 + q^2
    if c2 >= etasmall
        q = c3/sqrt(c2)/c2 ; eta = c4/c2/c2 ;
        if abs(q) > verysmall
            slope = (eta - 3)/q ;
            det = 8 + slope^2 ;
            qp = 0.5*( slope + sqrt(det) ) ;
            qm = 0.5*( slope - sqrt(det) ) ;
            if sign(q) == 1
                q = qp ;
            else
                q = qm ;
            end
        else
            q = 0 ;
        end
        eta = q^2 + 1 ;
        c3 = q*sqrt(c2)*c2 ;
        c4 = eta*c2^2 ;
        if realizable < - 1.d-6
            warning('c4 too small in four_node_flux_hyqmom')
            display(realizable)
```



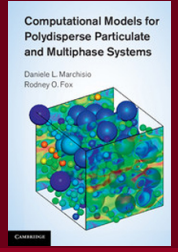
1-D Particle Trajectory Crossing: Kinetic-Based HyQMOM Program IV


```
            display(M)
        end
    else
        c3 = 0 ;
        c4 = c2^2 ;
    end
end
%
% HyQMOM parameters (scaled)
scale = sqrt(c2) ;
if c2 >= etasmall
    q = c3/sqrt(c2)/c2 ;
    eta = c4/c2/c2 ;
else
    q = 0 ; eta = 1 ;
end
% bound skewness < qmax
if q^2 > qmax^2
    slope = (eta - 3)/q ; % move towards Gaussian moments
    q = qmax*sign(q) ;
    eta = 3 + slope*q ;
    realizable = eta - 1 - q^2 ;
    if realizable < 0
        eta = 1 + q^2 ;
```



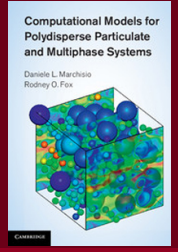
1-D Particle Trajectory Crossing: Kinetic-Based HyQMOM Program V


```
    end
end
% compute weights & abscissas (eigenvalues)
if eta > 1 + q^2
    up = zeros(1,4) ;
    up(1) = 0.5*(q + sqrt(4*eta - 3*q^2 + 4*sqrt((eta - q^2)*(eta - q^2 - 1)))) ;
    up(2) = 0.5*(q + sqrt(4*eta - 3*q^2 - 4*sqrt((eta - q^2)*(eta - q^2 - 1)))) ;
    up(3) = 0.5*(q - sqrt(4*eta - 3*q^2 - 4*sqrt((eta - q^2)*(eta - q^2 - 1)))) ;
    up(4) = 0.5*(q - sqrt(4*eta - 3*q^2 + 4*sqrt((eta - q^2)*(eta - q^2 - 1)))) ;
    A = [up ; up.^2 ; up.^3 ; up.^4 ] ;
    b = [0 1 q eta]' ;
    rho = A\b ;
else % eta = 1 + q^2
    rho = zeros(4,1) ;
    dm = sqrt(4*eta - 3*q^2) ;
    rho(1) = 1/(dm + q)/dm ;
    rho(2) = rho(1) ;
    rho(3) = 1/(dm - q)/dm ;
    rho(4) = rho(3) ;
    up = zeros(4,1) ;
    up(1) = 0.5*(q + dm) ;
    up(2) = up(1) ;
    up(3) = 0.5*(q - dm) ;
    up(4) = up(3) ;
```



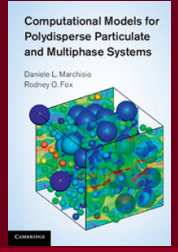
1-D Particle Trajectory Crossing: Kinetic-Based HyQMOM Program VI


```
end
%
% error checking
if max(isnan(rho)) == 1 || max(isnan(up)) == 1
    display(rho)
    display(up)
    error('corrupted moments in HyQMOM')
end
if min(rho) < 0
    format long
    display(rho)
    display(c2)
    display(q)
    display(eta)
    warning('negative weight in HyQMOM')
    format short
end
%
N(1) = rho(1) ;
N(2) = rho(2) ;
N(3) = rho(3) ;
N(4) = rho(4) ;
N = m0*N ;
%
```



1-D Particle Trajectory Crossing: Kinetic-Based HyQMOM Program VII

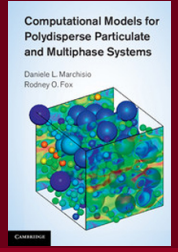

```
U(1) = scale*up(1);
U(2) = scale*up(2);
U(3) = scale*up(3);
U(4) = scale*up(4);
U = bu + U ;
%
% check moment error
m0o = sum(N) ;
m1o = sum(N.*U) ;
m2o = sum(N.*U.^2) ;
m3o = sum(N.*U.^3) ;
m4o = sum(N.*U.^4) ;
Mout = [ m0o m1o m2o m3o m4o] ;
err = Mout - M ;
if err(3) > 1.d-6 && m0 > verysmall
    display(err)
    display(M)
end
%
end
```



1-D Particle Trajectory Crossing: Moment Program I

Matlab Script for 1-D PTC: $\mathbf{N}, \mathbf{U} \Longrightarrow \mathbf{M}$ kinetic based

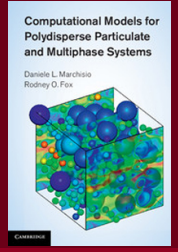
```
function mom = moments_4node_5mom(n,u)
% moments from four-node, kinetic-based HyQMOM weights and abscissas
%
% input:
% N = [ n1, n2, n3, n4] weights
% U = [ u1, u2, u3, u4] abscissas
%
% output: 5 velocity moments (in this order)
% M = [ m0, m1, m2, m3, m4]
%
mom = zeros(5,1) ;
k = [0 1 2 3 4] ;
for i = 1:5  % moments number
    for j = 1:4 % node number
        mom(i) = mom(i) + n(j)*u(j)^k(i) ;
    end
end
end
```



1-D Particle Trajectory Crossing: Kinetic-Based Flux Program I

Matlab Script for 1-D PTC: $\mathbf{N}_l, \mathbf{U}_l \,|\, \mathbf{N}_r, \mathbf{U}_r \Longrightarrow \mathbf{F}$

```
function F = flux_5mom(Nl,Ul,Nr,Ur)
% HyQMOM kinetic-based flux across cell face
%
% input:
% Nl = [ n1, n2, n3, n4] left weights
% Ul = [ u1, u2, u3, u4] left abscissas
% Nr = [ n1, n2, n3, n4] right weights
% Ur = [ u1, u2, u3, u4] right abscissas
%
% output: Net flux of moments across cell face
% F = [ F0, F1, F2, F3, F4]
%
F = zeros(5,1) ;
k = [0 1 2 3 4] ;
for i = 1:5  % moments number
    for j = 1:4  % nodes number
        F(i) = F(i) + Nl(j)*Ul(j)^k(i) * max(Ul(j),0) ...
                    + Nr(j)*Ur(j)^k(i) * min(Ur(j),0) ;
    end
end
end
```

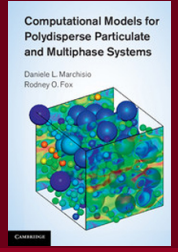


Particle Trajectory Crossing in 1-D

Uncorrelated moments

Loading movie. . .

16-moment, 27-node CHyQMOM

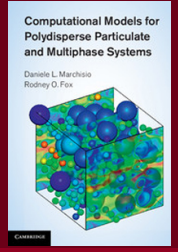


Particle Trajectory Crossing in 1-D

Quadrature weights and abscissas

Loading movie. . .

16-moment, 27-node CHyQMOM

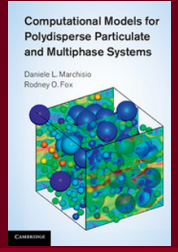


Particle Trajectory Crossing in 1-D

Perfectly correlated moments

Loading movie. . .

16-moment, 27-node CHyQMOM

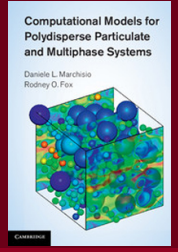


Particle Trajectory Crossing in 1-D

Quadrature weights and abscissas

Loading movie. . .

16-moment, 27-node CHyQMOM

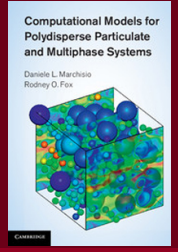


Particle Trajectory Crossing in 1-D

Partially correlated moments

Loading movie. . .

16-moment, 27-node CHyQMOM

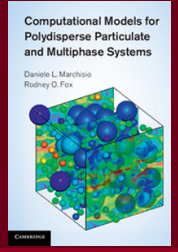


Particle Trajectory Crossing in 1-D

Quadrature weights and abscissas

Loading movie. . .

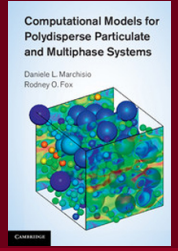
16-moment, 27-node CHyQMOM



Particle Trajectory Crossing in 2-D

Loading movie. . .

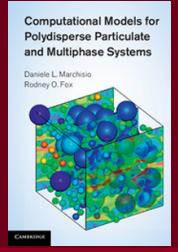
10-moment, 9-node CHyQMOM



Particle Trajectory Crossing in 3-D

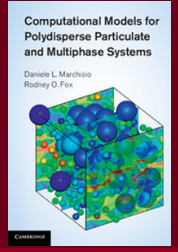
Loading movie. . .

16-moment, 27-node CHyQMOM



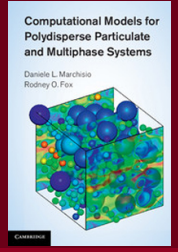
Summary of KBFVM

- When solving moment transport equations, **we must guarantee realizability!**
- First-order FV methods are realizable, but **too diffusive**
- Standard high-order FV methods lead to **unrealizable moments!**
- Kinetic-based flux functions can be designed to be realizable
- Use **quadrature representation** (e.g. HyQMOM) with high-order spatial reconstruction of weights
- High-order time-stepping schemes are also possible
- **KBFVM provide robust treatment of shocks/discontinuous solutions on unstructured grids**



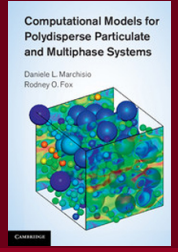
Summary of KBFVM

- When solving moment transport equations, **we must guarantee realizability!**
- First-order FV methods are realizable, but **too diffusive**
- Standard high-order FV methods lead to **unrealizable moments!**
- Kinetic-based flux functions can be designed to be realizable
- Use **quadrature representation** (e.g. HyQMOM) with high-order spatial reconstruction of weights
- High-order time-stepping schemes are also possible
- **KBFVM provide robust treatment of shocks/discontinuous solutions on unstructured grids**



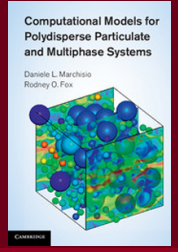
Summary of KBFVM

- When solving moment transport equations, **we must guarantee realizability!**
- First-order FV methods are realizable, but **too diffusive**
- Standard high-order FV methods lead to **unrealizable moments!**
- Kinetic-based flux functions can be designed to be realizable
- Use **quadrature representation** (e.g. HyQMOM) with high-order spatial reconstruction of weights
- High-order time-stepping schemes are also possible
- **KBFVM provide robust treatment of shocks/discontinuous solutions on unstructured grids**



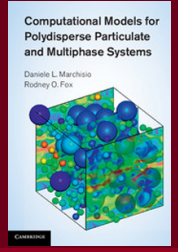
Summary of KBFVM

- When solving moment transport equations, **we must guarantee realizability!**
- First-order FV methods are realizable, but **too diffusive**
- Standard high-order FV methods lead to **unrealizable moments!**
- Kinetic-based flux functions can be designed to be realizable
- Use **quadrature representation** (e.g. HyQMOM) with high-order spatial reconstruction of weights
- High-order time-stepping schemes are also possible
- **KBFVM provide robust treatment of shocks/discontinuous solutions on unstructured grids**



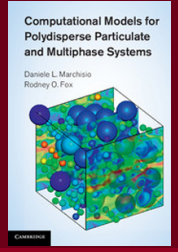
Summary of KBFVM

- When solving moment transport equations, **we must guarantee realizability!**
- First-order FV methods are realizable, but **too diffusive**
- Standard high-order FV methods lead to **unrealizable moments!**
- Kinetic-based flux functions can be designed to be realizable
- Use **quadrature representation** (e.g. HyQMOM) with high-order spatial reconstruction of weights
- High-order time-stepping schemes are also possible
- **KBFVM provide robust treatment of shocks/discontinuous solutions on unstructured grids**



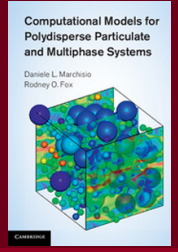
Summary of KBFVM

- When solving moment transport equations, **we must guarantee realizability!**
- First-order FV methods are realizable, but **too diffusive**
- Standard high-order FV methods lead to **unrealizable moments!**
- Kinetic-based flux functions can be designed to be realizable
- Use **quadrature representation** (e.g. HyQMOM) with high-order spatial reconstruction of weights
- High-order time-stepping schemes are also possible
- KBFVM provide robust treatment of shocks/discontinuous solutions on unstructured grids



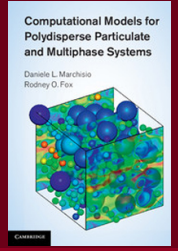
Summary of KBFVM

- When solving moment transport equations, **we must guarantee realizability!**
- First-order FV methods are realizable, but **too diffusive**
- Standard high-order FV methods lead to **unrealizable moments!**
- Kinetic-based flux functions can be designed to be realizable
- Use **quadrature representation** (e.g. HyQMOM) with high-order spatial reconstruction of weights
- High-order time-stepping schemes are also possible
- **KBFVM provide robust treatment of shocks/discontinuous solutions on unstructured grids**



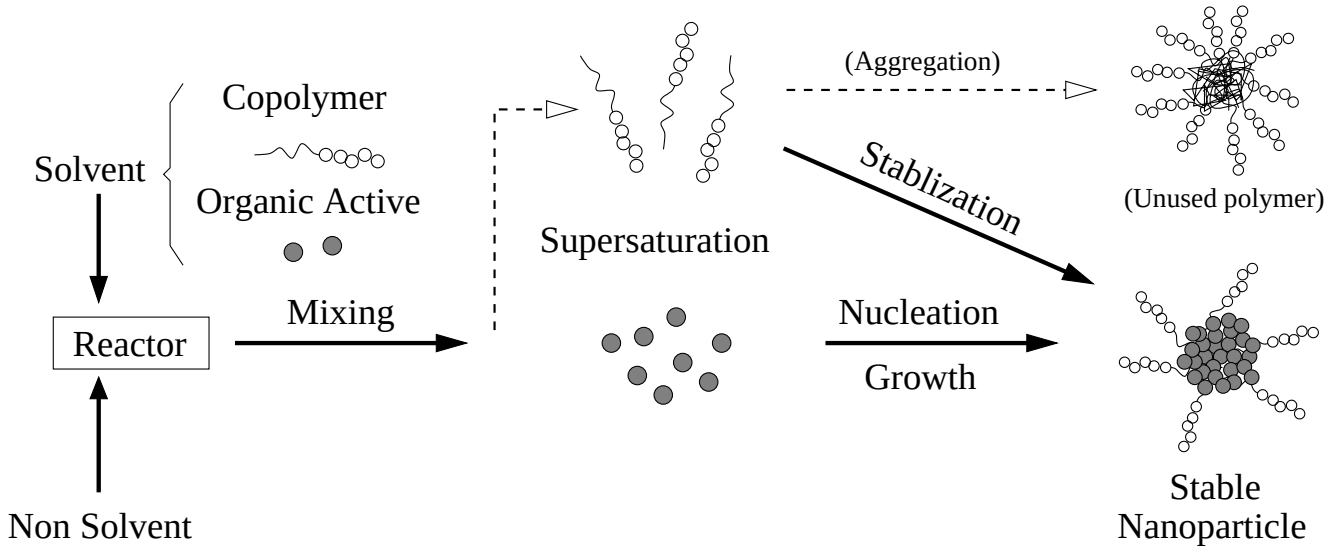
Part 5

APPLICATION TO FINE-PARTICLE FORMATION



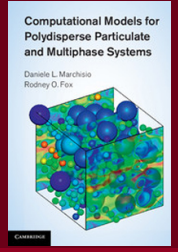
Nanoparticles in Turbulent Mixers

Flash Nanoprecipitation



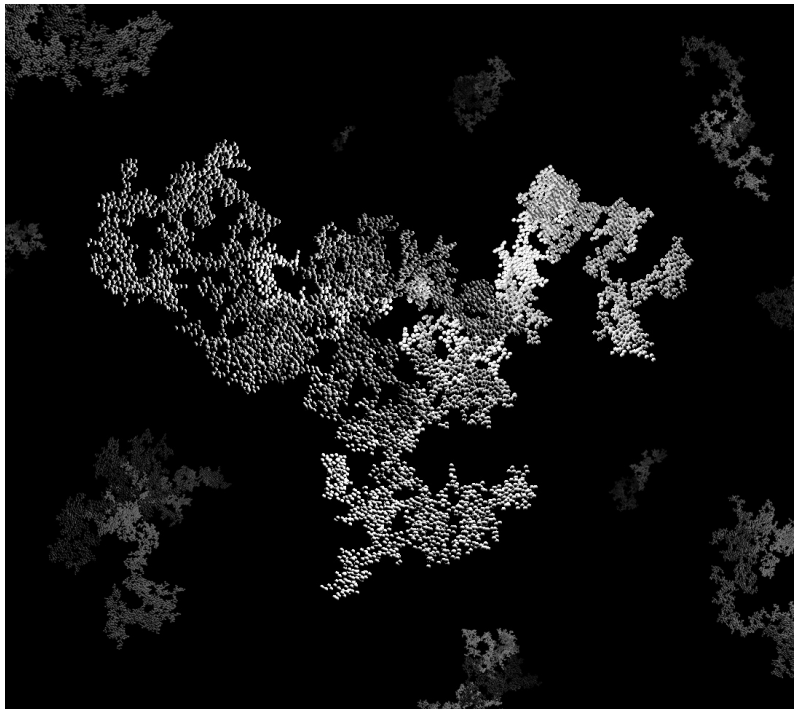
Produced in Multi-Inlet Vortex Reactor

Model needed for turbulent mixing and PBE

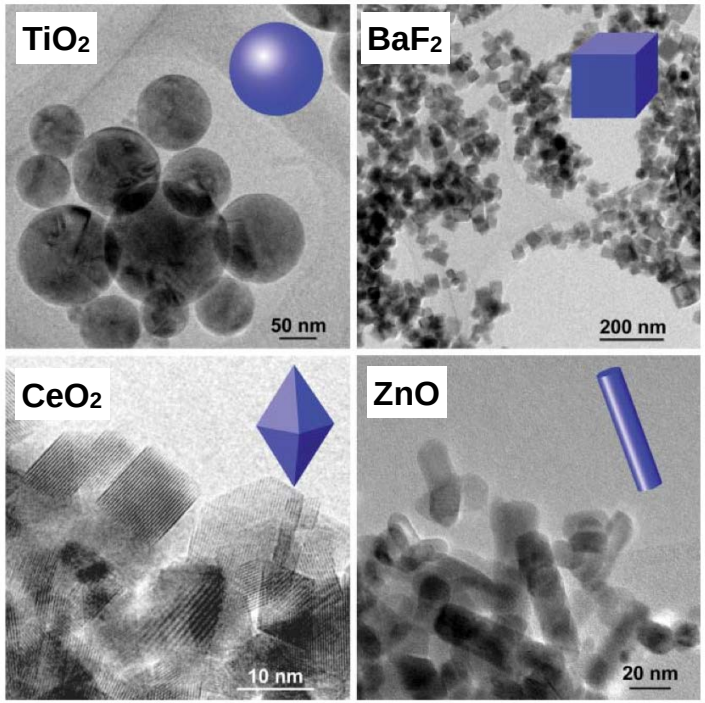


Nanoparticles in Turbulent Flames

Soot

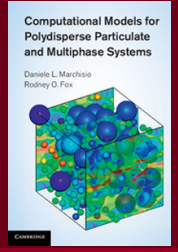


Metal oxides



(Strobel & Pratsinis 2007)

Model needed for turbulent combustion and PBE

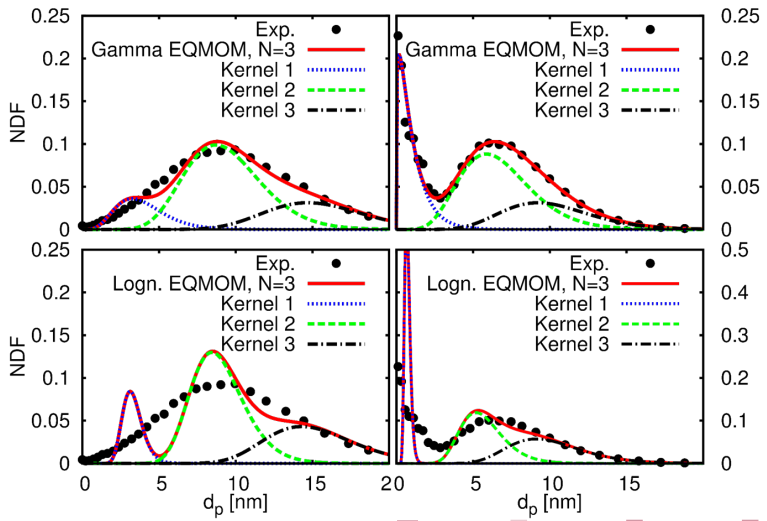


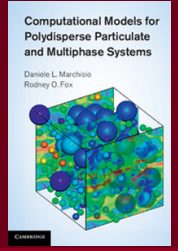
Computational Model

Part. size dist. (PSD)

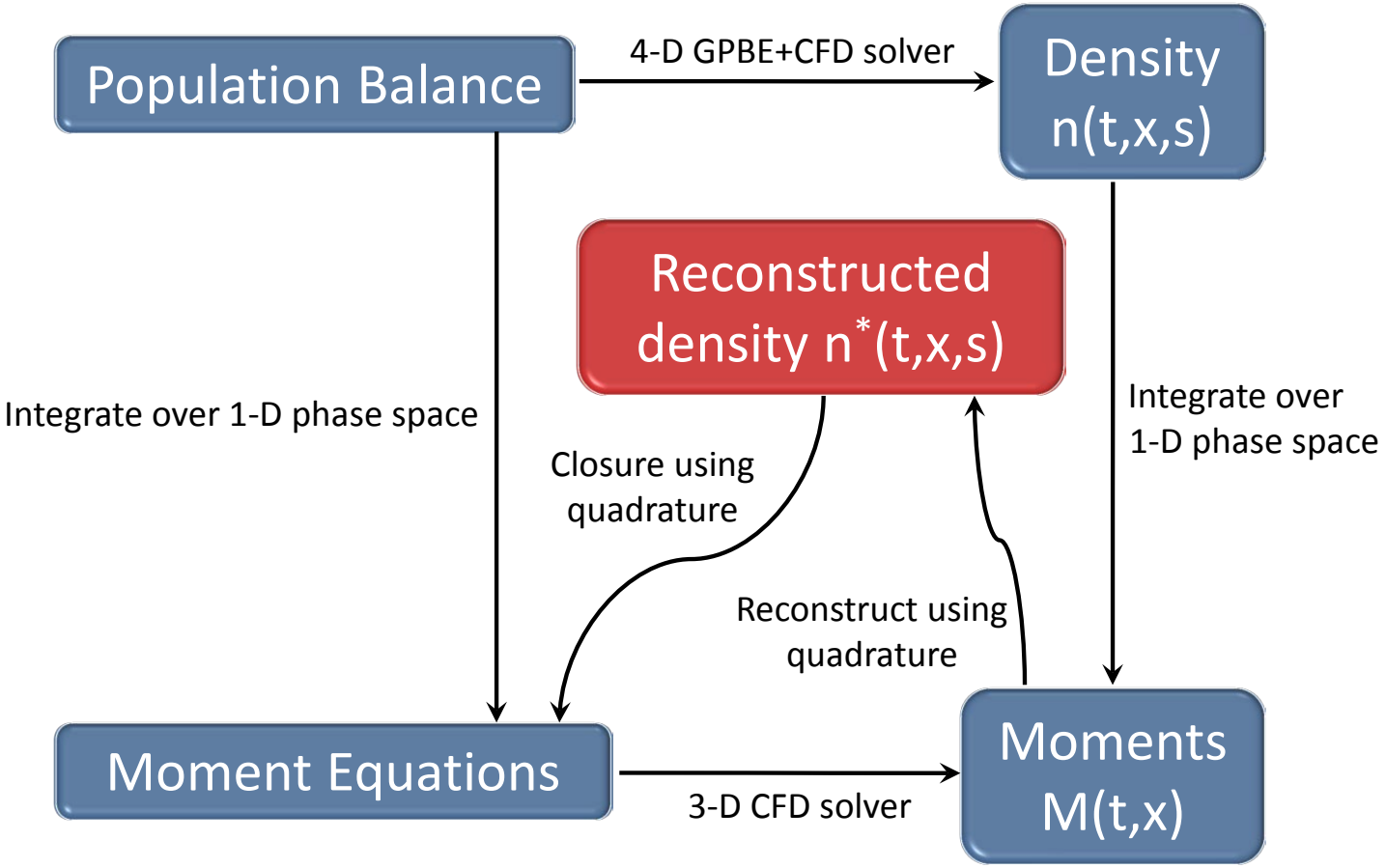
- PBE
  - Solve for NDF $n(v)$
  - Coupled to velocity, reactive scalars
  - Non-linear integro-PDE
- QBMM
  - Solve for moments of NDF
  - Close by reconstructing $n(v)$

$$\underbrace{\frac{\partial n(v)}{\partial t}}_{\left(\text{Rate of change}\right)} + \underbrace{\nabla\cdot\left(\mathbf{u}\,n(v)\right)}_{\text{(Convection)}} = \underbrace{\nabla\cdot\left(\Gamma(v,\Phi)\,\nabla n(v)\right)}_{\text{(Particle diffusion)}} + \underbrace{J(\Phi)\,f(v_0)}_{\text{(Nucelation)}}$$
$$+ \underbrace{\int_0^\infty \beta(v-v',v')\,n(v-v')\,n(v')\,dv'}_{\text{(Aggregation-birth)}}$$
$$- \underbrace{\int_0^\infty \beta(v,v')n(v)n(v')\,dv'}_{\text{(Aggregation-death)}} \quad ,$$

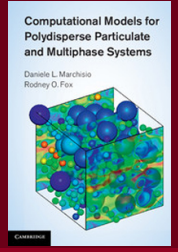




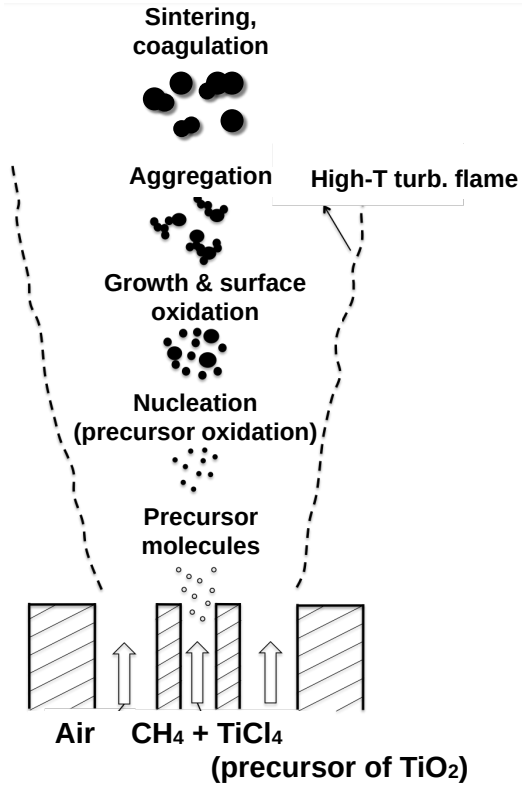
Quadrature-Based Moment Methods

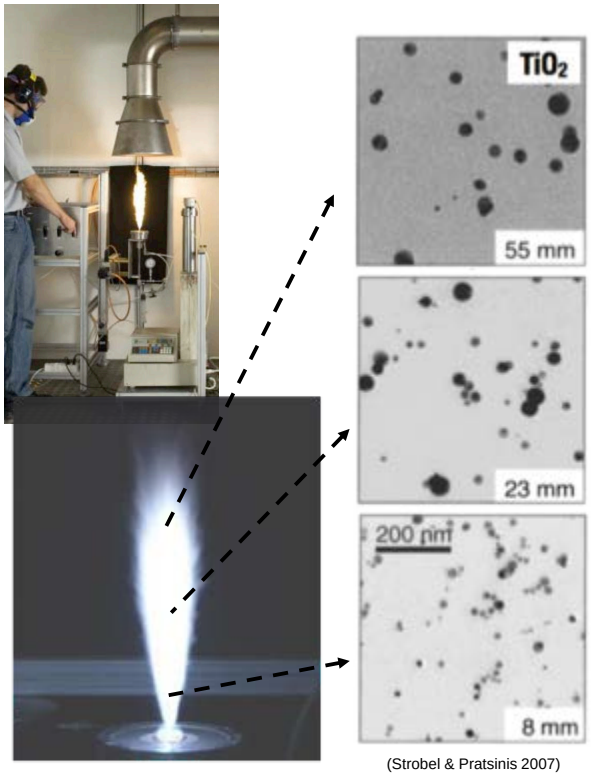


Close moment equations by reconstructing density function

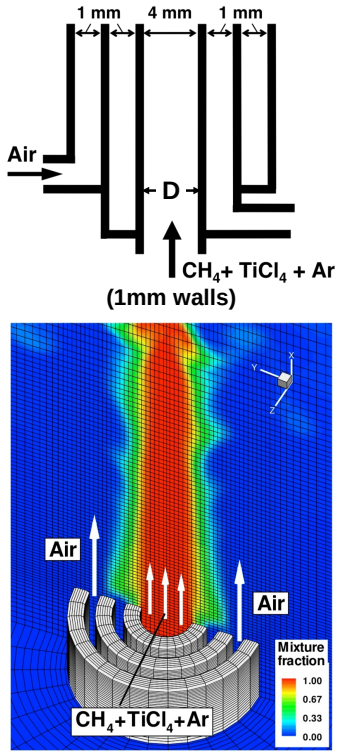


Flame Synthesis of Metal-Oxide Nanoparticles

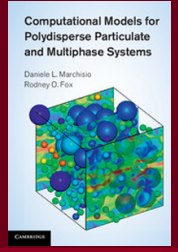




(Strobel & Pratsinis 2007)



- Flamelet model for turbulent combustion
- QBMM for volume–surface NDF $n(v, a)$
- PBE accounts for nucleation, growth, aggregation, and sintering



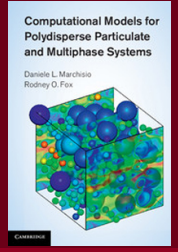
Bivariate Population Balance Equation

$$\frac{\partial n(v,a)}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{U}_g n(v,a) + \frac{\partial G_v(v,a)n(v,a)}{\partial v} + \frac{\partial G_a(v,a)n(v,a)}{\partial a} + \frac{\partial S_a(v,a)n(v,a)}{\partial a} = J(T,\phi)\delta(v-v_0)\delta(a-a_0) + B(v,a) - D(v,a)$$

- G_v, G_a: surface growth rates of volume and surface area
- S_a: rate of change of surface area due to sintering
- J: nucleation rate of particles volume v_0 and surface area a_0
- Aggregation birth and death terms:

$$B = \frac{1}{2}\int_0^a \int_0^v \beta(v-v^*, v^*, a-a^*, a^*) n(v-v^*, a-a^*) n(v^*, a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*$$

$$D = \int_0^\infty \int_0^\infty \beta(v, v^*, a, a^*) n(v,a) n(v^*, a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*$$



Mesoscale Models

- Nucleation: $k_{\text{nucl}} = 10^{13}$ (cm^6/mol^{2}s)

$$J(T,\phi) = k_{\text{nucl}} N_{\text{av}} [\text{Ti}_5\text{O}_6\text{Cl}_8][\text{O}_2]^2$$

$v_0 = 16.6 \times 10^{-29}$ m^3, $a_0 = 14.6 \times 10^{-19}$ m^2

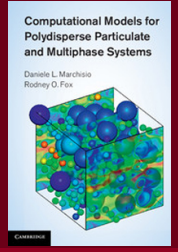
- Surface growth $k_{\text{s}} = 49 \exp(-8993/T)$ (m/s)

$$G_v(v,a) = k_{\text{s}} [\text{TiCl}_4] N_{\text{av}} v_0 a$$

$$G_a(v,a) = 4 k_{\text{s}} [\text{TiCl}_4] N_{\text{av}} v_0 \sqrt{\pi a}$$

N_{av} is Avagadro's number

Gas-phase species found from detailed combustion model for TiCl_4



Mesoscale Models

- Sintering

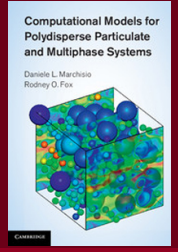
$$S_a(v,a) = \begin{cases} \frac{(a-a_s)}{\tau_f(d_p^*)} & \text{if } n_p \leq 2 \\ (n_p - 1)\left(\frac{0.41 a_p}{\tau_f(d_p)}\right) & \text{if } n_p > 2 \end{cases}$$

$$a_s = (36\pi v^2)^{1/3},\ d_p^* = (3v/\pi)^{1/3},\ d_p = 6v/a,\ a_p = 36\pi v^2/a^2,$$
$$n_p = a^3/(36\pi v^2)$$

Sintering time scale

$$\tau_f(x) = k_0 x^m \frac{T}{T_0} \exp\left[\frac{E_a}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right]$$

$$k_0 = 1 \times 10^{28}\ \text{m}^{-4},\ T_0 = 1400\ \text{K},\ m = 4,\ E_a = 1.5 \times 10^5\ \text{J/mol}$$



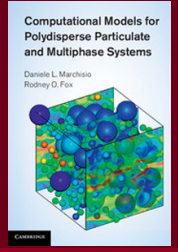
Mesoscale Models

- Aggregation kernel

$$\beta(v, a, v^*, a^*) = \frac{2RT}{3\mu N_{\text{av}}} \left(\frac{1}{v^{1/d_{\text{f}}}} + \frac{1}{v^{*1/d_{\text{f}}}} \right) \left(v^{1/d_{\text{f}}} + v^{*1/d_{\text{f}}} \right)$$

μ gas viscosity, $d_{\text{f}}(v, a)$ fractal dimension (≈ 2.5)

Models lead to stiff equations that are strongly coupled to gas-phase chemistry and temperature!



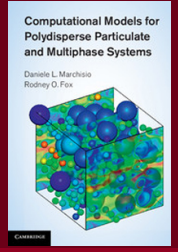
Mesoscale Models

- Aggregation kernel

$$\beta(v, a, v^*, a^*) = \frac{2RT}{3\mu N_{\text{av}}} \left(\frac{1}{v^{1/d_{\text{f}}}} + \frac{1}{v^{*1/d_{\text{f}}}} \right) \left(v^{1/d_{\text{f}}} + v^{*1/d_{\text{f}}} \right)$$

μ gas viscosity, $d_{\text{f}}(v, a)$ fractal dimension (≈ 2.5)

Models lead to stiff equations that are strongly coupled to gas-phase chemistry and temperature!



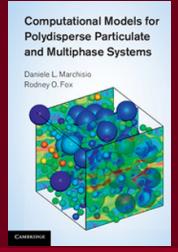
Mesoscale Models

- Aggregation kernel

$$\beta(v, a, v^*, a^*) = \frac{2RT}{3\mu N_{\text{av}}} \left(\frac{1}{v^{1/d_{\text{f}}}} + \frac{1}{v^{*1/d_{\text{f}}}} \right) \left(v^{1/d_{\text{f}}} + v^{*1/d_{\text{f}}} \right)$$

μ gas viscosity, $d_{\text{f}}(v, a)$ fractal dimension (≈ 2.5)

Models lead to stiff equations that are strongly coupled to gas-phase chemistry and temperature!



Bivariate Moment Equations

- Moments of NDF are defined by

$$m_{k,l} = \int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a$$

- Transport equation for moment $m_{k,l}$ is found from PBE using

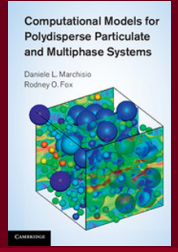
$$\int_0^\infty \int_0^\infty a^l v^k [\mathrm{PBE}]\,\mathrm{d}v\,\mathrm{d}a$$

- Accumulation term:

$$\int_0^\infty \int_0^\infty a^l v^k \frac{\partial n(v,a)}{\partial t}\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial}{\partial t}\int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial m_{k,l}}{\partial t}$$

- Advection term:

$$\int_0^\infty \int_0^\infty a^l v^k \nabla_{\mathbf{x}} \cdot \mathbf{U}_g n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \nabla_{\mathbf{x}} \cdot \mathbf{U}_g m_{k,l}$$



Bivariate Moment Equations

- Moments of NDF are defined by

$$m_{k,l} = \int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a$$

- Transport equation for moment $m_{k,l}$ is found from PBE using

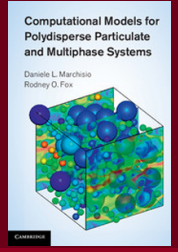
$$\int_0^\infty \int_0^\infty a^l v^k [\mathrm{PBE}]\,\mathrm{d}v\,\mathrm{d}a$$

- Accumulation term:

$$\int_0^\infty \int_0^\infty a^l v^k \frac{\partial n(v,a)}{\partial t}\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial}{\partial t}\int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial m_{k,l}}{\partial t}$$

- Advection term:

$$\int_0^\infty \int_0^\infty a^l v^k \nabla_{\mathbf{x}} \cdot \mathbf{U}_g n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \nabla_{\mathbf{x}} \cdot \mathbf{U}_g m_{k,l}$$



Bivariate Moment Equations

- Moments of NDF are defined by

$$m_{k,l} = \int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a$$

- Transport equation for moment $m_{k,l}$ is found from PBE using

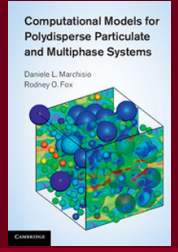
$$\int_0^\infty \int_0^\infty a^l v^k [\mathrm{PBE}]\,\mathrm{d}v\,\mathrm{d}a$$

- Accumulation term:

$$\int_0^\infty \int_0^\infty a^l v^k \frac{\partial n(v,a)}{\partial t}\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial}{\partial t}\int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial m_{k,l}}{\partial t}$$

- Advection term:

$$\int_0^\infty \int_0^\infty a^l v^k \nabla_{\mathbf{x}} \cdot \mathbf{U}_g n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \nabla_{\mathbf{x}} \cdot \mathbf{U}_g m_{k,l}$$



Bivariate Moment Equations

- Moments of NDF are defined by

$$m_{k,l} = \int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a$$

- Transport equation for moment $m_{k,l}$ is found from PBE using

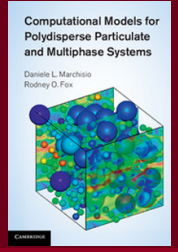
$$\int_0^\infty \int_0^\infty a^l v^k [\mathrm{PBE}]\,\mathrm{d}v\,\mathrm{d}a$$

- Accumulation term:

$$\int_0^\infty \int_0^\infty a^l v^k \frac{\partial n(v,a)}{\partial t}\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial}{\partial t}\int_0^\infty \int_0^\infty a^l v^k n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \frac{\partial m_{k,l}}{\partial t}$$

- Advection term:

$$\int_0^\infty \int_0^\infty a^l v^k \nabla_{\mathbf{x}} \cdot \mathbf{U}_g n(v,a)\,\mathrm{d}v\,\mathrm{d}a = \nabla_{\mathbf{x}} \cdot \mathbf{U}_g m_{k,l}$$



Bivariate Moment Equations

- Volume growth term:

$$\int_0^\infty \left(\int_0^\infty v^k \frac{\partial}{\partial v} [G_v(v,a) n(v,a)] \, \mathrm{d}v \right) a^l \, \mathrm{d}a$$

$$= \int_0^\infty \left[v^k G_v(v,a) n(v,a) \Big|_0^\infty - \int_0^\infty k v^{k-1} G_v(v,a) n(v,a) \, \mathrm{d}v \right] a^l \, \mathrm{d}a$$

- after integration by parts and $0 \leq G_v$

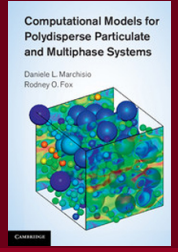
$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial v} [G_v(v,a) n(v,a)] dv da$$

$$= - \int_0^\infty \int_0^\infty k v^{k-1} a^l G_v(v,a) n(v,a) dv da$$

- Surface area growth term:

$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial a} [G_a(v,a) n(v,a)] \, \mathrm{d}v \, \mathrm{d}a$$

$$= - \int_0^\infty \int_0^\infty l v^k a^{l-1} G_a(v,a) n(v,a) \, \mathrm{d}v \, \mathrm{d}a$$



Bivariate Moment Equations

- Volume growth term:

$$\int_0^\infty \left(\int_0^\infty v^k \frac{\partial}{\partial v} [G_v(v,a) n(v,a)] \, \mathrm{d}v \right) a^l \, \mathrm{d}a$$

$$= \int_0^\infty \left[v^k G_v(v,a) n(v,a) \Big|_0^\infty - \int_0^\infty k v^{k-1} G_v(v,a) n(v,a) \, \mathrm{d}v \right] a^l \, \mathrm{d}a$$

- after integration by parts and $0 \leq G_v$

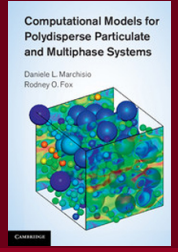
$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial v} [G_v(v,a) n(v,a)] dv da$$

$$= - \int_0^\infty \int_0^\infty k v^{k-1} a^l G_v(v,a) n(v,a) dv da$$

- Surface area growth term:

$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial a} [G_a(v,a) n(v,a)] \, \mathrm{d}v \, \mathrm{d}a$$

$$= - \int_0^\infty \int_0^\infty l v^k a^{l-1} G_a(v,a) n(v,a) \, \mathrm{d}v \, \mathrm{d}a$$



Bivariate Moment Equations

- Volume growth term:

$$\int_0^\infty \left(\int_0^\infty v^k \frac{\partial}{\partial v} [G_v(v,a) n(v,a)] \, \mathrm{d}v \right) a^l \, \mathrm{d}a$$

$$= \int_0^\infty \left[v^k G_v(v,a) n(v,a) \Big|_0^\infty - \int_0^\infty k v^{k-1} G_v(v,a) n(v,a) \, \mathrm{d}v \right] a^l \, \mathrm{d}a$$

- after integration by parts and $0 \leq G_v$

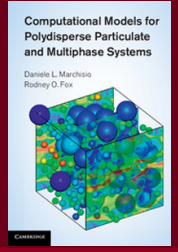
$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial v} [G_v(v,a) n(v,a)] dv da$$

$$= - \int_0^\infty \int_0^\infty k v^{k-1} a^l G_v(v,a) n(v,a) dv da$$

- Surface area growth term:

$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial a} [G_a(v,a) n(v,a)] \, \mathrm{d}v \, \mathrm{d}a$$

$$= - \int_0^\infty \int_0^\infty l v^k a^{l-1} G_a(v,a) n(v,a) \, \mathrm{d}v \, \mathrm{d}a$$



Bivariate Moment Equations

- Sintering term:

$$\int_0^\infty \left(\int_0^\infty a^l \frac{\partial}{\partial a} [S_a(v,a) n(v,a)] \, \mathrm{d}a \right) v^k \, \mathrm{d}v$$

$$= \int_0^\infty \left[a^l S_a(v,a) n(v,a) \Big|_0^\infty - \int_0^\infty l a^{l-1} S_a(v,a) n(v,a) \, \mathrm{d}v \right] v^k \, \mathrm{d}a$$

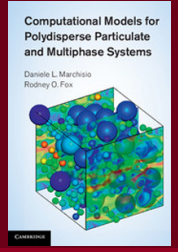
- after integration by parts

$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial a} [S_a(v,a) n(v,a)] \, \mathrm{d}v \, \mathrm{d}a$$

$$= - \int_0^\infty \int_0^\infty l v^k a^{l-1} S_a(v,a) n(v,a) \, \mathrm{d}v \, \mathrm{d}a$$

- Nucleation term:

$$\int_0^\infty \int_0^\infty J(T,\phi) \delta(v - v_0) \delta(a - a_0) a^l v^k \, \mathrm{d}v \, \mathrm{d}a = J(T,\phi) v_0^k a_0^l$$



Bivariate Moment Equations

- Sintering term:

$$\int_0^\infty \left(\int_0^\infty a^l \frac{\partial}{\partial a} [S_a(v,a) n(v,a)] \, \mathrm{d}a \right) v^k \, \mathrm{d}v$$

$$= \int_0^\infty \left[a^l S_a(v,a) n(v,a) \Big|_0^\infty - \int_0^\infty l a^{l-1} S_a(v,a) n(v,a) \, \mathrm{d}v \right] v^k \, \mathrm{d}a$$

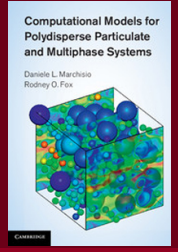
- after integration by parts

$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial a} [S_a(v,a) n(v,a)] \, \mathrm{d}v \, \mathrm{d}a$$

$$= - \int_0^\infty \int_0^\infty l v^k a^{l-1} S_a(v,a) n(v,a) \, \mathrm{d}v \, \mathrm{d}a$$

- Nucleation term:

$$\int_0^\infty \int_0^\infty J(T,\phi) \delta(v - v_0) \delta(a - a_0) a^l v^k \, \mathrm{d}v \, \mathrm{d}a = J(T,\phi) v_0^k a_0^l$$



Bivariate Moment Equations

- Sintering term:

$$\int_0^\infty \left(\int_0^\infty a^l \frac{\partial}{\partial a} [S_a(v,a) n(v,a)] \, \mathrm{d}a \right) v^k \, \mathrm{d}v$$

$$= \int_0^\infty \left[a^l S_a(v,a) n(v,a) \Big|_0^\infty - \int_0^\infty l a^{l-1} S_a(v,a) n(v,a) \, \mathrm{d}v \right] v^k \, \mathrm{d}a$$

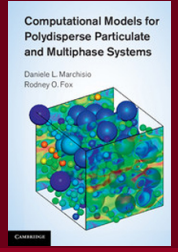
- after integration by parts

$$\int_0^\infty \int_0^\infty v^k a^l \frac{\partial}{\partial a} [S_a(v,a) n(v,a)] \, \mathrm{d}v \, \mathrm{d}a$$

$$= - \int_0^\infty \int_0^\infty l v^k a^{l-1} S_a(v,a) n(v,a) \, \mathrm{d}v \, \mathrm{d}a$$

- Nucleation term:

$$\int_0^\infty \int_0^\infty J(T,\phi) \delta(v - v_0) \delta(a - a_0) a^l v^k \, \mathrm{d}v \, \mathrm{d}a = J(T,\phi) v_0^k a_0^l$$



Bivariate Moment Equations

- Aggregation birth term:

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\,\mathrm{d}v\,\mathrm{d}a$$

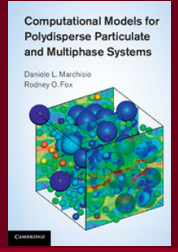
$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_0^v \beta(v-v^*,v^*,a-a^*,a^*)n(v-v^*,a-a^*)n(v^*,a^*)\,\mathrm{d}v^* v^k\,\mathrm{d}v\right]\mathrm{d}a^*\,\mathrm{d}a$$

- change in order of integration

$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_{v^*}^\infty \beta(v-v^*,v^*,a-a^*,a^*)n(v-v^*,a-a^*)v^k\,\mathrm{d}v\, n(v^*,a^*)\,\mathrm{d}v^*\right]\mathrm{d}a^*\,\mathrm{d}a$$

- change of variables $v' = v - v^*$

$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_0^\infty \beta(v',v^*,a-a^*,a^*)n(v',a-a^*)(v'+v^*)^k\,\mathrm{d}v' n(v^*,a^*)\,\mathrm{d}v^*\right]\mathrm{d}a^*\,\mathrm{d}a$$



Bivariate Moment Equations

- Aggregation birth term:

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\,\mathrm{d}v\,\mathrm{d}a$$

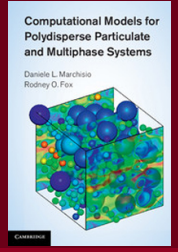
$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_0^v \beta(v-v^*,v^*,a-a^*,a^*)n(v-v^*,a-a^*)n(v^*,a^*)\,\mathrm{d}v^* v^k\,\mathrm{d}v\right]\mathrm{d}a^*\,\mathrm{d}a$$

- change in order of integration

$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_{v^*}^\infty \beta(v-v^*,v^*,a-a^*,a^*)n(v-v^*,a-a^*)v^k\,\mathrm{d}v\,n(v^*,a^*)\,\mathrm{d}v^*\right]\mathrm{d}a^*\,\mathrm{d}a$$

- change of variables $v' = v - v^*$

$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_0^\infty \beta(v',v^*,a-a^*,a^*)n(v',a-a^*)(v'+v^*)^k\,\mathrm{d}v'\,n(v^*,a^*)\,\mathrm{d}v^*\right]\mathrm{d}a^*\,\mathrm{d}a$$



Bivariate Moment Equations

- Aggregation birth term:

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\,\mathrm{d}v\,\mathrm{d}a$$

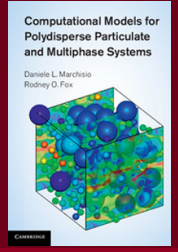
$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_0^v \beta(v-v^*,v^*,a-a^*,a^*)n(v-v^*,a-a^*)n(v^*,a^*)\,\mathrm{d}v^* v^k\,\mathrm{d}v\right]\mathrm{d}a^*\,\mathrm{d}a$$

- change in order of integration

$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_{v^*}^\infty \beta(v-v^*,v^*,a-a^*,a^*)n(v-v^*,a-a^*)v^k\,\mathrm{d}v\, n(v^*,a^*)\,\mathrm{d}v^*\right]\mathrm{d}a^*\,\mathrm{d}a$$

- change of variables $v' = v - v^*$

$$= \frac{1}{2}\int_0^\infty a^l \int_0^a \left[\int_0^\infty \int_0^\infty \beta(v',v^*,a-a^*,a^*)n(v',a-a^*)(v'+v^*)^k\,\mathrm{d}v' n(v^*,a^*)\,\mathrm{d}v^*\right]\mathrm{d}a^*\,\mathrm{d}a$$



Bivariate Moment Equations

- same steps for a

$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v'+v^*)^k (a'+a^*)^l \beta(v',v^*,a',a^*) n(v',a') n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v'\, \mathrm{d}a'$$

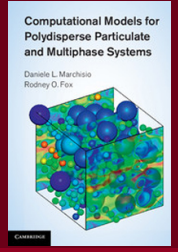
- combining everything yields

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\, \mathrm{d}v\, \mathrm{d}a = \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v+v^*)^k (a+a^*)^l \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

- Aggregation death term: $\beta(v,v^*,a,a^*) = \beta(v^*,v,a^*,a)$

$$\int_0^\infty \int_0^\infty v^k a^l D(v,a)\, \mathrm{d}v\, \mathrm{d}a = \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty \left(v^k a^l + v^{*k} a^{*l}\right) \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

N.B. These results are general for binary aggregation/agglomeration



Bivariate Moment Equations

- same steps for a

$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v'+v^*)^k (a'+a^*)^l \beta(v',v^*,a',a^*) n(v',a') n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v'\, \mathrm{d}a'$$

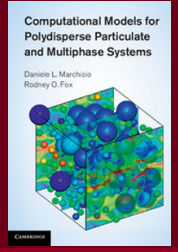
- combining everything yields

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v+v^*)^k (a+a^*)^l \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

- Aggregation death term: $\beta(v,v^*,a,a^*) = \beta(v^*,v,a^*,a)$

$$\int_0^\infty \int_0^\infty v^k a^l D(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty \left(v^k a^l + v^{*k} a^{*l}\right) \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

N.B. These results are general for binary aggregation/agglomeration



Bivariate Moment Equations

- same steps for a

$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v'+v^*)^k (a'+a^*)^l \beta(v',v^*,a',a^*) n(v',a') n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v'\, \mathrm{d}a'$$

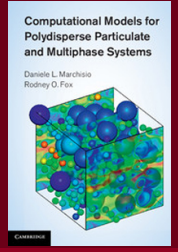
- combining everything yields

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v+v^*)^k (a+a^*)^l \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

- Aggregation death term: $\beta(v,v^*,a,a^*) = \beta(v^*,v,a^*,a)$

$$\int_0^\infty \int_0^\infty v^k a^l D(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty \left(v^k a^l + v^{*k} a^{*l}\right) \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

N.B. These results are general for binary aggregation/agglomeration



Bivariate Moment Equations

- same steps for a

$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v'+v^*)^k (a'+a^*)^l \beta(v',v^*,a',a^*) n(v',a') n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v'\, \mathrm{d}a'$$

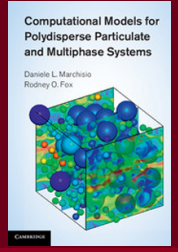
- combining everything yields

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v+v^*)^k (a+a^*)^l \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

- Aggregation death term: $\beta(v,v^*,a,a^*) = \beta(v^*,v,a^*,a)$

$$\int_0^\infty \int_0^\infty v^k a^l D(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty \left(v^k a^l + v^{*k} a^{*l}\right) \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

N.B. These results are general for binary aggregation/agglomeration



Bivariate Moment Equations

- same steps for a

$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v'+v^*)^k (a'+a^*)^l \beta(v',v^*,a',a^*) n(v',a') n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v'\, \mathrm{d}a'$$

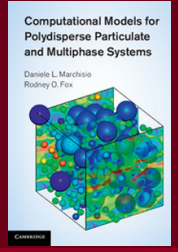
- combining everything yields

$$\int_0^\infty \int_0^\infty v^k a^l B(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty (v+v^*)^k (a+a^*)^l \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

- Aggregation death term: $\beta(v,v^*,a,a^*) = \beta(v^*,v,a^*,a)$

$$\int_0^\infty \int_0^\infty v^k a^l D(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$= \frac{1}{2}\int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty \left(v^k a^l + v^{*k} a^{*l}\right) \beta(v,v^*,a,a^*) n(v,a) n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

N.B. These results are general for binary aggregation/agglomeration



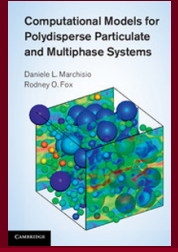
Bivariate Moment Equations

Final result:

$$\frac{\partial m_{k,l}}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{U}_g m_{k,l} =$$
$$J(T,\phi) v_0^k a_0^l + \int_0^\infty \int_0^\infty k v^{k-1} a^l G_v(v,a)\, n(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$+ \int_0^\infty \int_0^\infty l v^k a^{l-1} \left[G_a(v,a) + S_a(v,a)\right] n(v,a)\, \mathrm{d}v\, \mathrm{d}a$$
$$+ \frac{1}{2} \int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty \left[(v+v^*)^k (a+a^*)^l - v^k a^l - v^{*k} a^{*l}\right]$$
$$\beta(v,v^*,a,a^*)\, n(v,a)\, n(v^*,a^*)\, \mathrm{d}v^*\, \mathrm{d}a^*\, \mathrm{d}v\, \mathrm{d}a$$

where $\mathbf{U}_g$ is gas-phase velocity from flow code

QBMM are needed to close the RHS



Application of CQMOM to Moments

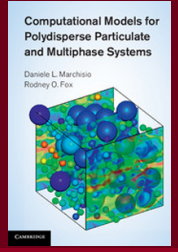
Reconstructed NDF: $N_v = N_a = 3$

$$n^*(v,a) = \sum_{i=1}^{N_v} \sum_{j=1}^{N_a} w_{i,j}\delta(v - v_i)\delta(a - a_{i,j})$$

CQMOM approximation for 21 moments:

$$m_{k,l} = \sum_{i=1}^{N_v} \sum_{j=1}^{N_a} w_{i,j} v_i^k a_{i,j}^l$$

$$\begin{array}{cccccc} m_{0,0} & m_{0,1} & m_{0,2} & m_{0,3} & m_{0,4} & m_{0,5} \\ m_{1,0} & m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} \\ m_{2,0} & m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} \\ m_{3,0} & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ m_{4,0} & \langle a \rangle_1 & \langle a^2 \rangle_1 & \langle a^3 \rangle_1 & \langle a^4 \rangle_1 & \langle a^5 \rangle_1 \\ m_{5,0} & \langle a \rangle_2 & \langle a^2 \rangle_2 & \langle a^3 \rangle_2 & \langle a^4 \rangle_2 & \langle a^5 \rangle_2 \\ \downarrow & \langle a \rangle_3 & \langle a^2 \rangle_3 & \langle a^3 \rangle_3 & \langle a^4 \rangle_3 & \langle a^5 \rangle_3 \\ \{w_i, v_i\} & & & & & \end{array} \quad \begin{array}{l} \\ \\ \\ \\ \rightarrow \{w_{1,j}, a_{1,j}\} \\ \rightarrow \{w_{2,j}, a_{2,j}\} \\ \rightarrow \{w_{3,j}, a_{3,j}\} \end{array}$$



Application of CQMOM to Moments

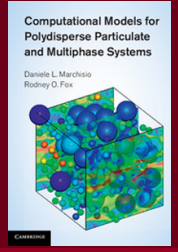
Reconstructed NDF: $N_v = N_a = 3$

$$n^*(v,a) = \sum_{i=1}^{N_v} \sum_{j=1}^{N_a} w_{i,j}\delta(v - v_i)\delta(a - a_{i,j})$$

CQMOM approximation for 21 moments:

$$m_{k,l} = \sum_{i=1}^{N_v} \sum_{j=1}^{N_a} w_{i,j} v_i^k a_{i,j}^l$$

$m_{0,0}$	$m_{0,1}$	$m_{0,2}$	$m_{0,3}$	$m_{0,4}$	$m_{0,5}$	
$m_{1,0}$	$m_{1,1}$	$m_{1,2}$	$m_{1,3}$	$m_{1,4}$	$m_{1,5}$	
$m_{2,0}$	$m_{2,1}$	$m_{2,2}$	$m_{2,3}$	$m_{2,4}$	$m_{2,5}$	
$m_{3,0}$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
$m_{4,0}$	$\langle a \rangle_1$	$\langle a^2 \rangle_1$	$\langle a^3 \rangle_1$	$\langle a^4 \rangle_1$	$\langle a^5 \rangle_1$	$\rightarrow \{w_{1,j}, a_{1,j}\}$
$m_{5,0}$	$\langle a \rangle_2$	$\langle a^2 \rangle_2$	$\langle a^3 \rangle_2$	$\langle a^4 \rangle_2$	$\langle a^5 \rangle_2$	$\rightarrow \{w_{2,j}, a_{2,j}\}$
$\downarrow$	$\langle a \rangle_3$	$\langle a^2 \rangle_3$	$\langle a^3 \rangle_3$	$\langle a^4 \rangle_3$	$\langle a^5 \rangle_3$	$\rightarrow \{w_{3,j}, a_{3,j}\}$
$\{w_i, v_i\}$						



Application of CQMOM to Transport Equation

Final result:

$$\frac{\partial m_{k,l}}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{U}_g m_{k,l} = J(T,\phi) v_0^k a_0^l \qquad \text{nucleation}$$

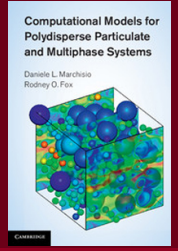
$$+ \sum_{i=1}^{N_v} \sum_{j=1}^{N_a} w_{i,j} k v_i^{k-1} a_{i,j}^l G_v(v_i, a_{i,j}) \qquad \text{volume change}$$

$$+ \sum_{i=1}^{N_v} \sum_{j=1}^{N_a} w_{i,j} l v_i^k a_{i,j}^{l-1} \left[G_a(v_i, a_{i,j}) + S_a(v_i, a_{i,j}) \right] \qquad \text{area change}$$

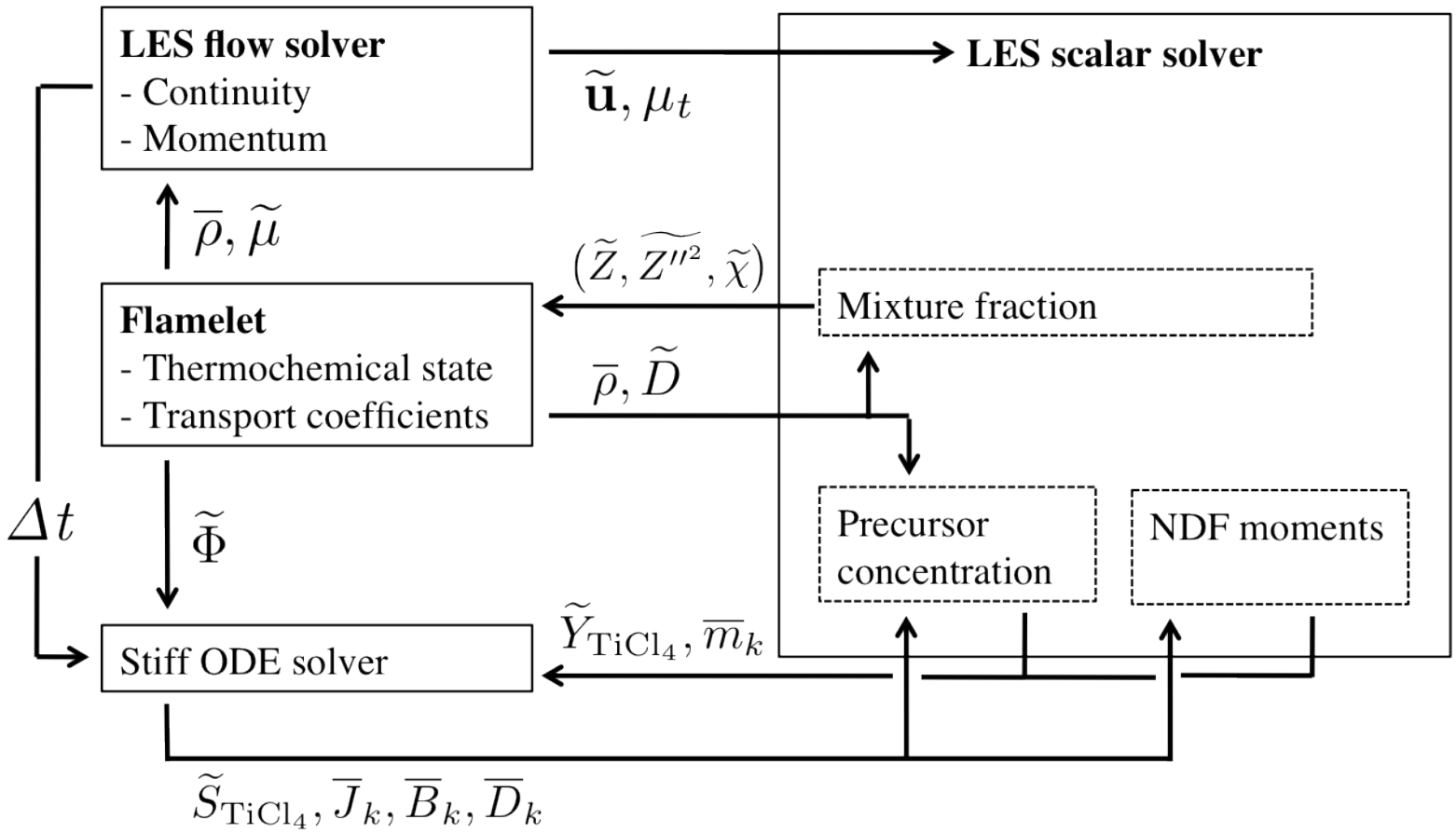
$$+ \frac{1}{2} \sum_{i=1}^{N_v} \sum_{j=1}^{N_a} \sum_{m=1}^{N_v} \sum_{n=1}^{N_a} w_{i,j} w_{m,n} \qquad \text{aggregation}$$

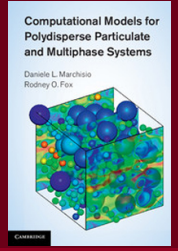
$$\left[(v_i + v_m)^k (a_{i,j} + a_{m,n})^l - v_i^k a_{i,j}^l - v_m^k a_{m,n}^l \right] \beta(v_i, v_m, a_{i,j}, a_{m,n})$$

RHS is now closed

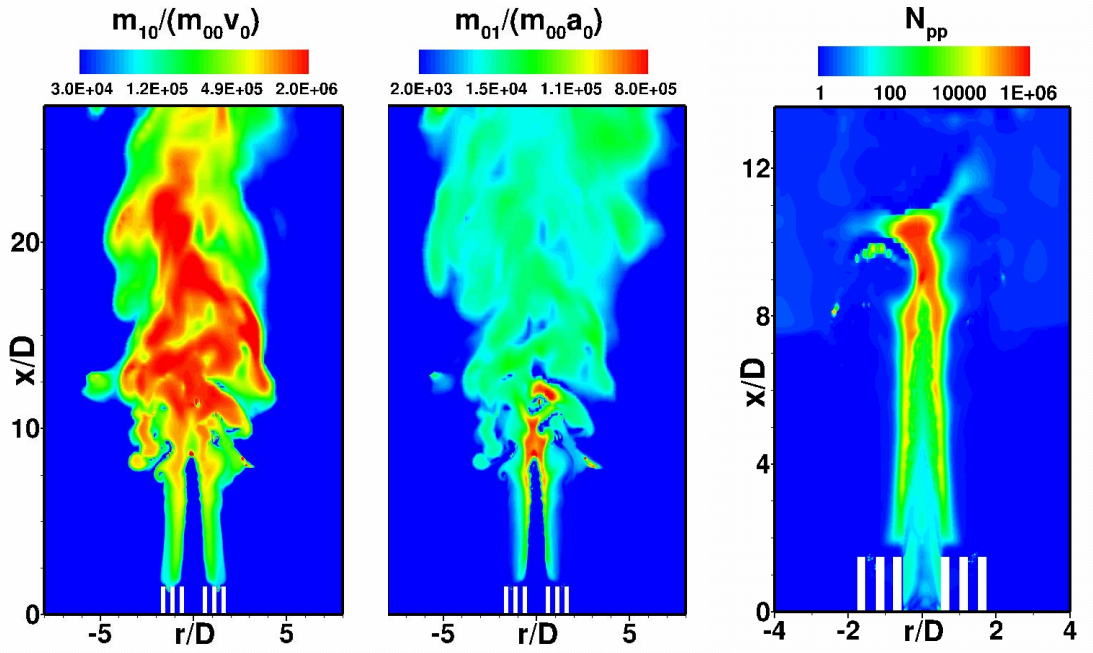


Implementation in LES code

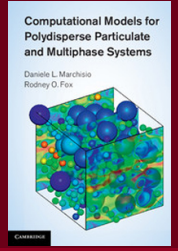




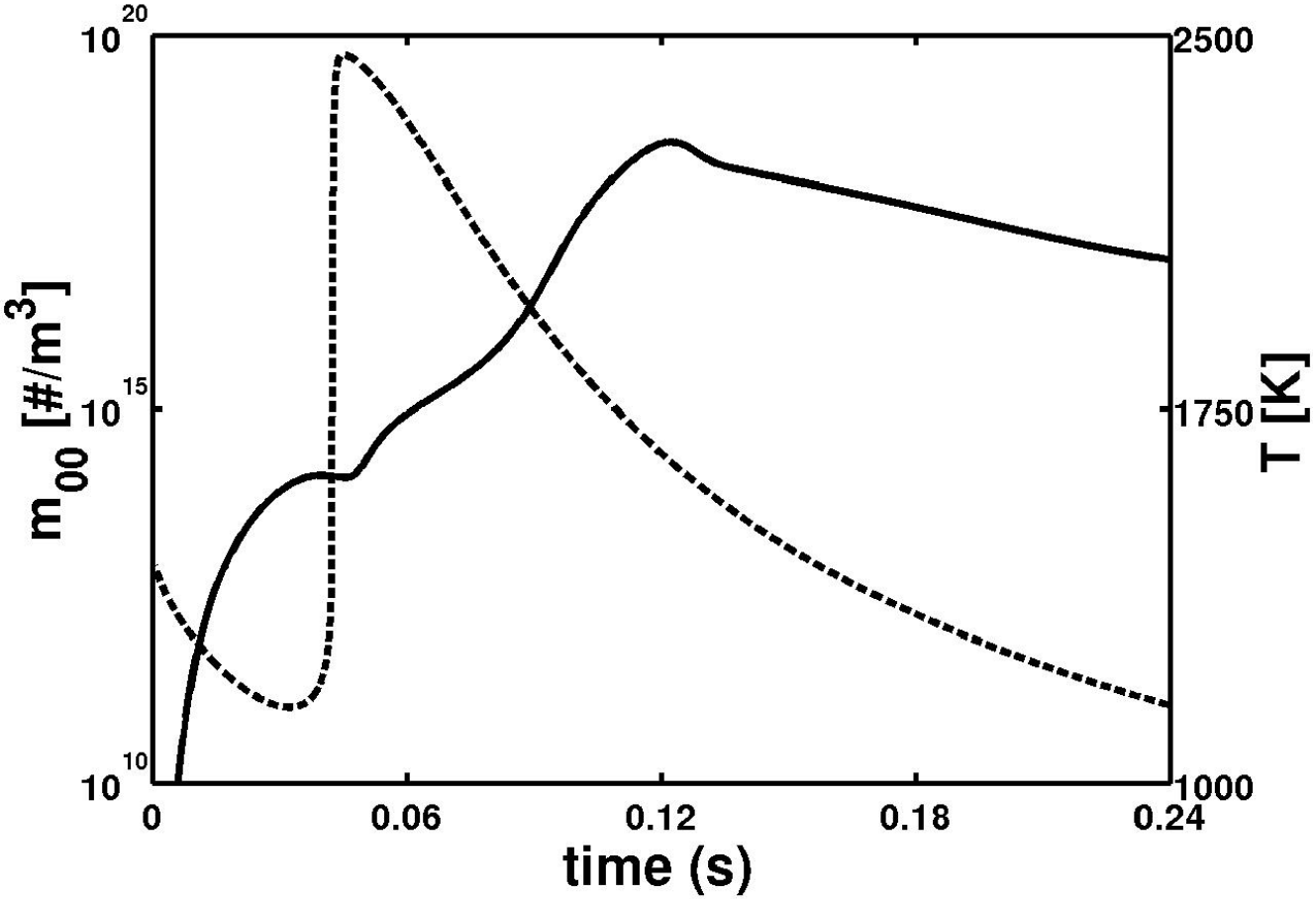
LES Results

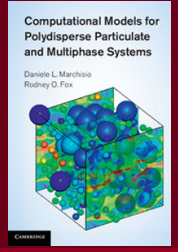


- m_{10} mean particle volume
- m_{01} mean particle surface area
- N_{pp} primary particles (nuclei)

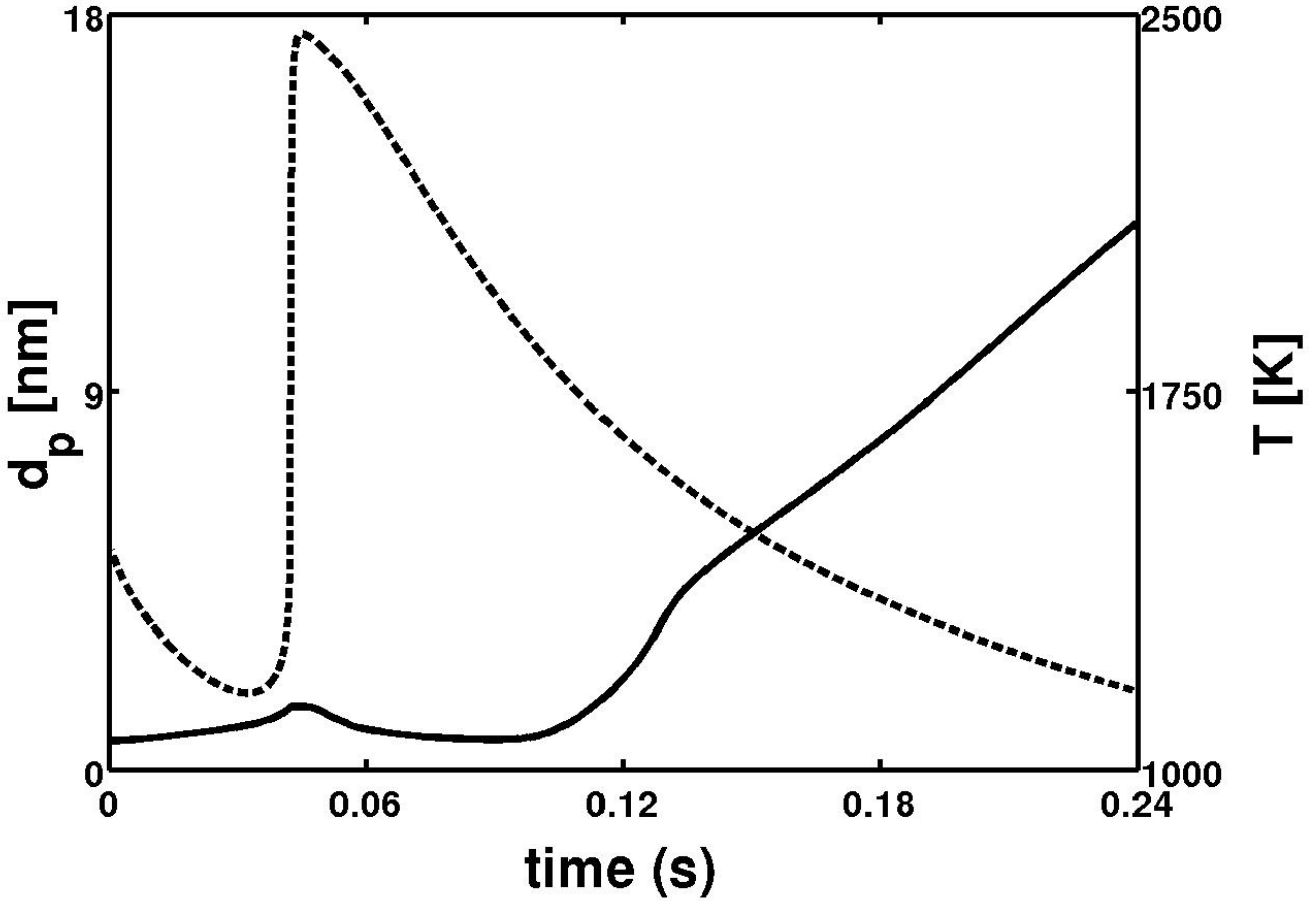


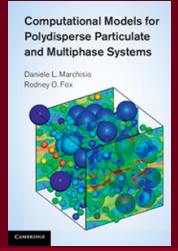
Number concentration vs. residence time



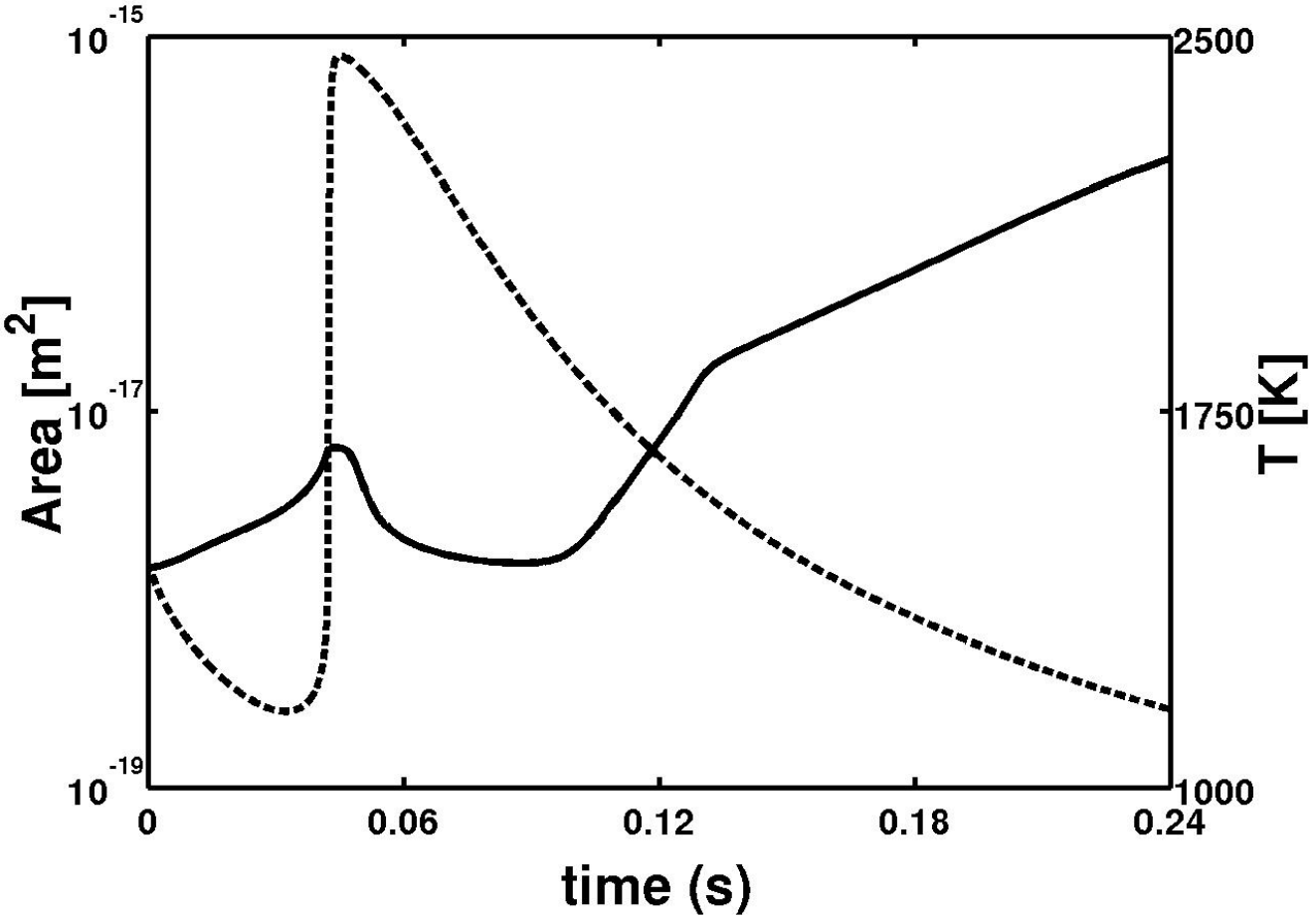


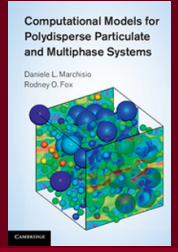
d_p vs. residence time





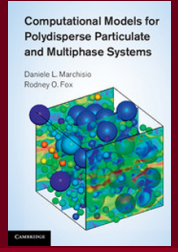
Surface area vs. residence time





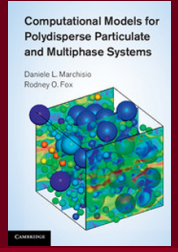
Summary for Fine-Particle Applications

- Complex flow physics
  - Turbulence in non-canonical geometries
  - Coupling with turbulent mixing and combustion
  - Heat transfer, acoustics, compressible flow
- Complex chemistry
  - Non-equilibrium chemistry (rich, lean)
  - Premixed/nonpremixed/partially premixed combustion
  - Fine particles (‘slow’ chemistry, high Sc, radiation)
- Fine particle dynamics
  - Particle size/chemical composition distribution
  - Nucleation, oxidation, surface growth
  - Aggregation, coagulation, sintering



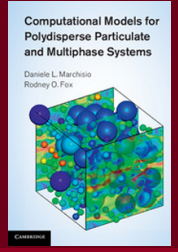
Summary for Fine-Particle Applications

- Complex flow physics
  - Turbulence in non-canonical geometries
  - Coupling with turbulent mixing and combustion
  - Heat transfer, acoustics, compressible flow
- Complex chemistry
  - Non-equilibrium chemistry (rich, lean)
  - Premixed/nonpremixed/partially premixed combustion
  - Fine particles (‘slow’ chemistry, high Sc, radiation)
- Fine particle dynamics
  - Particle size/chemical composition distribution
  - Nucleation, oxidation, surface growth
  - Aggregation, coagulation, sintering

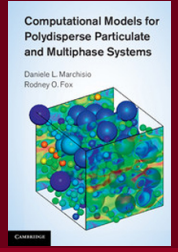


Summary for Fine-Particle Applications

- Complex flow physics
  - Turbulence in non-canonical geometries
  - Coupling with turbulent mixing and combustion
  - Heat transfer, acoustics, compressible flow
- Complex chemistry
  - Non-equilibrium chemistry (rich, lean)
  - Premixed/nonpremixed/partially premixed combustion
  - Fine particles (‘slow’ chemistry, high Sc, radiation)
- Fine particle dynamics
  - Particle size/chemical composition distribution
  - Nucleation, oxidation, surface growth
  - Aggregation, coagulation, sintering



APPLICATION TO BUBBLY FLOWS



Bubble velocity distribution

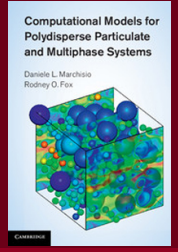
- Due to $\rho_g \ll \rho_l$, bubble have very small Stokes numbers
- In GPBE, bubble velocity is monokinetic:

$$n(\mathbf{v}_p, \xi) = n(\xi)\, \delta\left(\mathbf{v}_p - \mathbf{U}_p(\xi)\right)$$

- In words, bubble velocity depends only on "size" and velocity fluctuations for fixed "size" are negligible
- However, many forces in addition to drag are needed

Moments

- Size moments needed to reconstruct $n(\xi)$
- Joint size-velocity moments $\langle \xi^k \mathbf{v}_p \rangle$ needed to approximate $\mathbf{U}_p(\xi)$



QBMM for Bubble Flows

Monokinetic assumption

Bubble velocity distribution

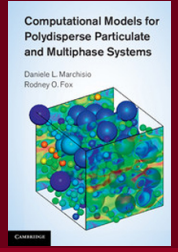
- Due to $\rho_g \ll \rho_l$, bubble have very small Stokes numbers
- In GPBE, bubble velocity is monokinetic:

$$n(\mathbf{v}_p, \xi) = n(\xi)\,\delta\left(\mathbf{v}_p - \mathbf{U}_p(\xi)\right)$$

- In words, bubble velocity depends only on "size" and velocity fluctuations for fixed "size" are negligible
- However, many forces in addition to drag are needed

Moments

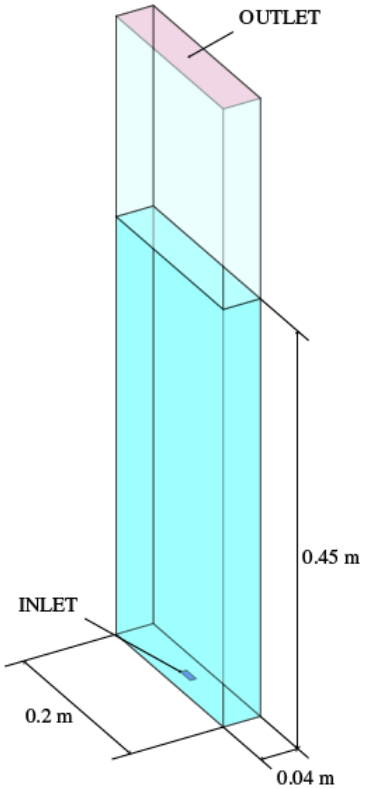
- Size moments needed to reconstruct $n(\xi)$
- Joint size-velocity moments $\langle \xi^k \mathbf{v}_p \rangle$ needed to approximate $\mathbf{U}_p(\xi)$

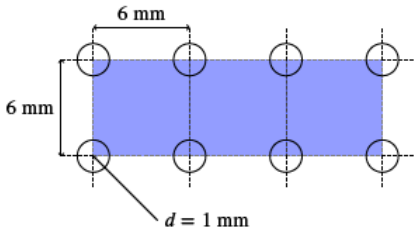


QBMM for Bubble Columns

Geometry description

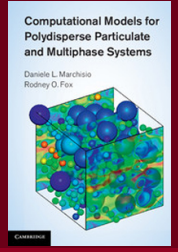
Rectangular bubble column





- Contaminated air–water
- Superficial gas velocity: 2.4–21.3 m/s
- Exp. measur.: gas hold-up, plume oscillation period (POP), mean Sauter diameter

Diaz, M.E., Montes, F.J., Galan, M.A., 2008. Chemical Engineering and Processing 47, 1867-1876.

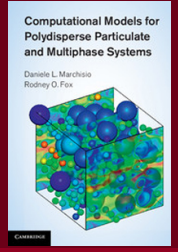


QBMM for Bubble Columns

Computational details for OpenFOAM

- QMOM and CQMOM implemented in OpenFOAM solver `compressibleTwoPhaseEulerFoam` (two-phase Eulerian-Eulerian equations) with compressibility effects disabled (isothermal system)
- Drag coefficient calculated through terminal velocity (20 cm/s) and other drag correlations but other interfacial forces neglected
- Standard k–ε model for liquid turbulence
- PBM solved with QMOM for four moments (M_0, M_1, M_2, M_3) corresponding to a two-node quadrature and CQMOM for nine moments ($M_{0,0}$, $M_{1,0}$, $M_{2,0}$, $M_{3,0}$, $M_{4,0}$, $M_{5,0}$, $M_{0,1}$, $M_{1,1}$, $M_{2,1}$) corresponding to a three-node quadrature
- Adaptive first-order Euler time discretization scheme with Δt chosen so that CFL condition is respected
- Inlet condition for gas bubbles: $\alpha_G = 0.5$; inlet gas velocity to match gassing rate; lognormal BSD centered on mean size[2]

[2] Geary, N.W., Rice, R.G., 1991. AIChE J. 37, 161-168.



QBMM for Bubble Columns

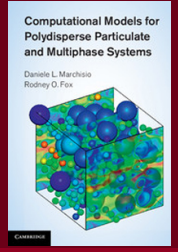
Computational details for OpenFOAM

Numerical schemes and boundary conditions

Variable	Scheme	Inlet	Outlet	Wall
Gas vol. frac.	Limited Second Order Upwind	0.5	Gradient zero	Gradient zero
Gas velocity	Limited Second Order Upwind	Dep. on flow rate	Gradient zero with backflow	Free-slip wall
Liquid velocity	Limited Second Order Upwind	0.0 m/s	Gradient zero	No-slip wall
Pressure	First Order Upwind	Gradient zero	1 bar	Gradient zero
k	Limited Second Order Upwind	Based on turb. Int. = 5%	$1 \cdot 10^{-4}\,\mathrm{m}^2\mathrm{s}^{-2}$ (backflow only)	Gradient zero
ϵ	Limited Second Order Upwind	length scale equal to the hole diam.	$1 \cdot 10^{-5}\,\mathrm{m}^2\mathrm{s}^{-3}$ (backflow only)	Gradient zero
Moments	First Order Upwind	Log-normal distrib.	Gradient zero	Gradient zero

$$M_k = M_0 \exp\left\{k\mu + \frac{k^2\sigma^2}{2}\right\}, \mu = \log\left(\frac{m^2}{\sqrt{v+m^2}}\right) \sigma = \sqrt{\log\left(\frac{v}{m^2}+1\right)}$$

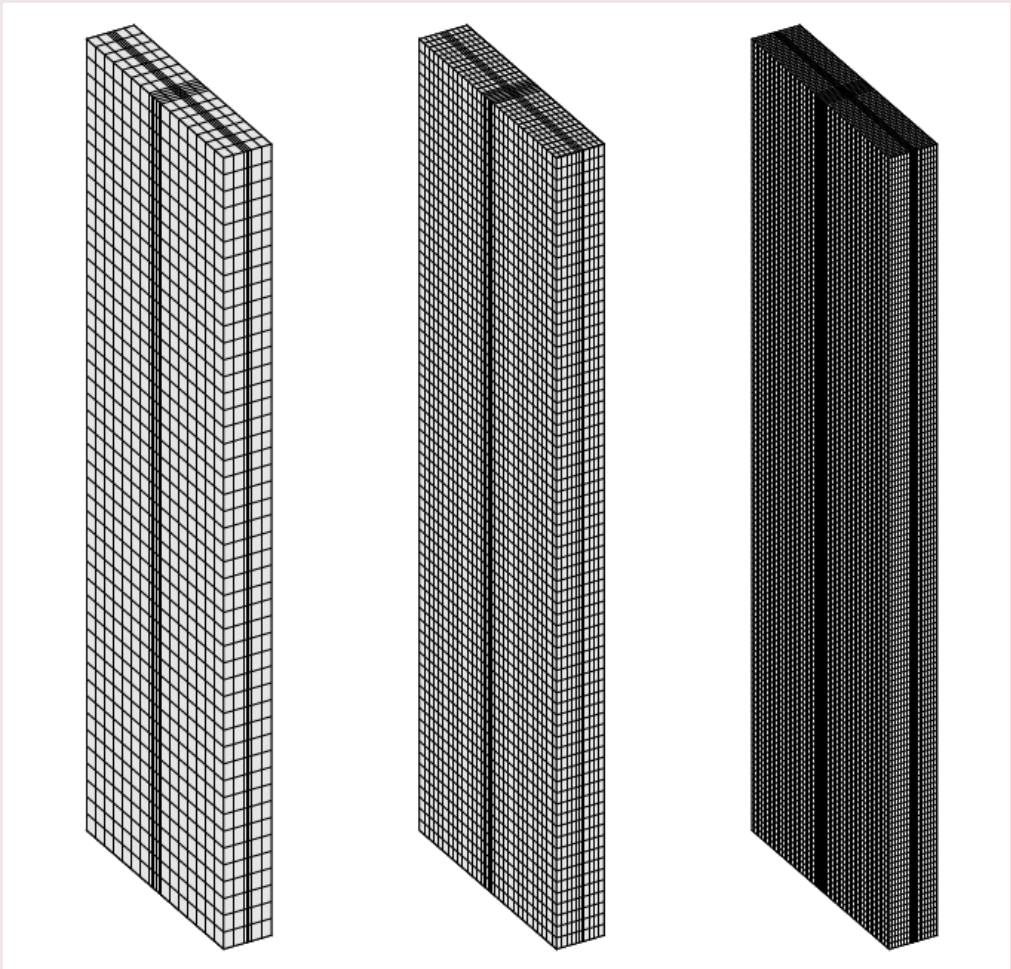
$$\text{Gas vol. frac.} = k_V M_3 = k_V M_0 \exp\left\{3\mu + \frac{9\sigma^2}{2}\right\}$$



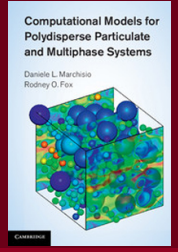
QBMM for Bubble Columns

Computational details for OpenFOAM

Computational grids

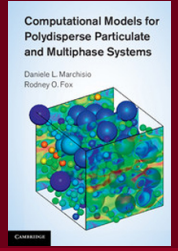


1. coarse grid (17 width $\times$ 7 depth $\times$ 45 height)
2. medium grid (32 width $\times$ 11 depth $\times$ 70 height)
3. fine grid (62 width $\times$ 19 depth $\times$ 128 height)



Mesh size effect on calculated gas hold-up, Plume Oscillation Period and Sauter Diameter compared with experiments for $U_G = 2.4$ mm/s

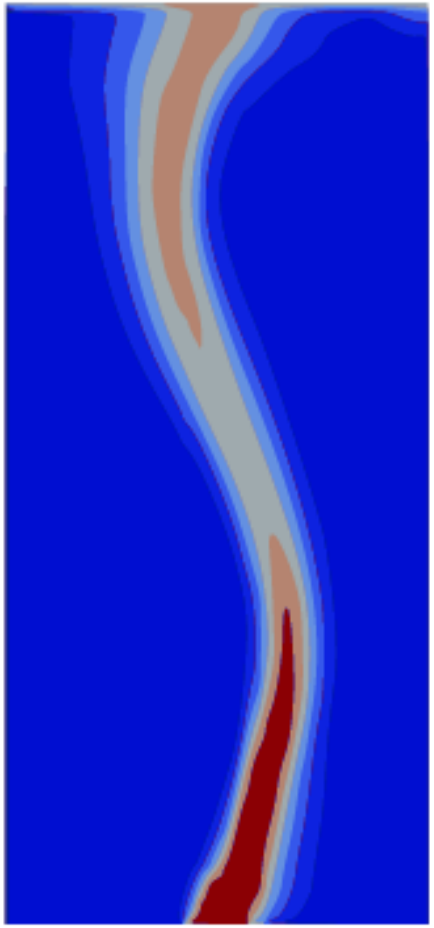
Grid size	**Num.**	**Hold-up**	**POP** (s)	d_{32} (mm)
Coarse	5355	0.64%	7.78	5.78
Medium	24640	0.62%	10.57	6.92
Fine	150784	0.57%	11.71	5.93
Experiment		0.69%	11.37	6.83

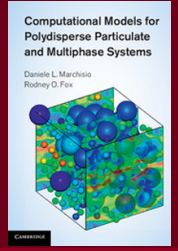


QBMM for Bubble Columns

flow field and gas profile at 2.4 mm/s

Experiment	Gas. vol. frac.	Liq. vel. field

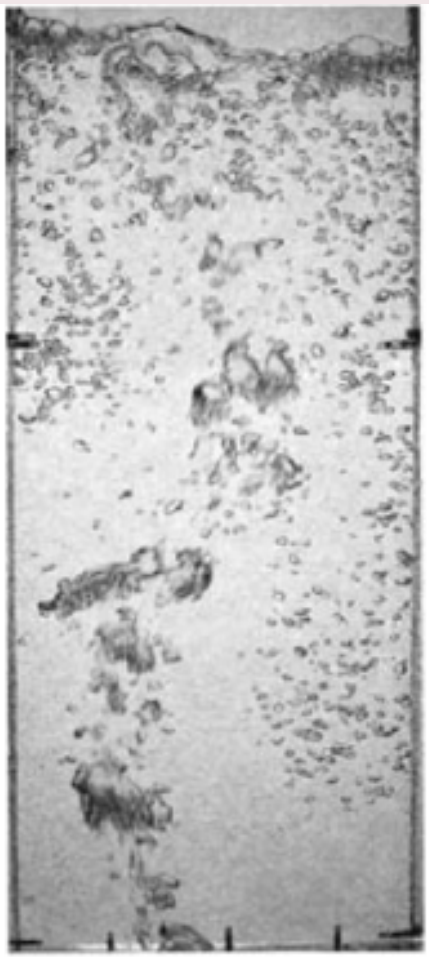


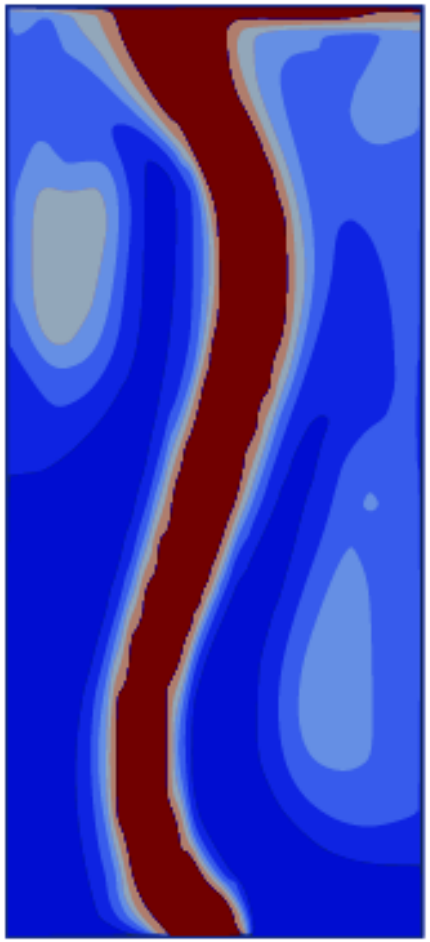


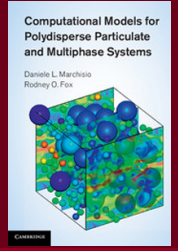
QBMM for Bubble Columns

Flow field and gas profile at 11.9 mm/s

Experiment	Gas. vol. frac.	Liq. vel. field



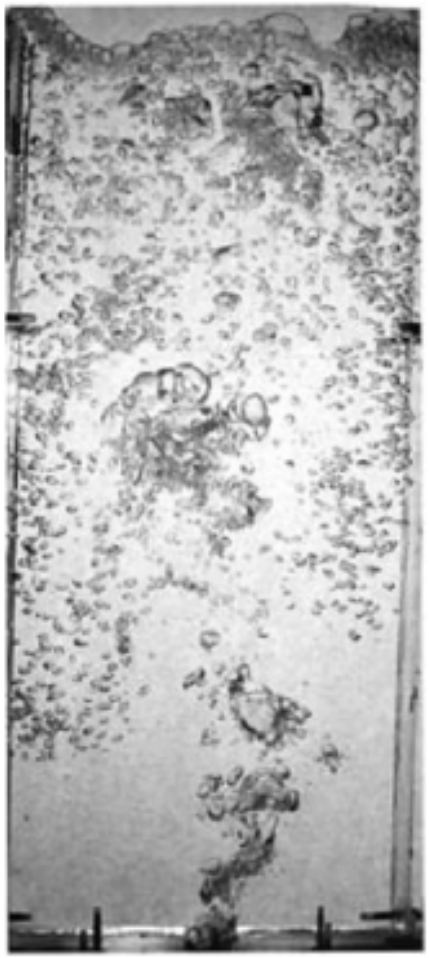


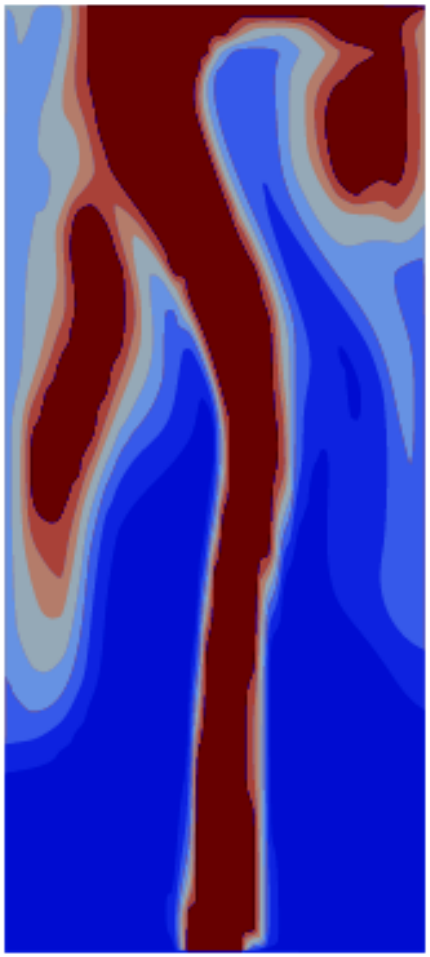


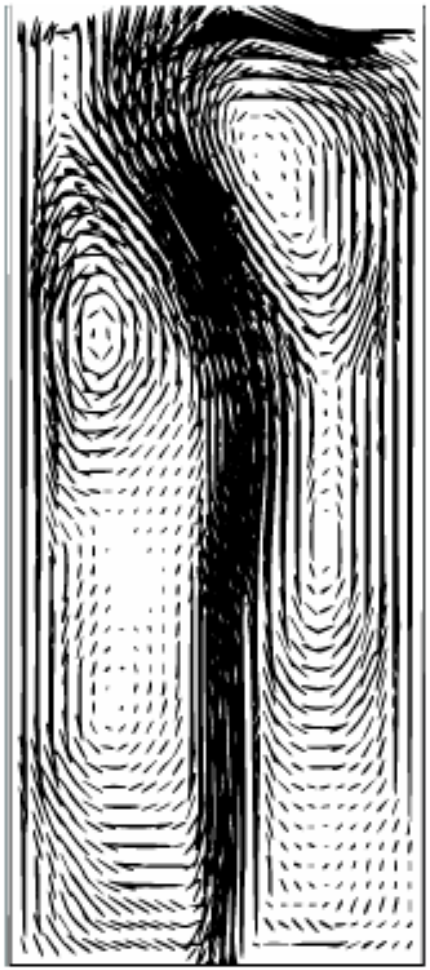
QBMM for Bubble Columns

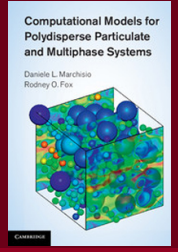
Flow field and gas profile at 21.3 mm/s

Experiment	Gas. vol. frac.	Liq. vel. field





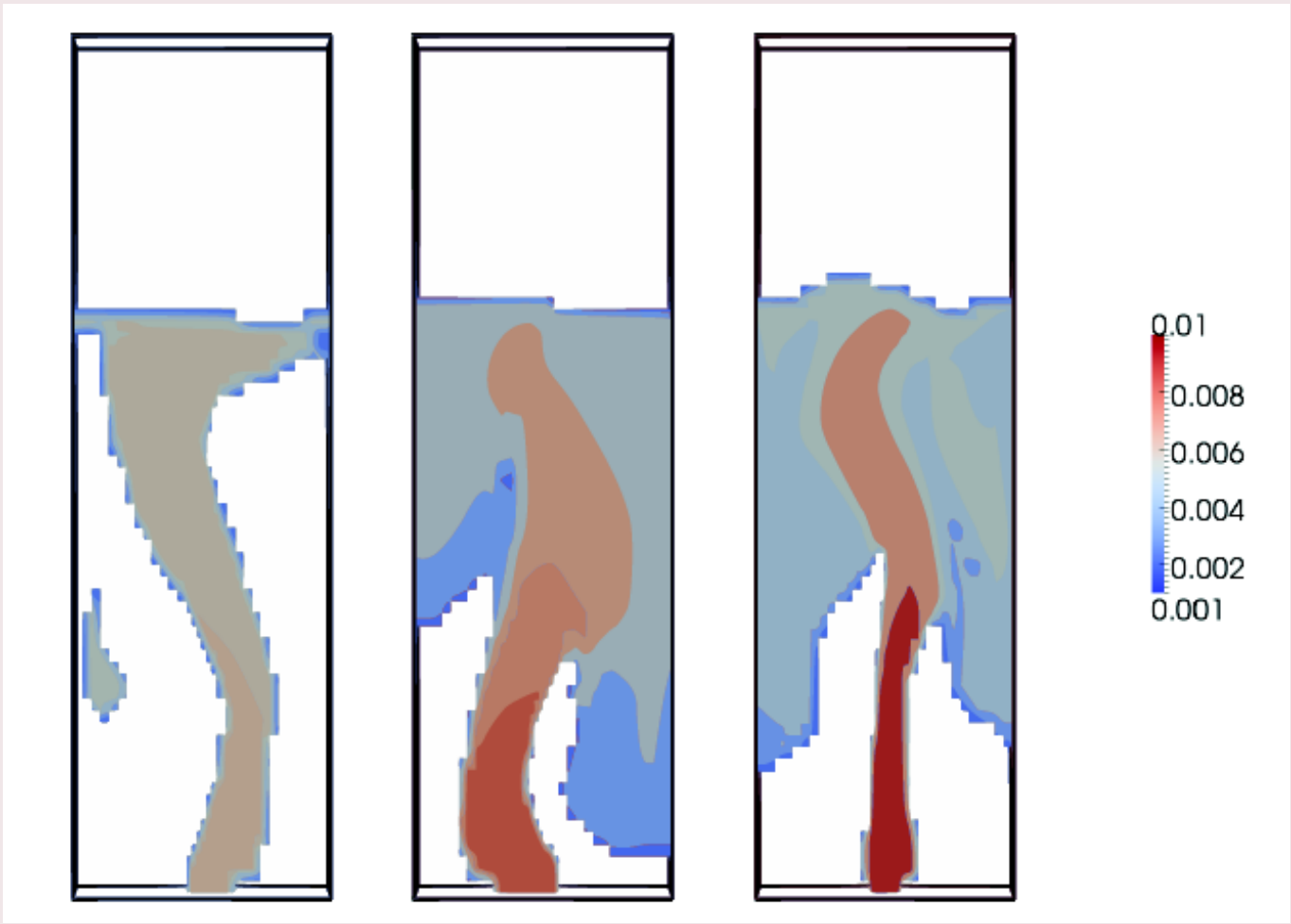


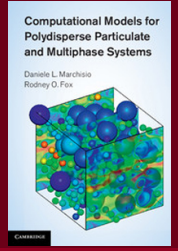


Results QBMM

Mean bubble size

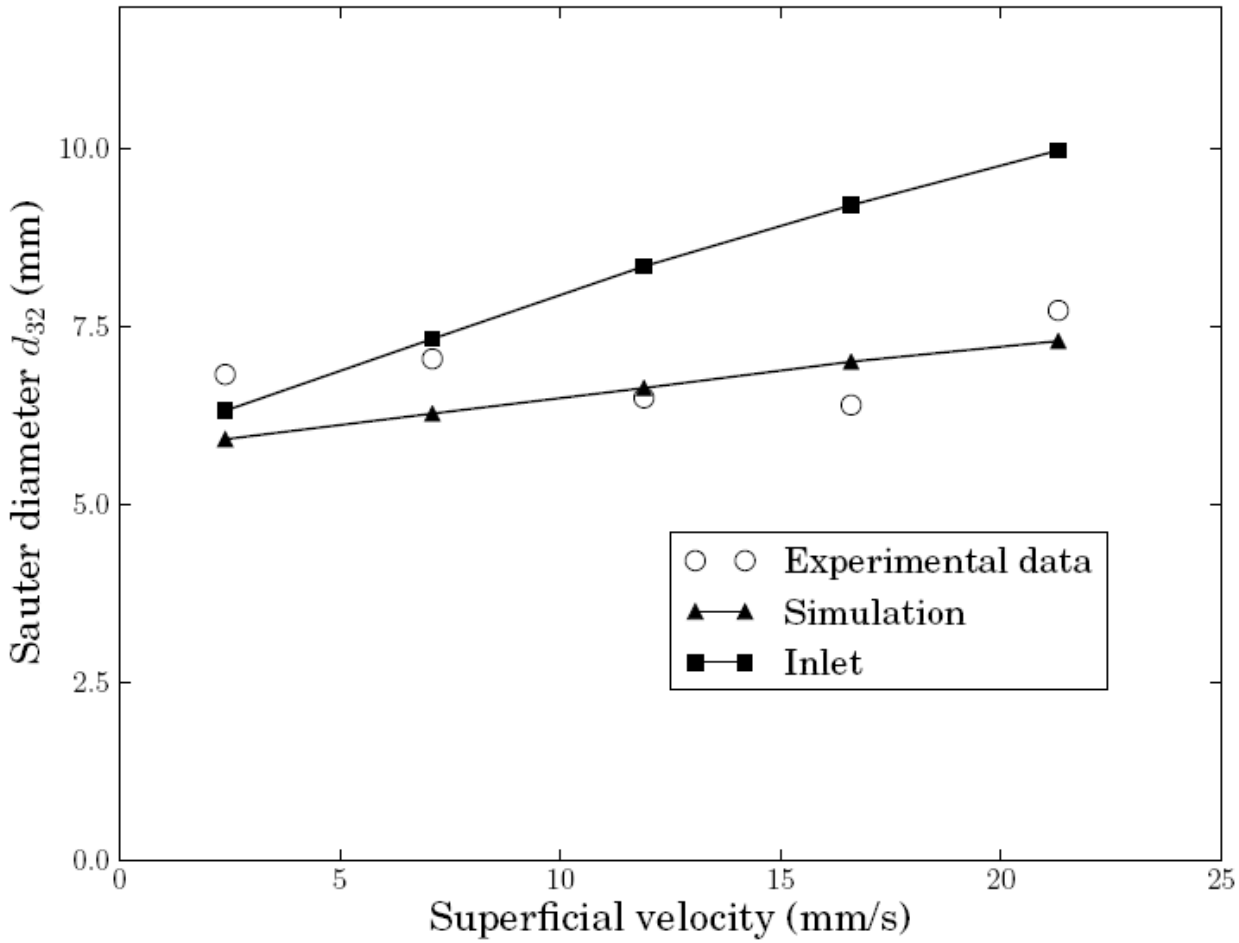
2.4 mm/s 11.9 mm/s 21.3 mm/s (m)

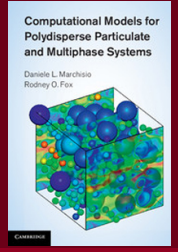




QBMM for Bubble Columns

Mean bubble size

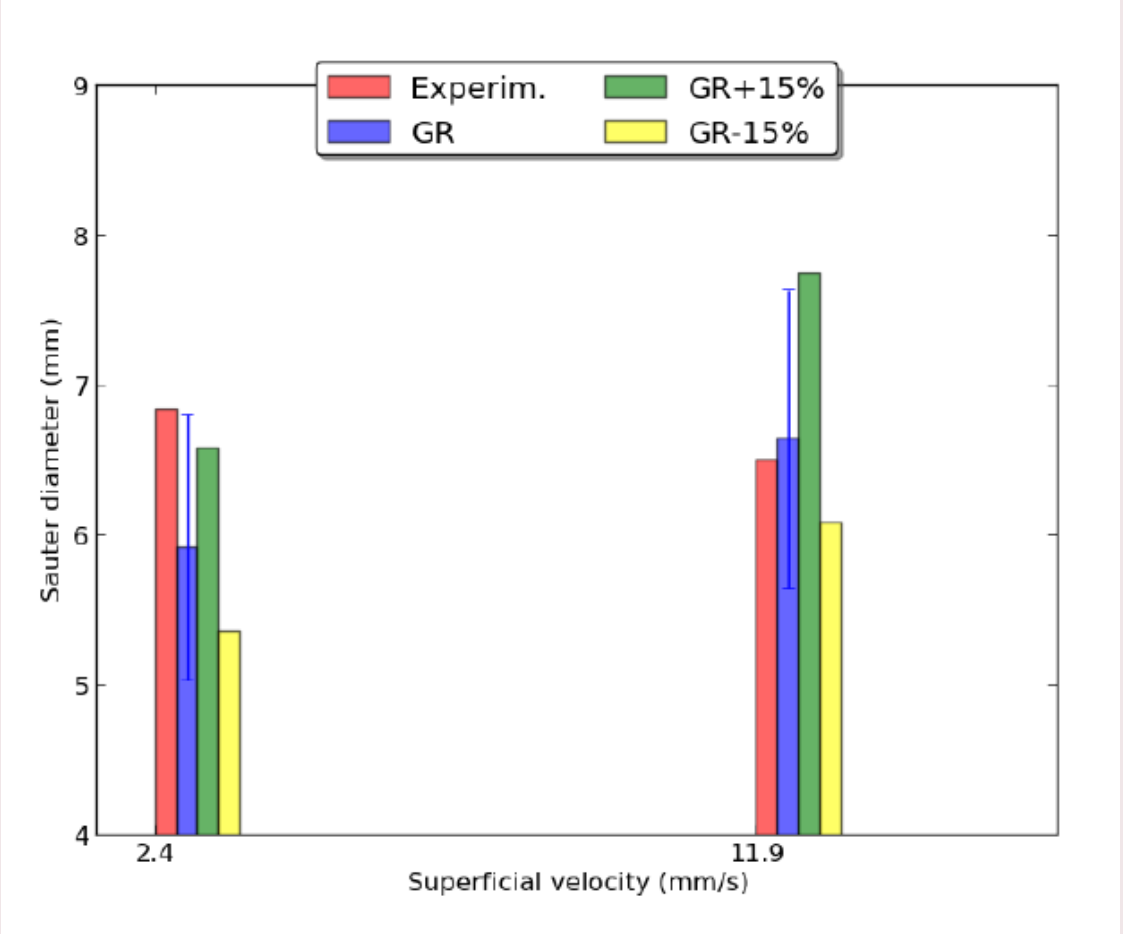


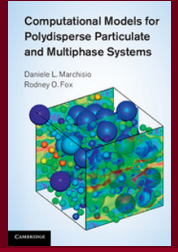


QBMM for Bubble Columns

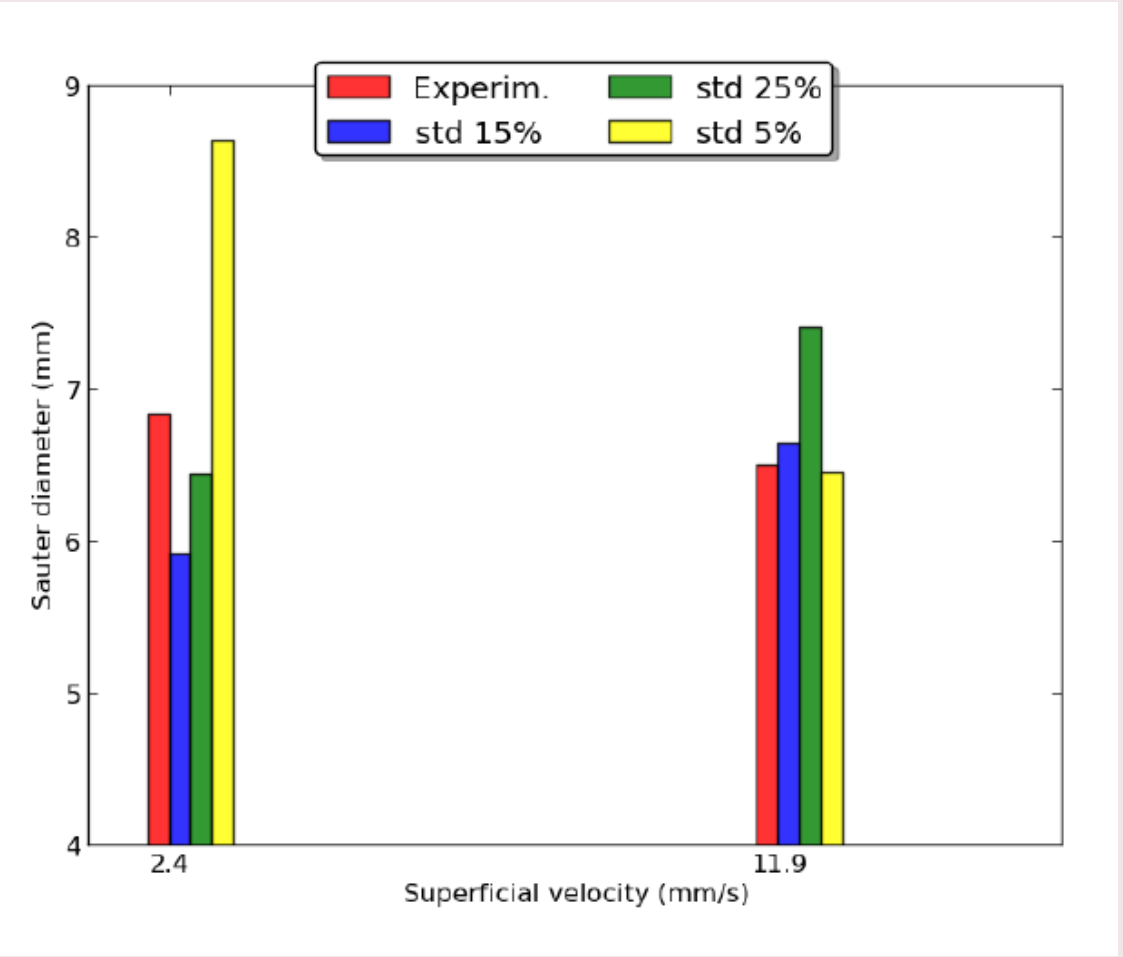
Mean bubble size

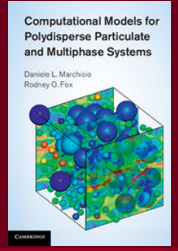
Effect of inlet mean bubble size





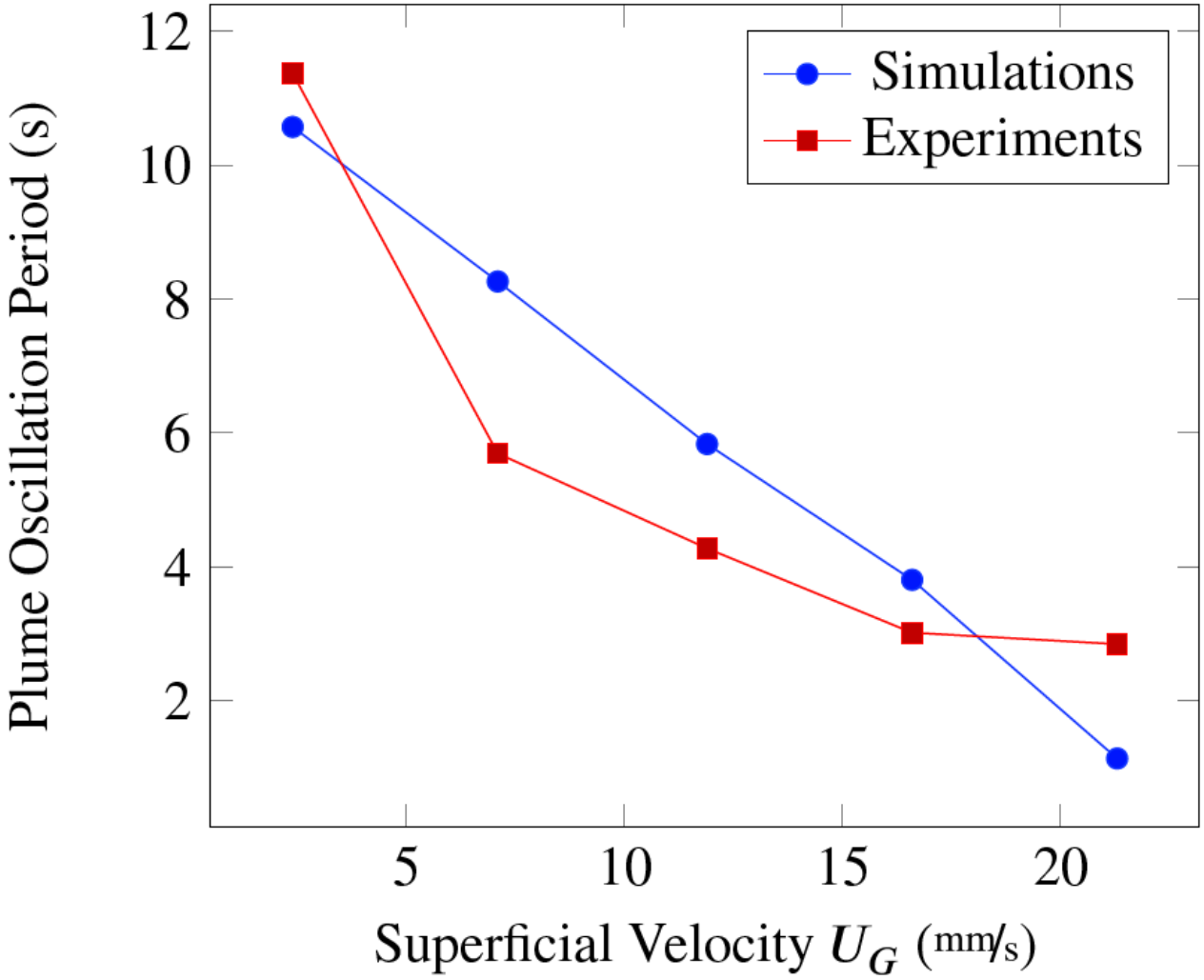
Effect of inlet lognormal std deviation

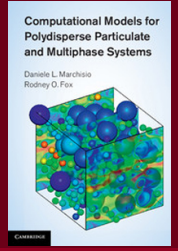




QBMM for Bubble Columns

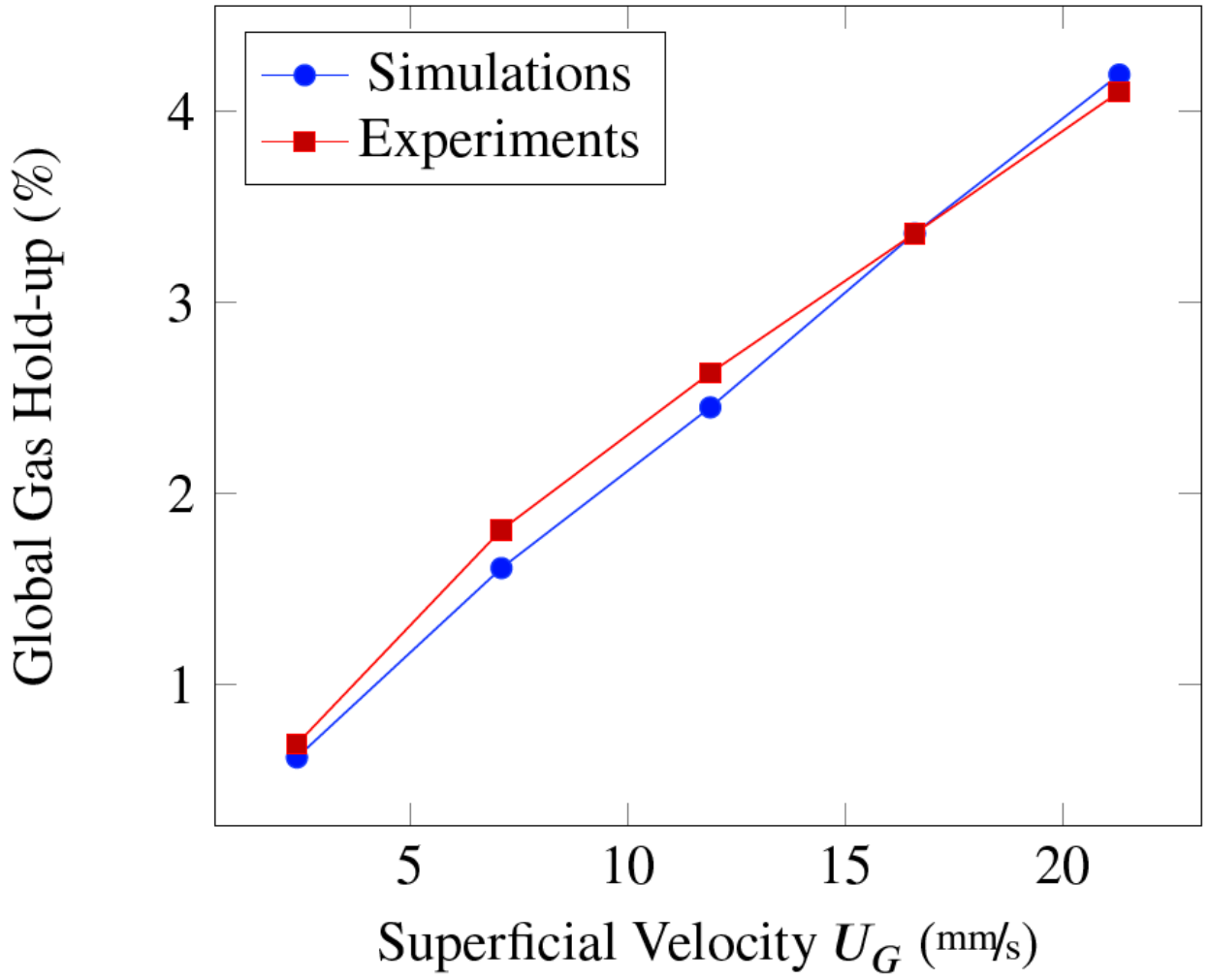
Plume oscillation period

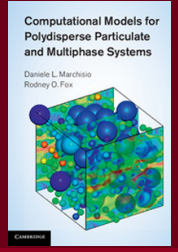




QBMM for Bubble Columns

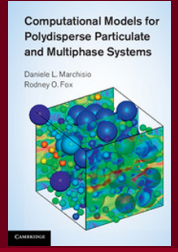
Gas hold-up





Comparison of experimental data and calculated results for global gas volume fraction (hold-up), POP and mean Sauter diameter varying the superficial gas velocity U_G

Superf. vel. (mm/s)	**Hold-up**	**POP** (s)	d_{32} (mm)
2.4	0.62%	10.10	6.01
Exp.	0.69%	11.37	6.83
7.1	1.61%	8.26	6.28
Exp.	1.81%	5.69	7.05
11.9	2.45%	5.83	6.89
Exp.	2.63%	4.27	6.50
16.6	3.36%	3.80	7.01
Exp.	3.36%	3.01	6.40
21.3	4.19%	3.84	7.96
Exp.	4.10%	2.84	7.73

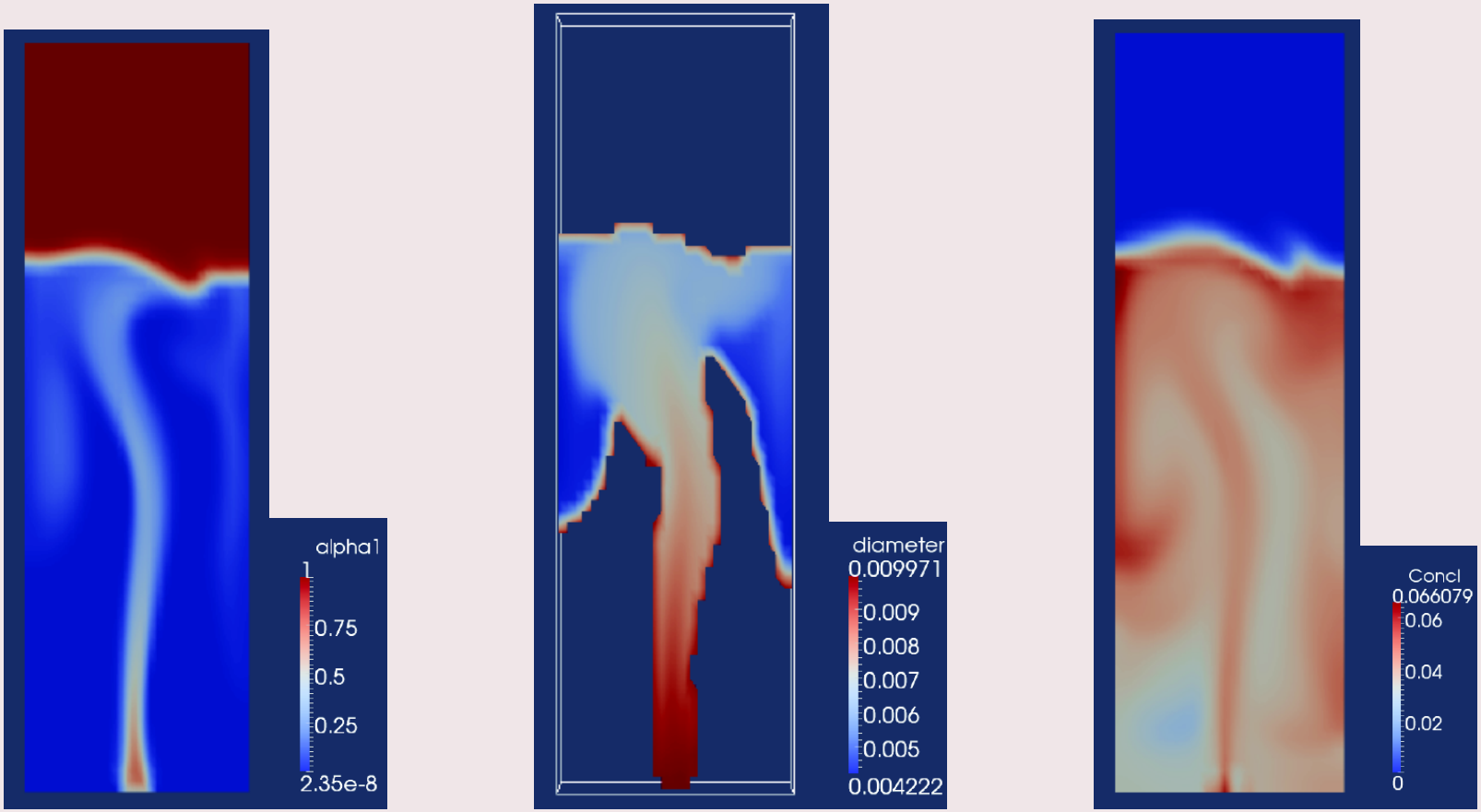


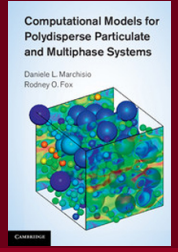
QBMM for Bubble Columns

CQMOM implementation for oxygen transfer

$U_G = 21.3$ mm/s ($t = 1.5$ s)

Gas vol. frac. / Sauter d. (m) / liq. ox. conc. (mol/m^3)



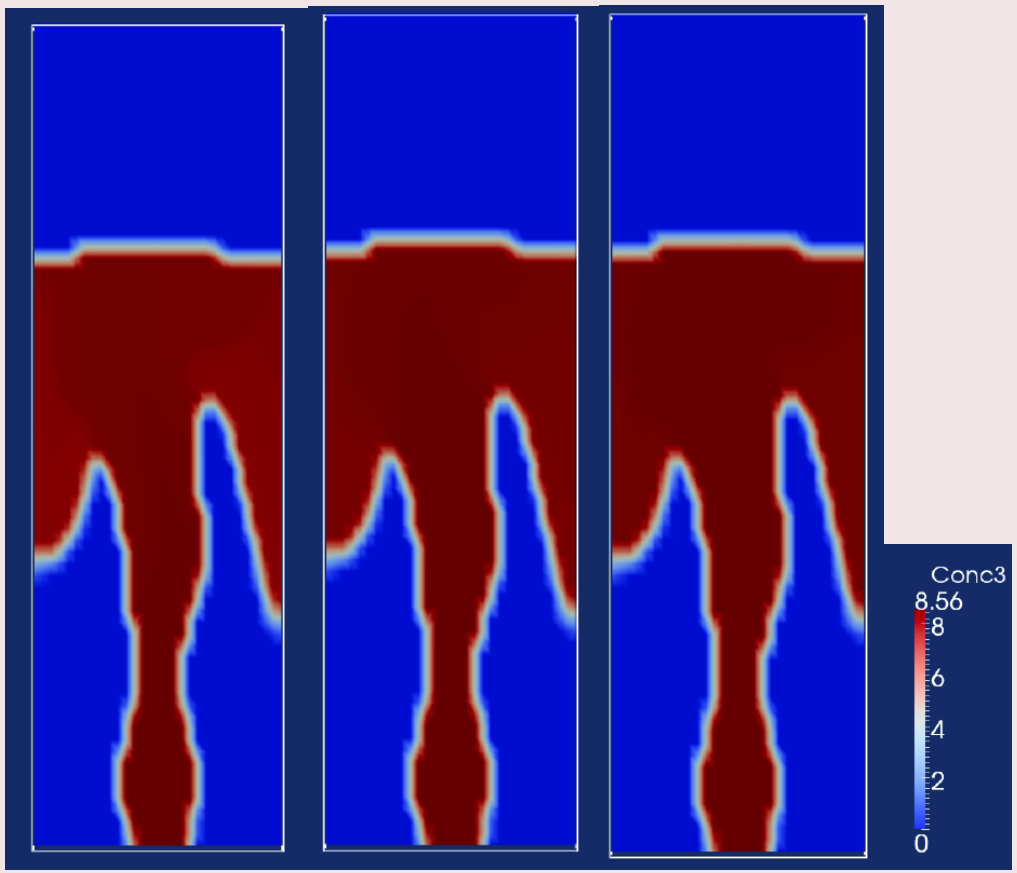


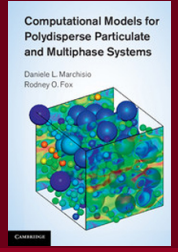
QBMM for Bubble Columns

CQMOM implementation for oxygen transfer

$U_G = 21.3$ mm/s ($t = 1.5$ s)

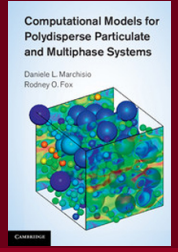
gas ox. conc. (mol/m^3) / liq. ox. conc. (mol/m^3)





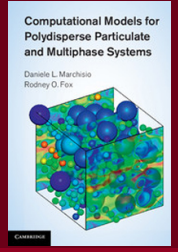
Coalescence and Breakage

- Consider a batch homogeneous GAS–LIQUID system (i.e. one isolated cell of computational domain)
- Gas bubbles are characterized by SIZE (L) and by composition through MOLES (ϕ) of active component
- Bubbles undergo ONLY coalescence and breakup (no mass transfer)
- Initial condition: bivariate (in size and composition) Gaussian distribution
- There is no flux of bubbles through cell
- Evolution of NDF is predicted with DSMC



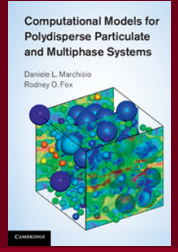
Coalescence and Breakage

- Consider a batch homogeneous GAS–LIQUID system (i.e. one isolated cell of computational domain)
- Gas bubbles are characterized by SIZE (L) and by composition through MOLES (ϕ) of active component
- Bubbles undergo ONLY coalescence and breakup (no mass transfer)
- Initial condition: bivariate (in size and composition) Gaussian distribution
- There is no flux of bubbles through cell
- Evolution of NDF is predicted with DSMC



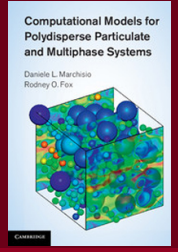
Coalescence and Breakage

- Consider a batch homogeneous GAS–LIQUID system (i.e. one isolated cell of computational domain)
- Gas bubbles are characterized by SIZE (L) and by composition through MOLES (ϕ) of active component
- Bubbles undergo ONLY coalescence and breakup (no mass transfer)
- Initial condition: bivariate (in size and composition) Gaussian distribution
- There is no flux of bubbles through cell
- Evolution of NDF is predicted with DSMC



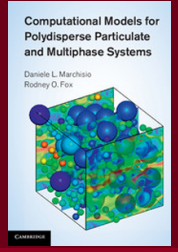
Coalescence and Breakage

- Consider a batch homogeneous GAS–LIQUID system (i.e. one isolated cell of computational domain)
- Gas bubbles are characterized by SIZE (L) and by composition through MOLES (ϕ) of active component
- Bubbles undergo ONLY coalescence and breakup (no mass transfer)
- Initial condition: bivariate (in size and composition) Gaussian distribution
- There is no flux of bubbles through cell
- Evolution of NDF is predicted with DSMC



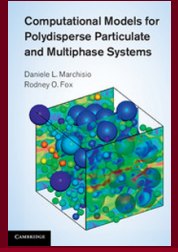
Coalescence and Breakage

- Consider a batch homogeneous GAS–LIQUID system (i.e. one isolated cell of computational domain)
- Gas bubbles are characterized by SIZE (L) and by composition through MOLES (ϕ) of active component
- Bubbles undergo ONLY coalescence and breakup (no mass transfer)
- Initial condition: bivariate (in size and composition) Gaussian distribution
- There is no flux of bubbles through cell
- Evolution of NDF is predicted with DSMC



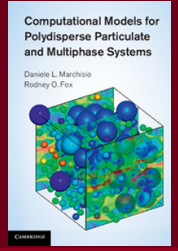
Coalescence and Breakage

- Consider a batch homogeneous GAS–LIQUID system (i.e. one isolated cell of computational domain)
- Gas bubbles are characterized by SIZE (L) and by composition through MOLES (ϕ) of active component
- Bubbles undergo ONLY coalescence and breakup (no mass transfer)
- Initial condition: bivariate (in size and composition) Gaussian distribution
- There is no flux of bubbles through cell
- Evolution of NDF is predicted with DSMC



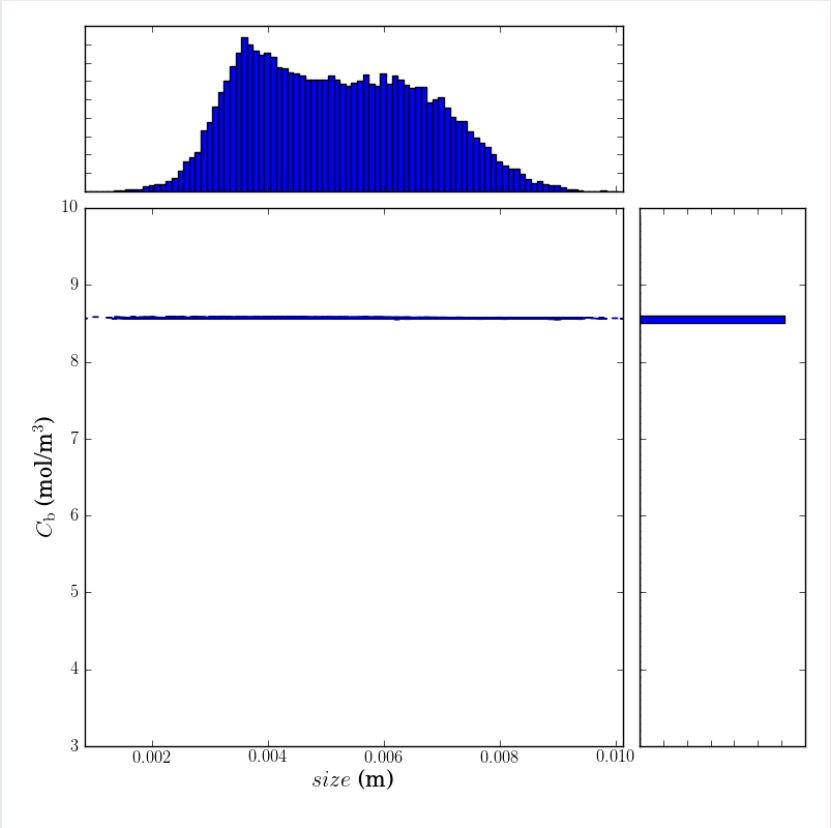
Coalescence and Breakage

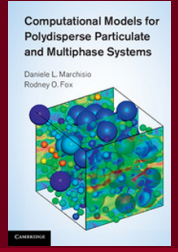
- Consider a batch homogeneous GAS–LIQUID system (i.e. one isolated cell of computational domain)
- Gas bubbles are characterized by SIZE (L) and by composition through MOLES (ϕ) of active component
- Bubbles undergo ONLY coalescence and breakup (no mass transfer)
- Initial condition: bivariate (in size and composition) Gaussian distribution
- There is no flux of bubbles through cell
- Evolution of NDF is predicted with DSMC



Coalescence and Breakage

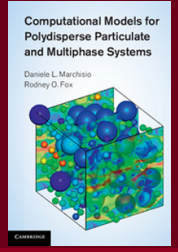
NDF at steady state for bubble size and concentration





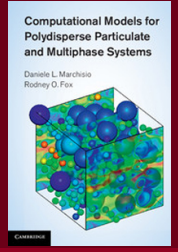
Coalescence and Breakage, Mass Transfer, Chemical Reaction and Convection

- Consider one single cell of finite volume code
- Bubble coalescence, breakup, mass transfer, chemical reaction in liquid phase
- Initial condition: bubbles with same concentration and lognormal bubble size distribution
- Constant flux of bubbles enters cell with NDF corresponding to initial condition
- Evolution of NDF is predicted with DSMC



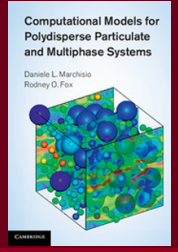
Coalescence and Breakage, Mass Transfer, Chemical Reaction and Convection

- Consider one single cell of finite volume code
- Bubble coalescence, breakup, mass transfer, chemical reaction in liquid phase
- Initial condition: bubbles with same concentration and lognormal bubble size distribution
- Constant flux of bubbles enters cell with NDF corresponding to initial condition
- Evolution of NDF is predicted with DSMC



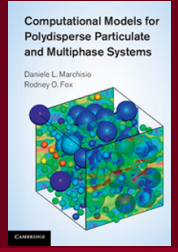
Coalescence and Breakage, Mass Transfer, Chemical Reaction and Convection

- Consider one single cell of finite volume code
- Bubble coalescence, breakup, mass transfer, chemical reaction in liquid phase
- Initial condition: bubbles with same concentration and lognormal bubble size distribution
- Constant flux of bubbles enters cell with NDF corresponding to initial condition
- Evolution of NDF is predicted with DSMC



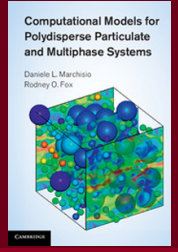
Coalescence and Breakage, Mass Transfer, Chemical Reaction and Convection

- Consider one single cell of finite volume code
- Bubble coalescence, breakup, mass transfer, chemical reaction in liquid phase
- Initial condition: bubbles with same concentration and lognormal bubble size distribution
- Constant flux of bubbles enters cell with NDF corresponding to initial condition
- Evolution of NDF is predicted with DSMC



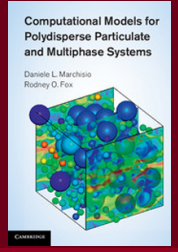
Coalescence and Breakage, Mass Transfer, Chemical Reaction and Convection

- Consider one single cell of finite volume code
- Bubble coalescence, breakup, mass transfer, chemical reaction in liquid phase
- Initial condition: bubbles with same concentration and lognormal bubble size distribution
- Constant flux of bubbles enters cell with NDF corresponding to initial condition
- Evolution of NDF is predicted with DSMC



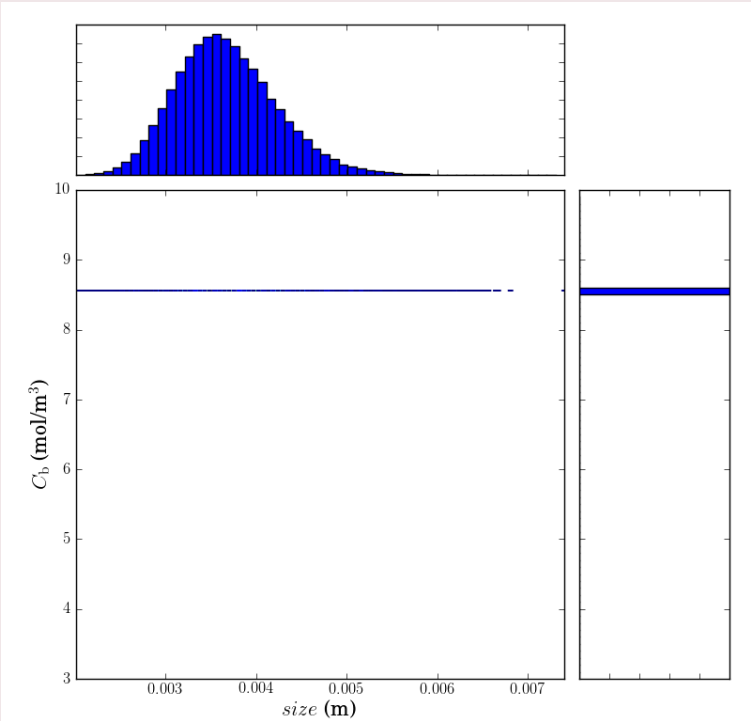
Coalescence and Breakage, Mass Transfer, Chemical Reaction and Convection

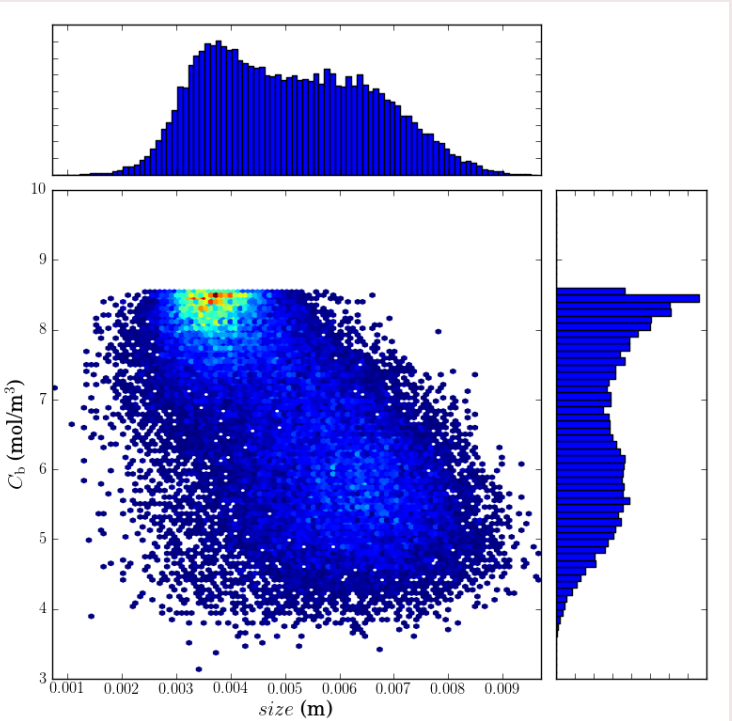
- Consider one single cell of finite volume code
- Bubble coalescence, breakup, mass transfer, chemical reaction in liquid phase
- Initial condition: bubbles with same concentration and lognormal bubble size distribution
- Constant flux of bubbles enters cell with NDF corresponding to initial condition
- Evolution of NDF is predicted with DSMC

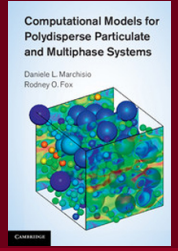


Coalescence and Breakage, Mass Transfer, Chemical Reaction and Convection

Initial NDF and at steady state for bubble size and concentration

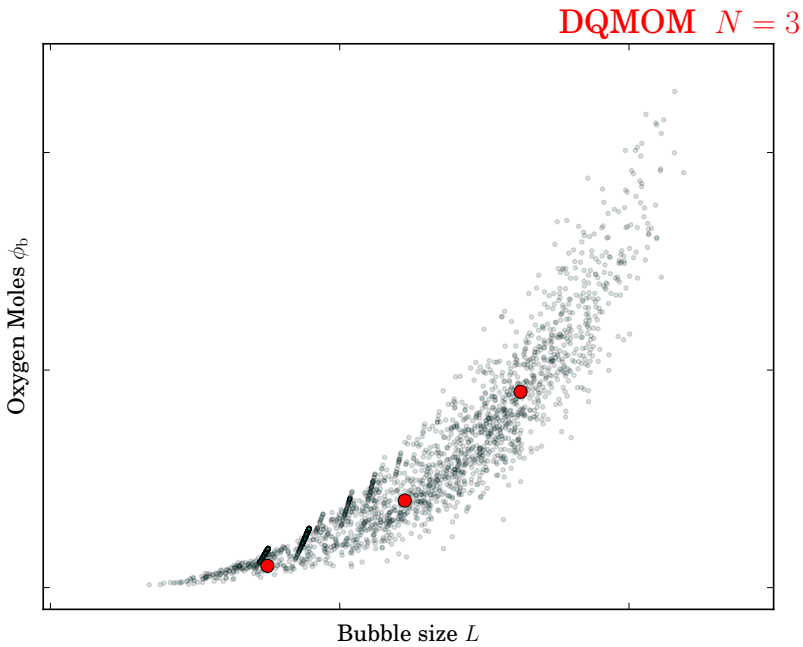




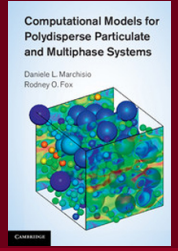


Comparison DSMC with QBMM

DQMOM with $N = 3$ tracking 9 moments of NDF

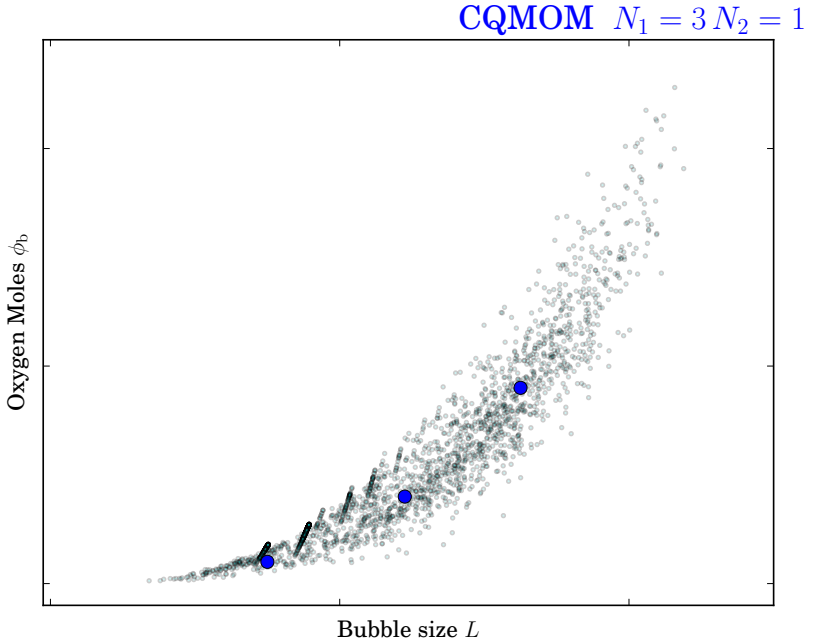


$$\begin{matrix} M_{0,1} & M_{1,1} & M_{2,1} & & & \\ M_{0,0} & M_{1,0} & M_{2,0} & M_{3,0} & M_{4,0} & M_{5,0} \end{matrix}$$

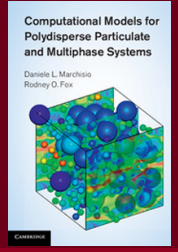


Comparison DSMC with QBMM

CQMOM with $N_1 = 3$ and $N_2 = 1$ tracking 9 moments of NDF

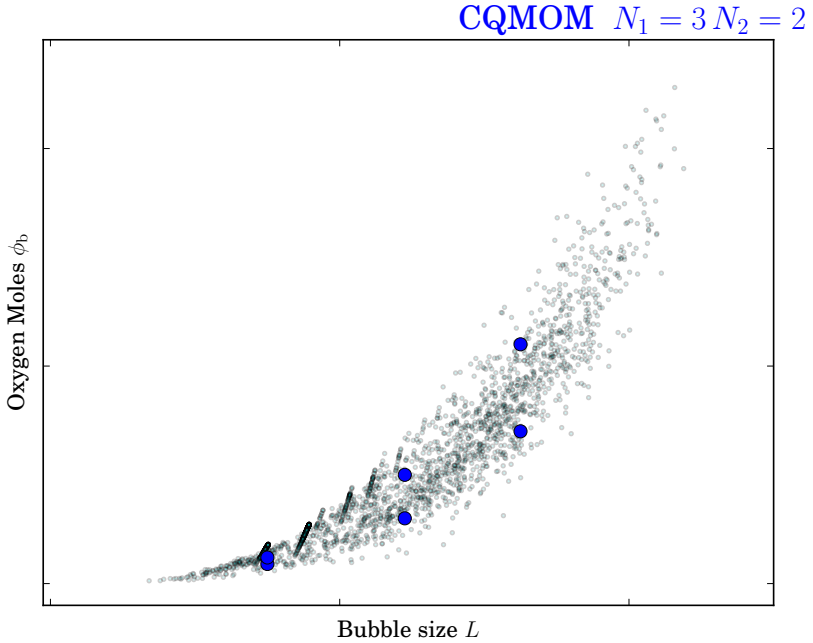


$$M_{0,1} \quad M_{1,1} \quad M_{2,1}$$
$$M_{0,0} \quad M_{1,0} \quad M_{2,0} \quad M_{3,0} \quad M_{4,0} \quad M_{5,0}$$

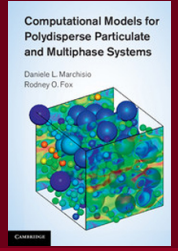


Comparison DSMC with QBMM

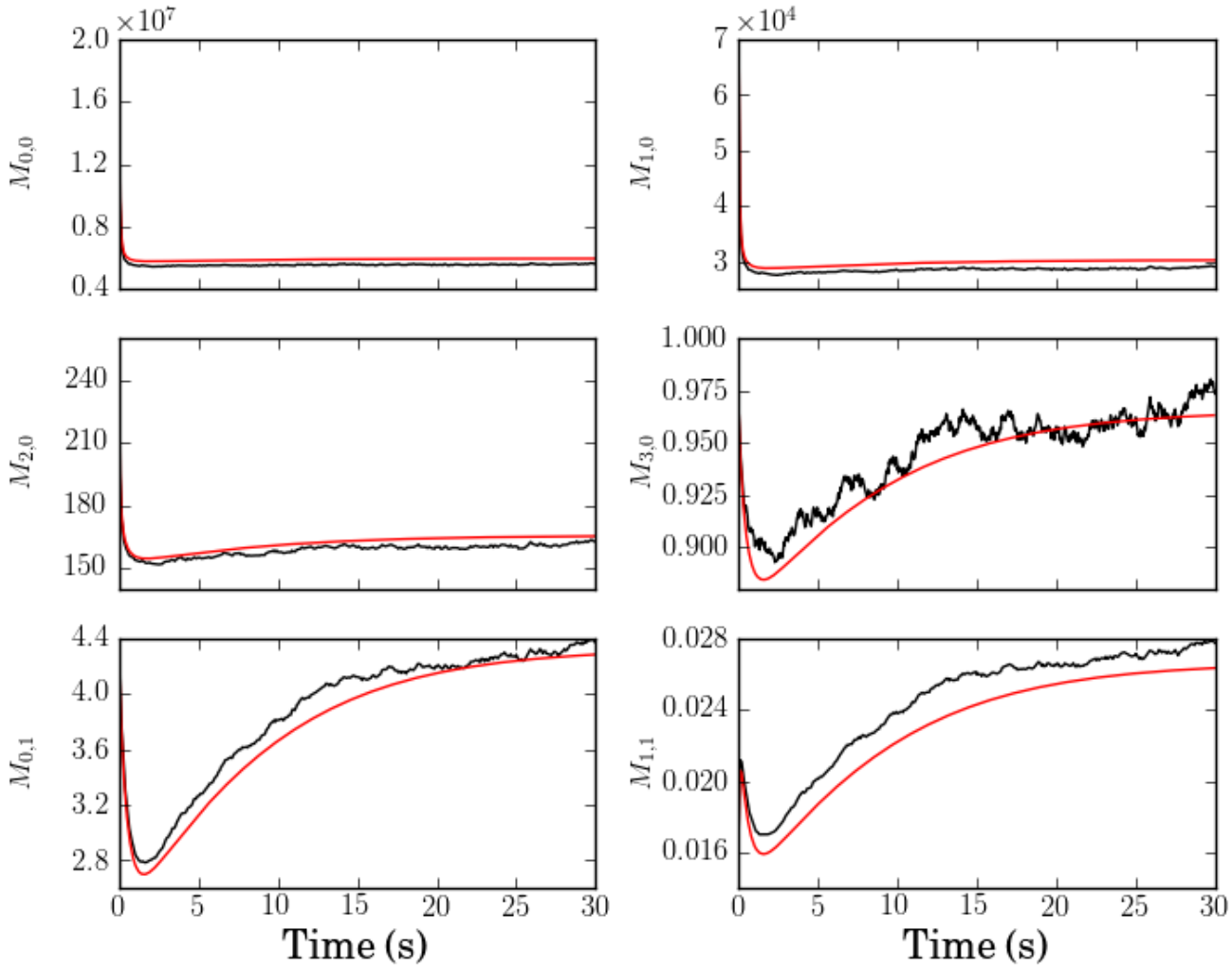
CQMOM with $N_1 = 3$ and $N_2 = 2$ tracking 15 moments of NDF



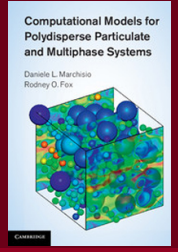
$$
\begin{array}{cccccc}
M_{0,3} & M_{1,3} & M_{2,3} & & & \\
M_{0,2} & M_{1,2} & M_{2,2} & & & \\
M_{0,1} & M_{1,1} & M_{2,1} & & & \\
M_{0,0} & M_{1,0} & M_{2,0} & M_{3,0} & M_{4,0} & M_{5,0}
\end{array}
$$



Comparison DSMC with QBMM



Coalescence, breakage and mass transfer. Solid black line: DSMC method. Red line CQMOM with $N_1 = 3$ and $N_2 = 1$ and DQMOM with $N = 3$.

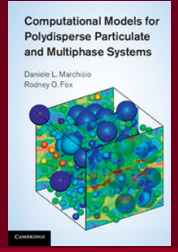


Comparison DSMC with QBMM

Mean percentage error of QBMM for examined case at steady state

	$M_{0,0}$	$M_{1,0}$	$M_{2,0}$	$M_{3,0}$	$M_{4,0}$	$M_{5,0}$	$M_{6,0}$	$M_{7,0}$
N=2	10.1	7.9	4.6	0.5	6.5	22.3	38.3	50.1
N=3	7.9	5.2	2.3	0.5	3.3	6.2	8.2	6.9
N=4	7.7	5.1	2.4	0.4	3.3	6.4	9.5	12.7

	$M_{0,1}$	$M_{1,1}$	$M_{2,1}$	$M_{3,1}$
N=2	2.1	6.5	21.4	44.6
N=3	1.9	1.8	4.7	6.2
N=4	3.2	1.7	3.4	6.4

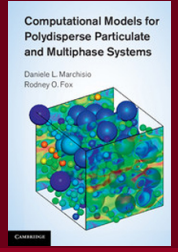


Comparison DSMC with QBMM

- With DSMC evolution of population of bubbles is represented through its stochastic equivalent (with $N \approx 100$ notional bubbles)
- With QBMM evolution is tracked deterministically with $N \approx 3-6$ macro-bubbles!
- These macro-bubbles are centered at quadrature nodes
- Therefore all unclosed terms appearing in equations for moments are closed with great accuracy
- Two distributions share same moments of NDF:

$$M_{k,l}(\mathbf{x},t) = \iint n(L,\phi_b;\mathbf{x},t) L^k \phi_b^l \, \mathrm{d}L \, \mathrm{d}\phi_b$$

- **Sharing same moments does not imply that quadrature has to be realizable (for example $M_{2,1}$ has to be included!)**

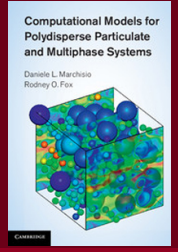


Comparison DSMC with QBMM

- With DSMC evolution of population of bubbles is represented through its stochastic equivalent (with $N \approx 100$ notional bubbles)
- With QBMM evolution is tracked deterministically with $N \approx 3-6$ macro-bubbles!
- These macro-bubbles are centered at quadrature nodes
- Therefore all unclosed terms appearing in equations for moments are closed with great accuracy
- Two distributions share same moments of NDF:

$$M_{k,l}(\mathbf{x},t) = \iint n(L,\phi_b;\mathbf{x},t) L^k \phi_b^l \, \mathrm{d}L \, \mathrm{d}\phi_b$$

- **Sharing same moments does not imply that quadrature has to be realizable (for example $M_{2,1}$ has to be included!)**

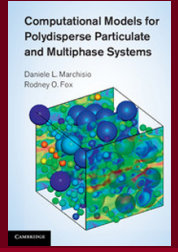


Comparison DSMC with QBMM

- With DSMC evolution of population of bubbles is represented through its stochastic equivalent (with $N \approx 100$ notional bubbles)
- With QBMM evolution is tracked deterministically with $N \approx 3 - 6$ macro-bubbles!
- These macro-bubbles are centered at quadrature nodes
- Therefore all unclosed terms appearing in equations for moments are closed with great accuracy
- Two distributions share same moments of NDF:

$$M_{k,l}(\mathbf{x}, t) = \iint n(L, \phi_b; \mathbf{x}, t) L^k \phi_b^l \, \mathrm{d}L \, \mathrm{d}\phi_b$$

- **Sharing same moments does not imply that quadrature has to be realizable (for example $M_{2,1}$ has to be included!)**

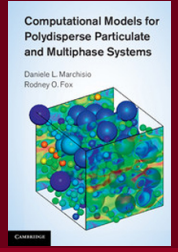


Comparison DSMC with QBMM

- With DSMC evolution of population of bubbles is represented through its stochastic equivalent (with $N \approx 100$ notional bubbles)
- With QBMM evolution is tracked deterministically with $N \approx 3 - 6$ macro-bubbles!
- These macro-bubbles are centered at quadrature nodes
- Therefore all unclosed terms appearing in equations for moments are closed with great accuracy
- Two distributions share same moments of NDF:

$$M_{k,l}(\mathbf{x}, t) = \iint n(L, \phi_b; \mathbf{x}, t) L^k \phi_b^l \, \mathrm{d}L \, \mathrm{d}\phi_b$$

- **Sharing same moments does not imply that quadrature has to be realizable (for example $M_{2,1}$ has to be included!)**

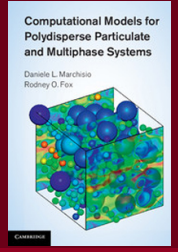


Comparison DSMC with QBMM

- With DSMC evolution of population of bubbles is represented through its stochastic equivalent (with $N \approx 100$ notional bubbles)
- With QBMM evolution is tracked deterministically with $N \approx 3 - 6$ macro-bubbles!
- These macro-bubbles are centered at quadrature nodes
- Therefore all unclosed terms appearing in equations for moments are closed with great accuracy
- Two distributions share same moments of NDF:

$$M_{k,l}(\mathbf{x}, t) = \iint n(L, \phi_b; \mathbf{x}, t) L^k \phi_b^l \, \mathrm{d}L \, \mathrm{d}\phi_b$$

- **Sharing same moments does not imply that quadrature has to be realizable (for example $M_{2,1}$ has to be included!)**

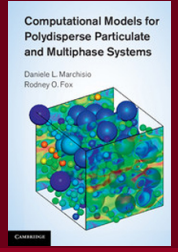


Comparison DSMC with QBMM

- With DSMC evolution of population of bubbles is represented through its stochastic equivalent (with $N \approx 100$ notional bubbles)
- With QBMM evolution is tracked deterministically with $N \approx 3 - 6$ macro-bubbles!
- These macro-bubbles are centered at quadrature nodes
- Therefore all unclosed terms appearing in equations for moments are closed with great accuracy
- Two distributions share same moments of NDF:

$$M_{k,l}(\mathbf{x}, t) = \iint n(L, \phi_b; \mathbf{x}, t) L^k \phi_b^l \, \mathrm{d}L \, \mathrm{d}\phi_b$$

- **Sharing same moments does not imply that quadrature has to be realizable (for example $M_{2,1}$ has to be included!)**

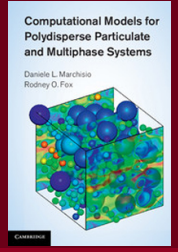


Comparison DSMC with QBMM

- With DSMC evolution of population of bubbles is represented through its stochastic equivalent (with $N \approx 100$ notional bubbles)
- With QBMM evolution is tracked deterministically with $N \approx 3 - 6$ macro-bubbles!
- These macro-bubbles are centered at quadrature nodes
- Therefore all unclosed terms appearing in equations for moments are closed with great accuracy
- Two distributions share same moments of NDF:

$$M_{k,l}(\mathbf{x}, t) = \iint n(L, \phi_b; \mathbf{x}, t) L^k \phi_b^l \, \mathrm{d}L \, \mathrm{d}\phi_b$$

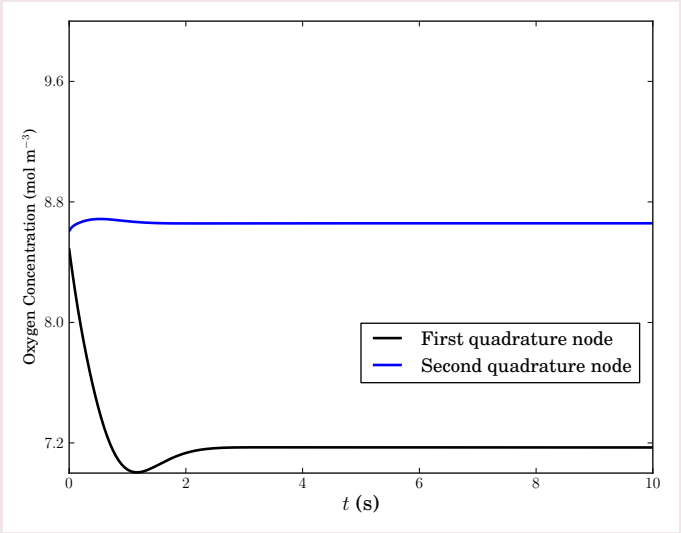
- **Sharing same moments does not imply that quadrature has to be realizable (for example $M_{2,1}$ has to be included!)**



Quadrature Realizability

Coalescence and breakage WITHOUT MASS TRANSFER

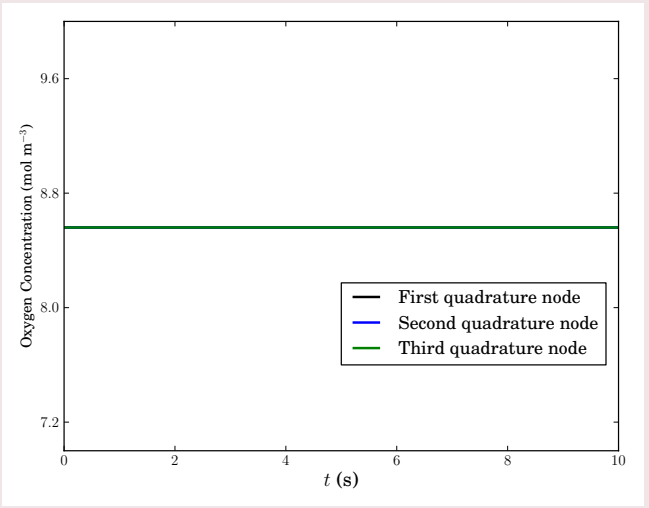
$N = 2$ for 6 moments



$M_{0,1}$ $M_{1,1}$

$M_{0,0}$ $M_{1,0}$ $M_{2,0}$ $M_{3,0}$

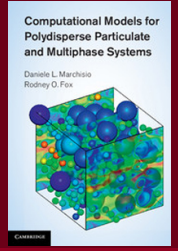
$N = 3$ for 9 moments



$M_{0,1}$ $M_{1,1}$ $M_{2,1}$

$M_{0,0}$ $M_{1,0}$ $M_{2,0}$ $M_{3,0}$ $M_{4,0}$ $M_{5,0}$

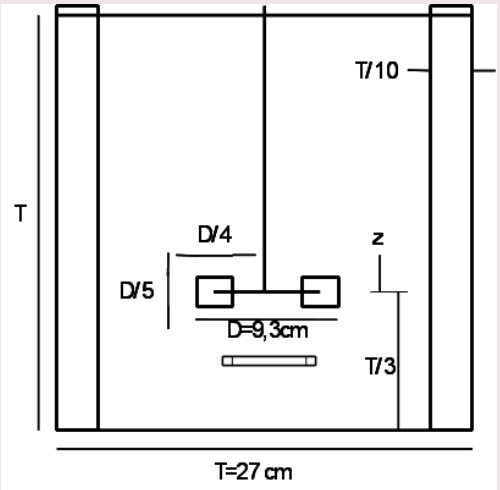
Without including $M_{2,1}$ quadrature is not realizable!

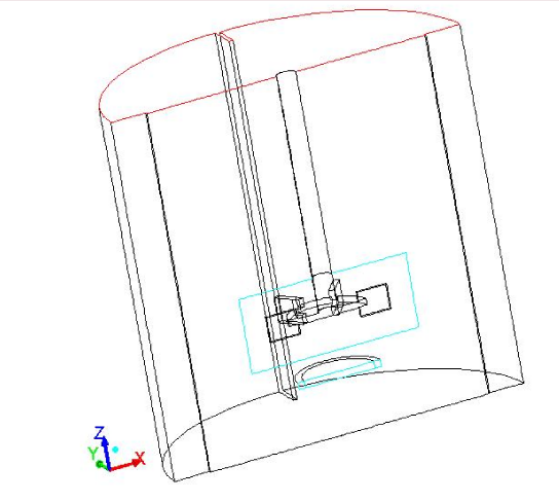


DSMC for Stirred Tanks

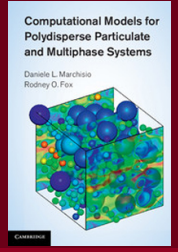
Let us now consider a realistic stirred tank

- Rushton turbine (six blades) - reactor volume 15,3 l
- Gassing rate $\approx$ 0,062 l/s; stirring rate 100–300 rpm
- Flow field simulation ANSYS/Fluent13 - DSMC in-house code





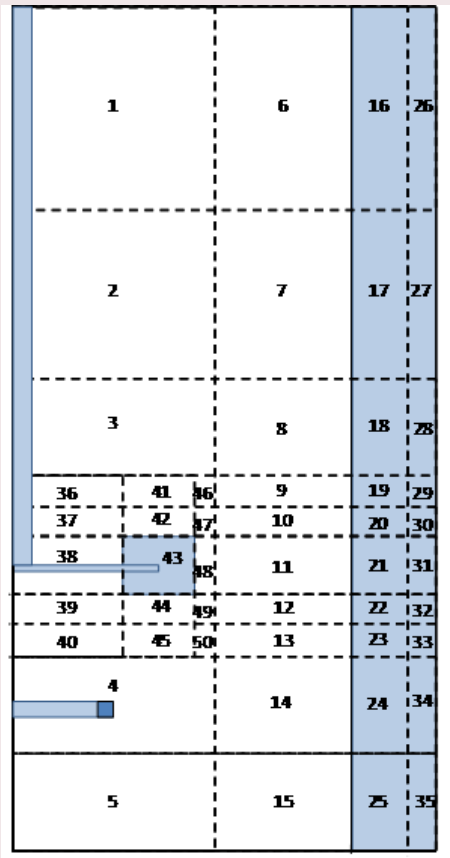
Wu & Patterson, Chem. Eng. Sci., 44, 2207-2221, 1989

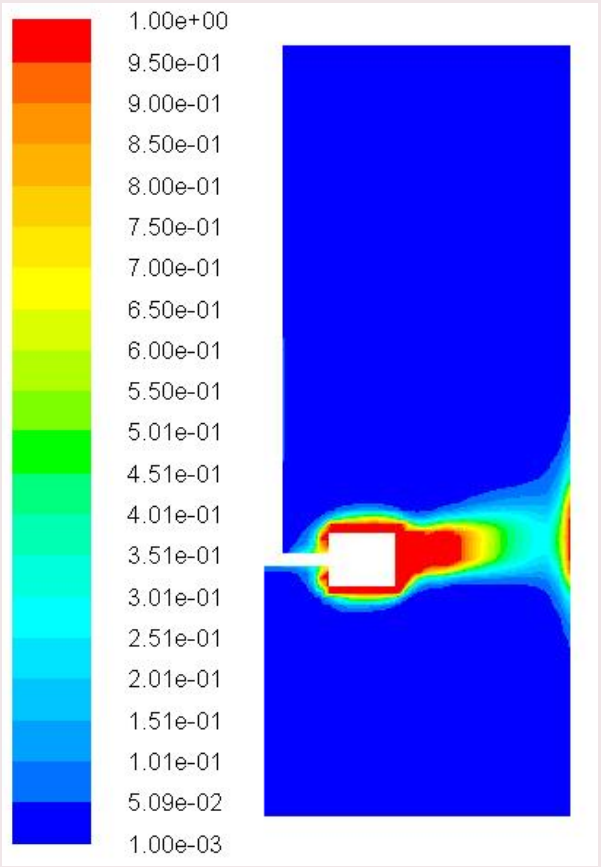


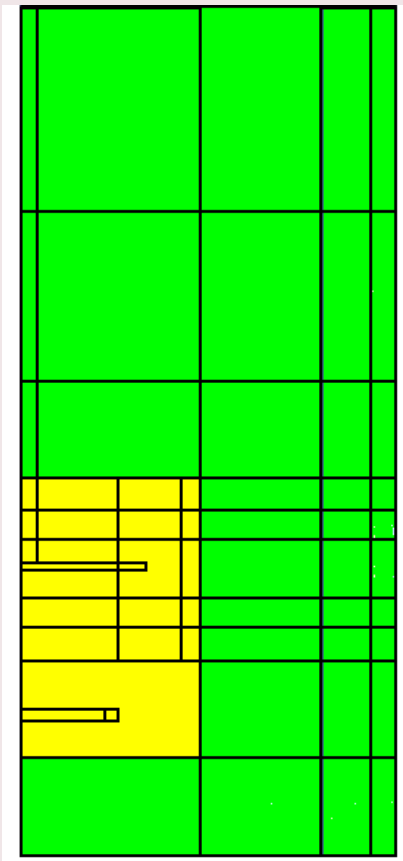
DSMC for Stirred Tanks

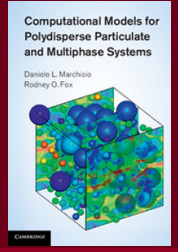
Compartment model

Tot. numb. comp. Turb. diss. rate m^2s^{-3} $N \approx 2$





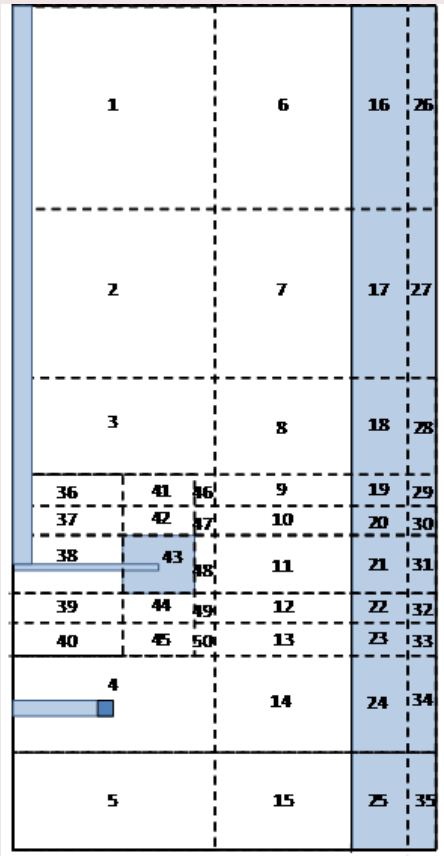


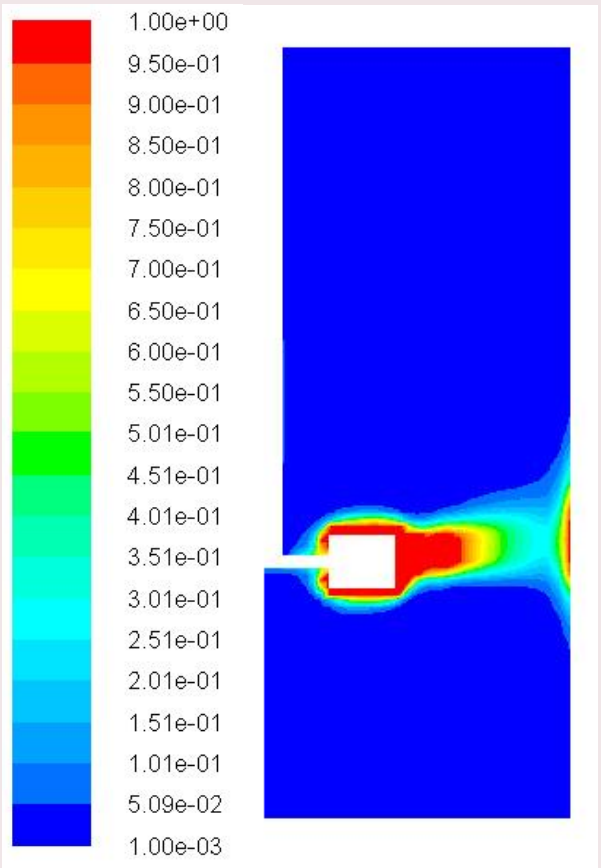


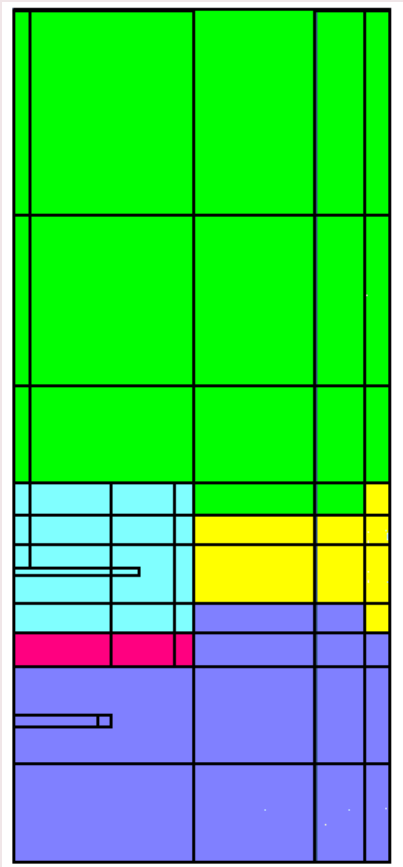
DSMC for Stirred Tanks

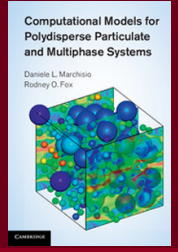
Compartment model

Tot. numb. comp. Turb. diss. rate m^2s^{-3} $N \approx 5$





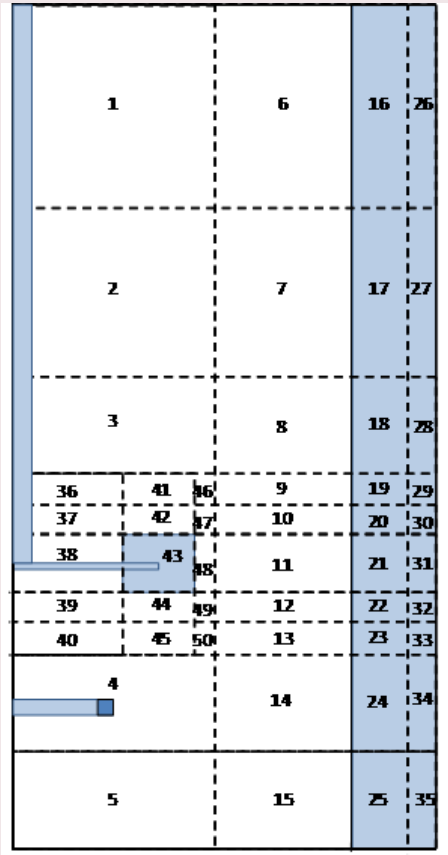


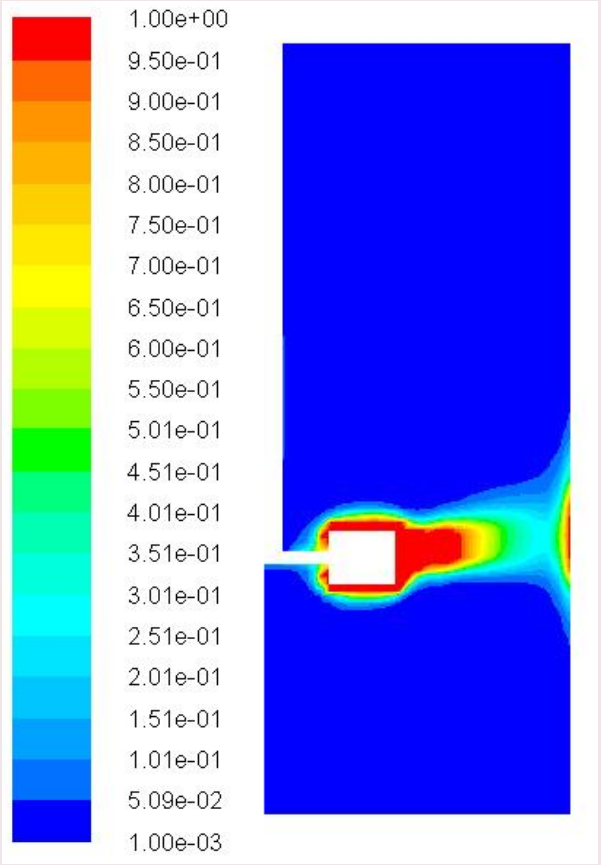


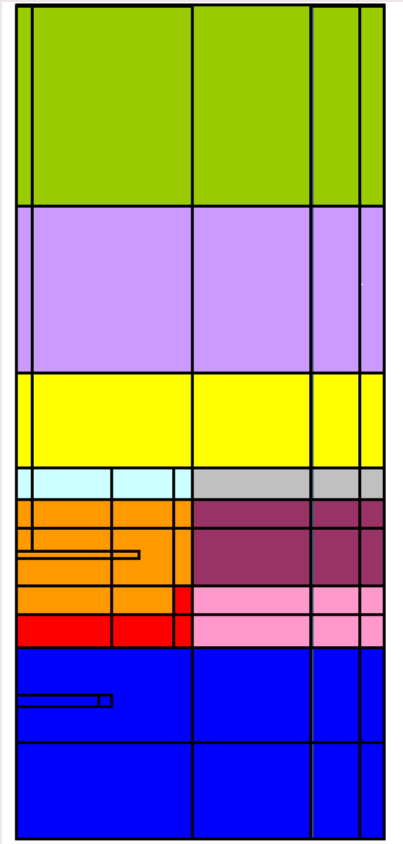
DSMC for Stirred Tanks

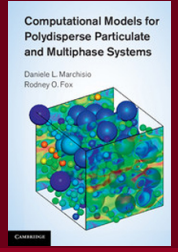
Compartment model

Tot. numb. comp. Turb. diss. rate m^2s^{-3} $N \approx 10$



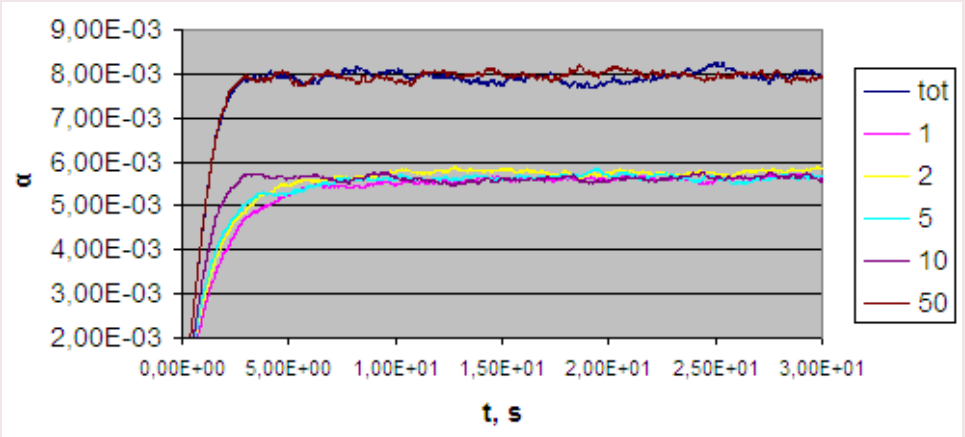


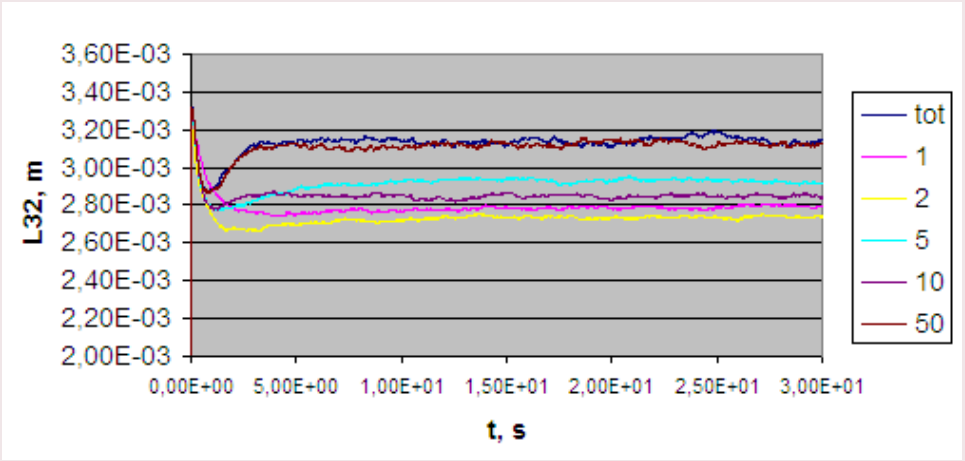


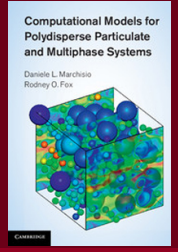


DSMC for Stirred Tanks

Effect number of comp. on gas hold-up and Sauter diameter

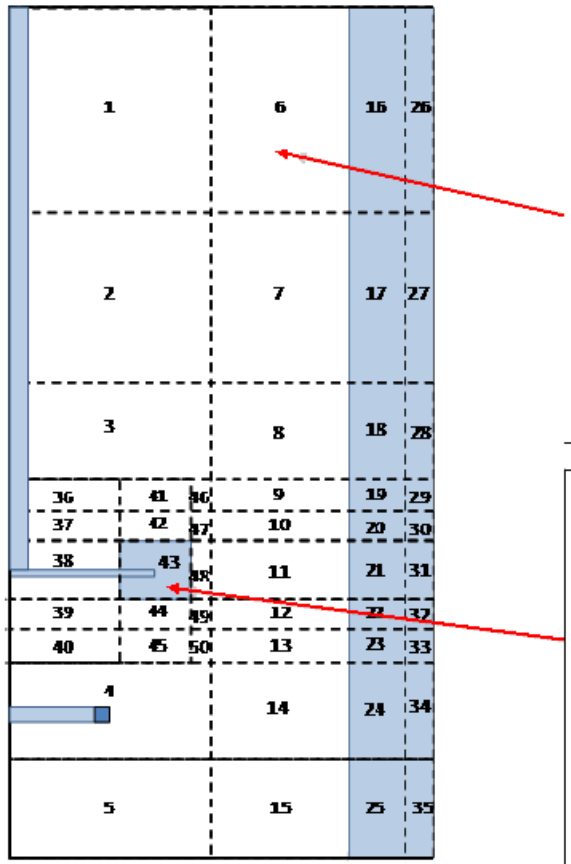


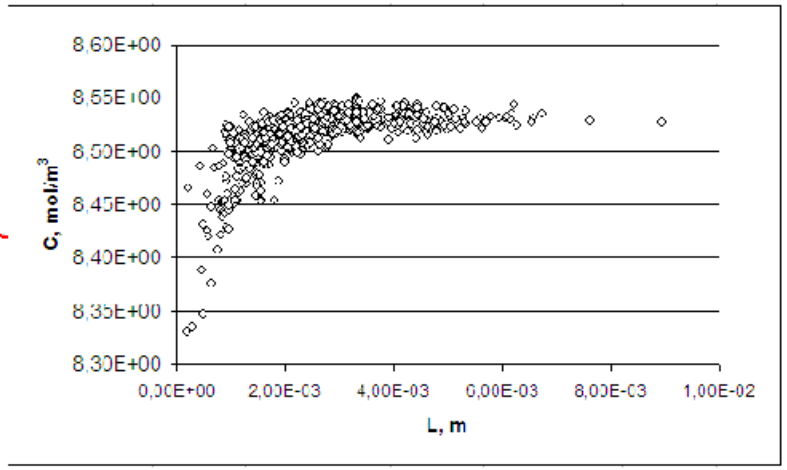


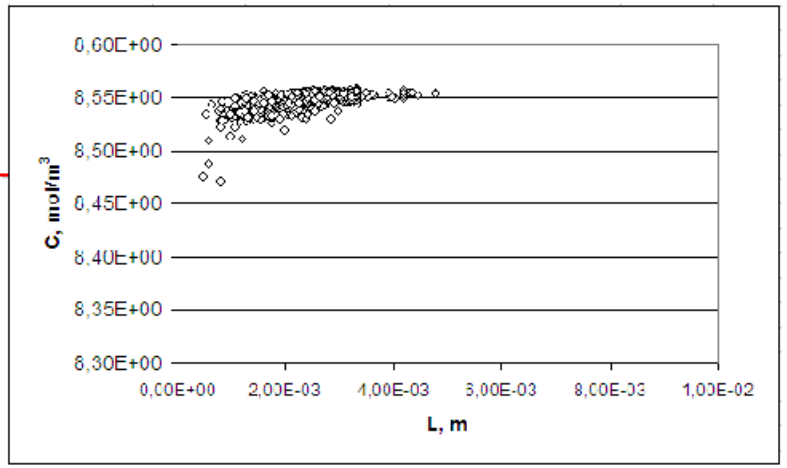


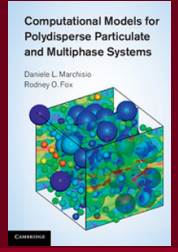
DSMC for Stirred Tanks

Bubble size-composition in two regions



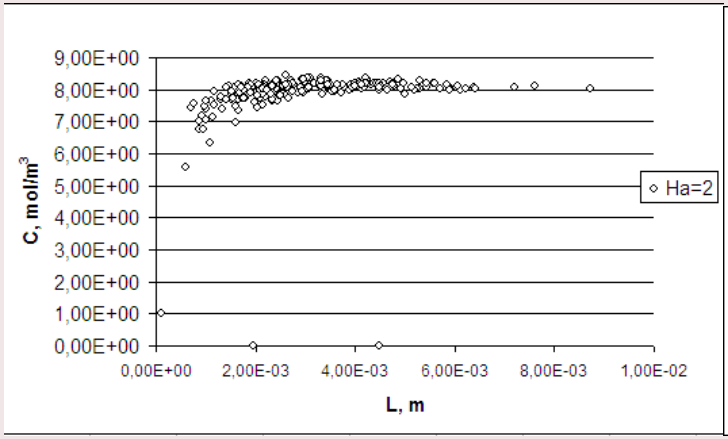


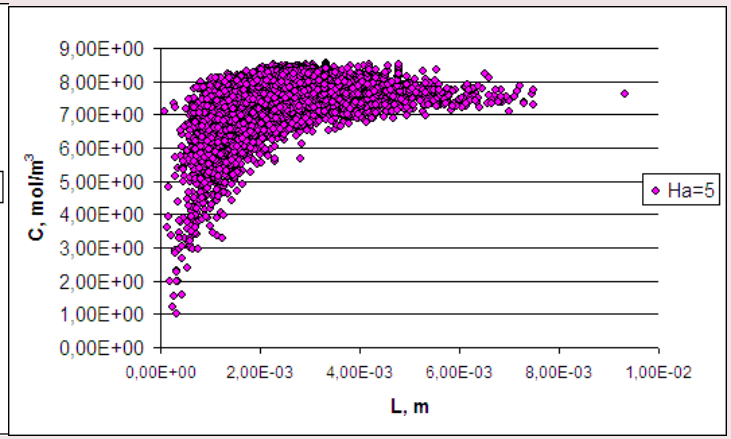


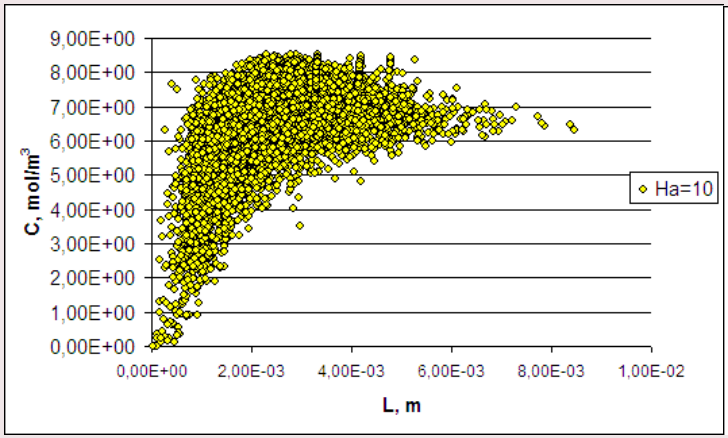


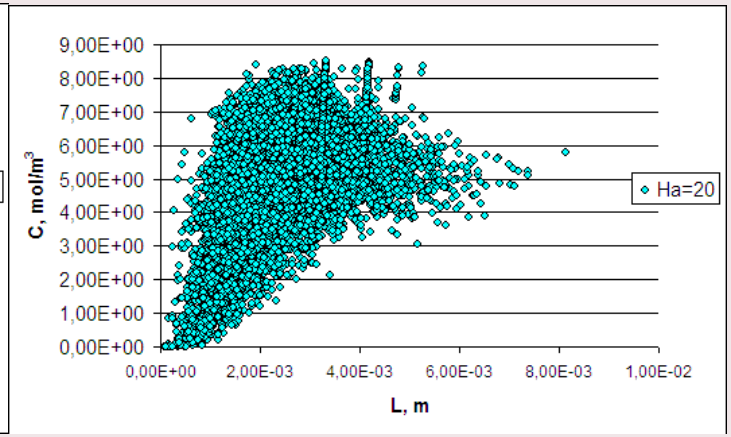
DSMC for Stirred Tanks

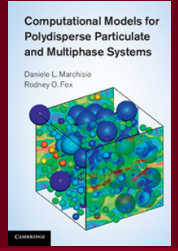
Effect of chemical reaction (Ha = $\sqrt{kC_{O_2}D}/k_l$)



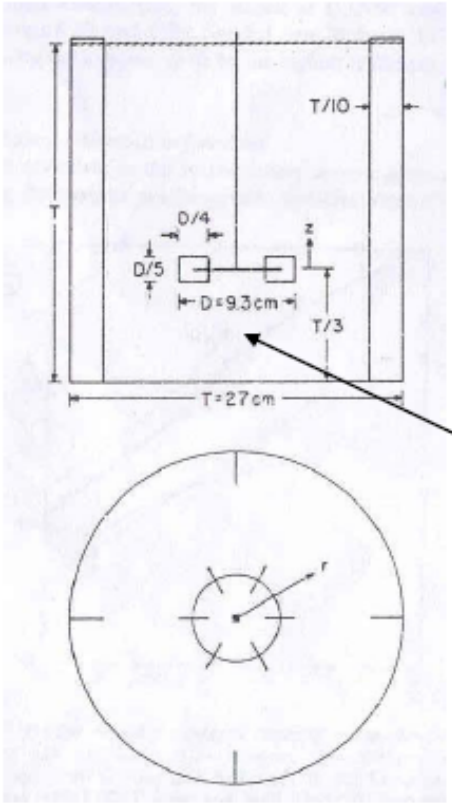








Simulation of Stirred Tanks with QBMM



Standard Rushton turbine

Size baffles and blades: 0.44 cm

Diameter shaft : 1.46 cm

Diameter disk : 6.02 cm

Width disk : 0.47 cm

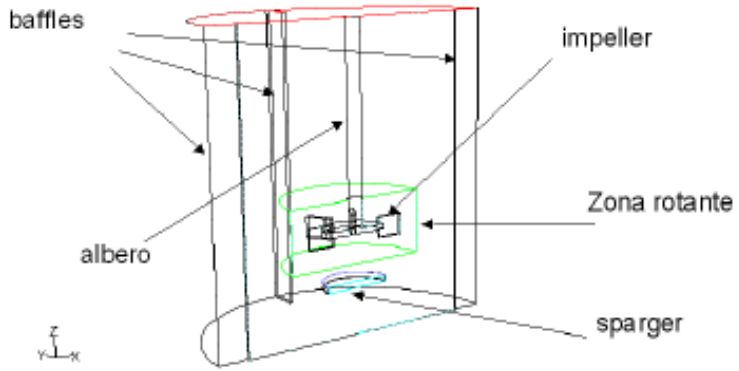
$V_{liq} \cong 15{,}3$ liter

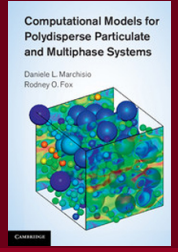
Sparger: z = -4.5 cm; d = 6 cm;

46 holes d = 0.5 mm

Simulation carried out on half reactor

Ncelle: 166563





Drag Laws

- C_D Tomiyama with Bakker's correction for accounting for turbulence effects

$$C_D = \max\left\{\min\left(\frac{24}{\mathrm{Re}}\left(1+0.15\mathrm{Re}^{0.687}\right), \frac{72}{\mathrm{Re}}\right), \frac{8}{3}\frac{Eo}{Eo+4}\right\}$$

Calculated with $\mu_{eff} = \mu_l + C_l \rho_l \varepsilon^{1/3} d^{4/3}$ (0.2)

Bakker's correction

effective viscosity | laminar viscosity | turbulent viscosity

- C_D calculated by assuming a Terminal Velocity constant and equal to 13 cm/s

$$C_D = \frac{4 d_b (\rho_l - \rho_g)}{3 \rho_l U_\infty^2}$$

Relevant groups

Gas Flow Number

$$Fl_g = \frac{Q_g}{ND^3}$$

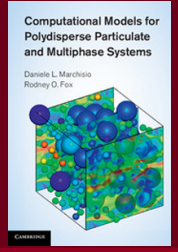
Froude Number

$$Fr = \frac{N^2 D}{g}$$

D: impeller size

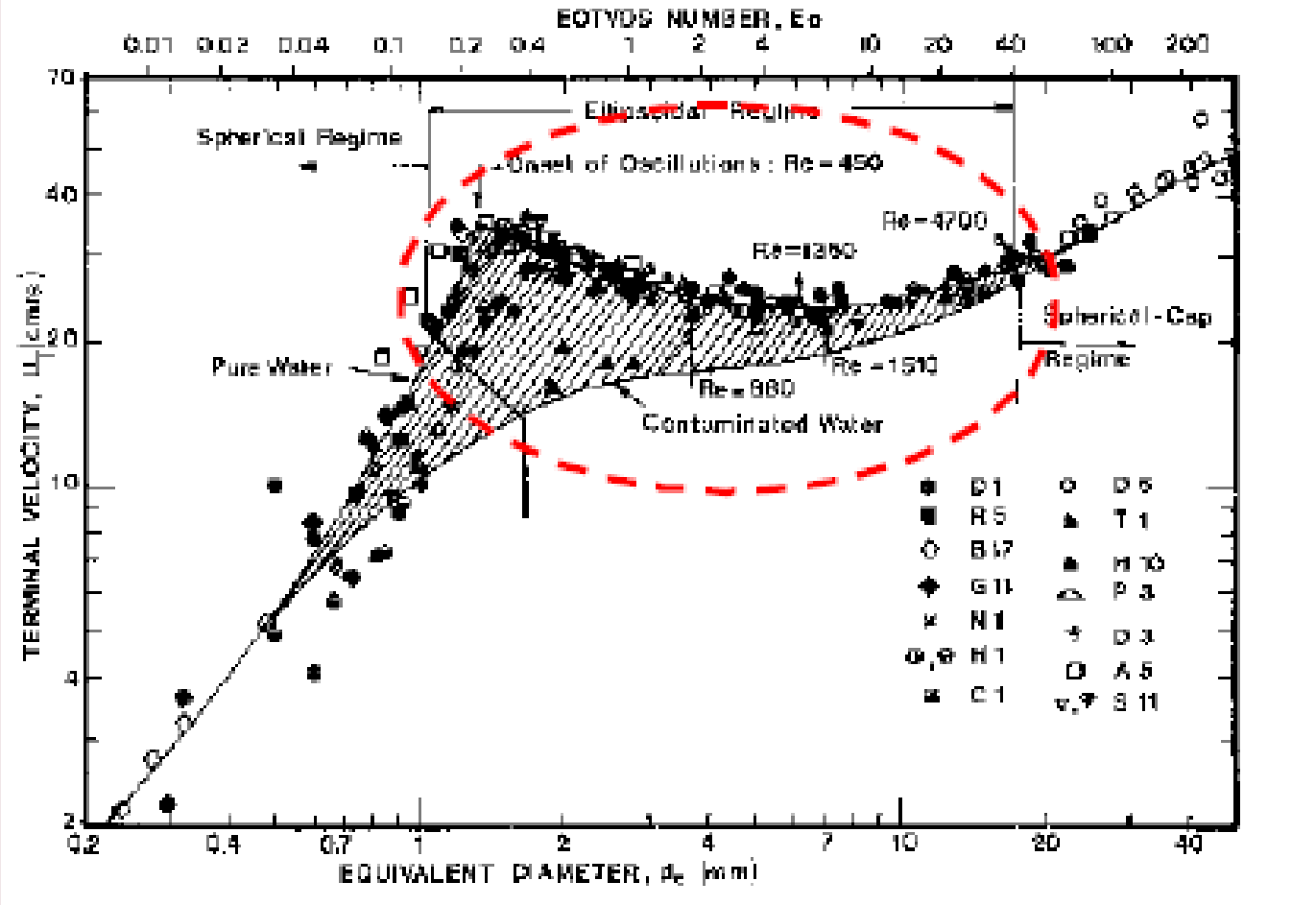
N: stirring rate (rps)

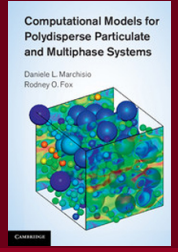
g: gravity acceleration



Drag Laws

Drag force is evaluated from terminal velocity taking into account effect of turbulence and swarms (U_t = 13 – 8 cm/s)

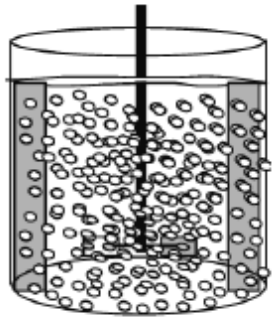




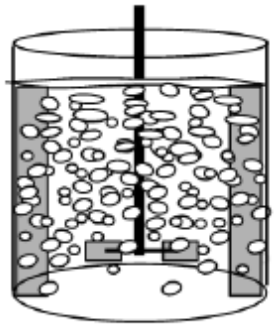
Fluid-Dynamic Regimes in Stirred Tanks

FLUID-DYNAMIC REGIMES

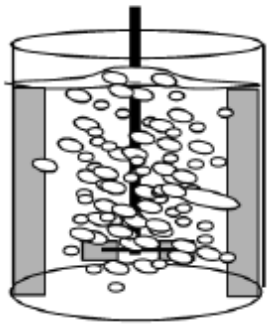
	300rpm	500rpm
$F_{lg}\_CD = (13*F_r\hat{}2)*(D/T)\hat{}5$ Complete dispersion (Nienow 1977) soglia max per F_{lg} per avere ricircolo gas	0.0035	0.0273
$F_{lg}\_CD = 0.2*((D/T)\hat{}0.5)*F_r$ soglia per complete dispersion transition (correlazione 2)	0.0278	0.0773
$F_{lg}\_F = 30F_r*(D/T)\hat{}3.5$ soglia min per F_{lg} per avere flooding	0.1705	0.4737



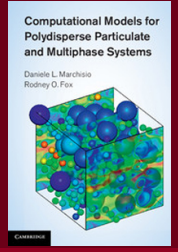
Complete dispersion



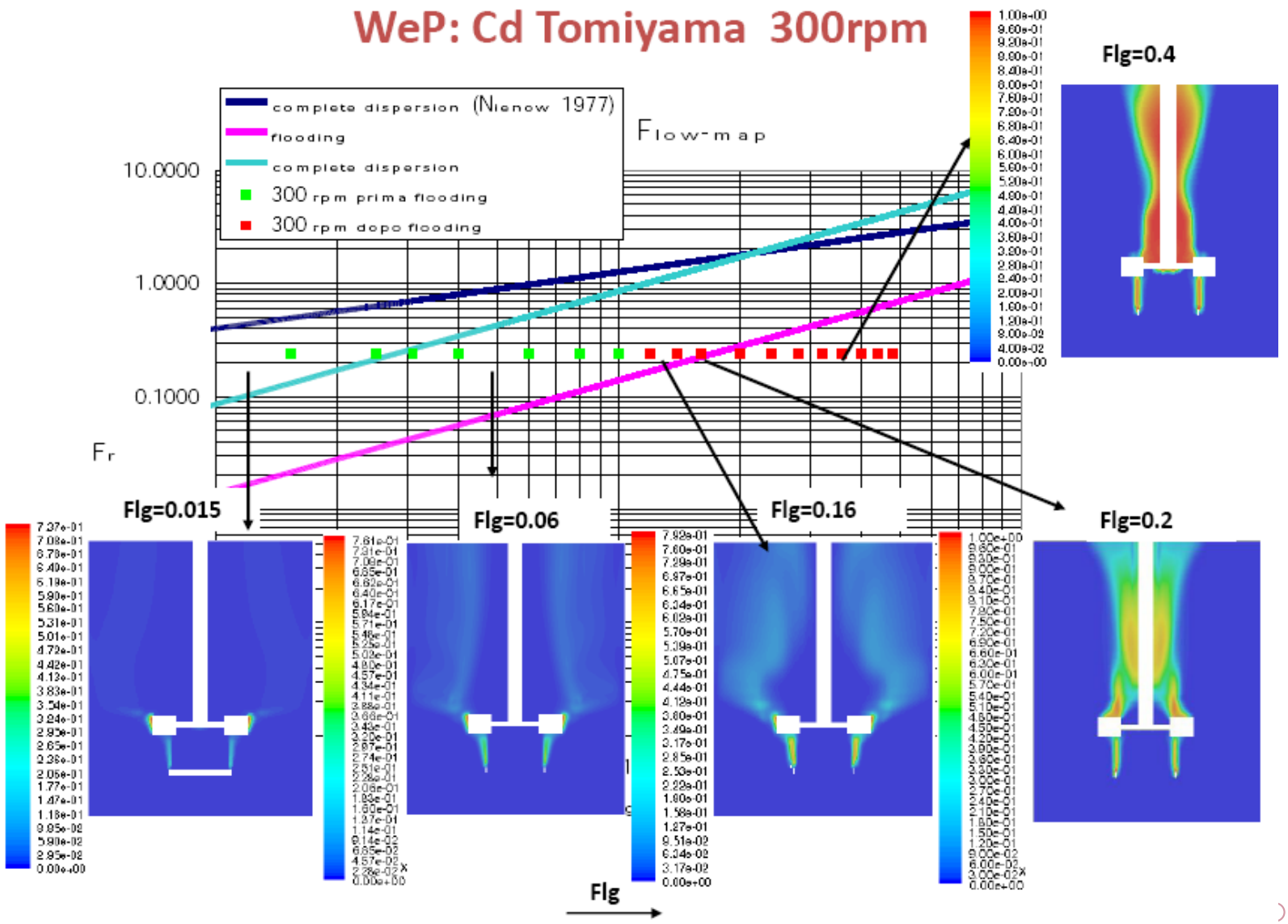
Loading

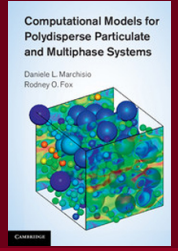


Flooding

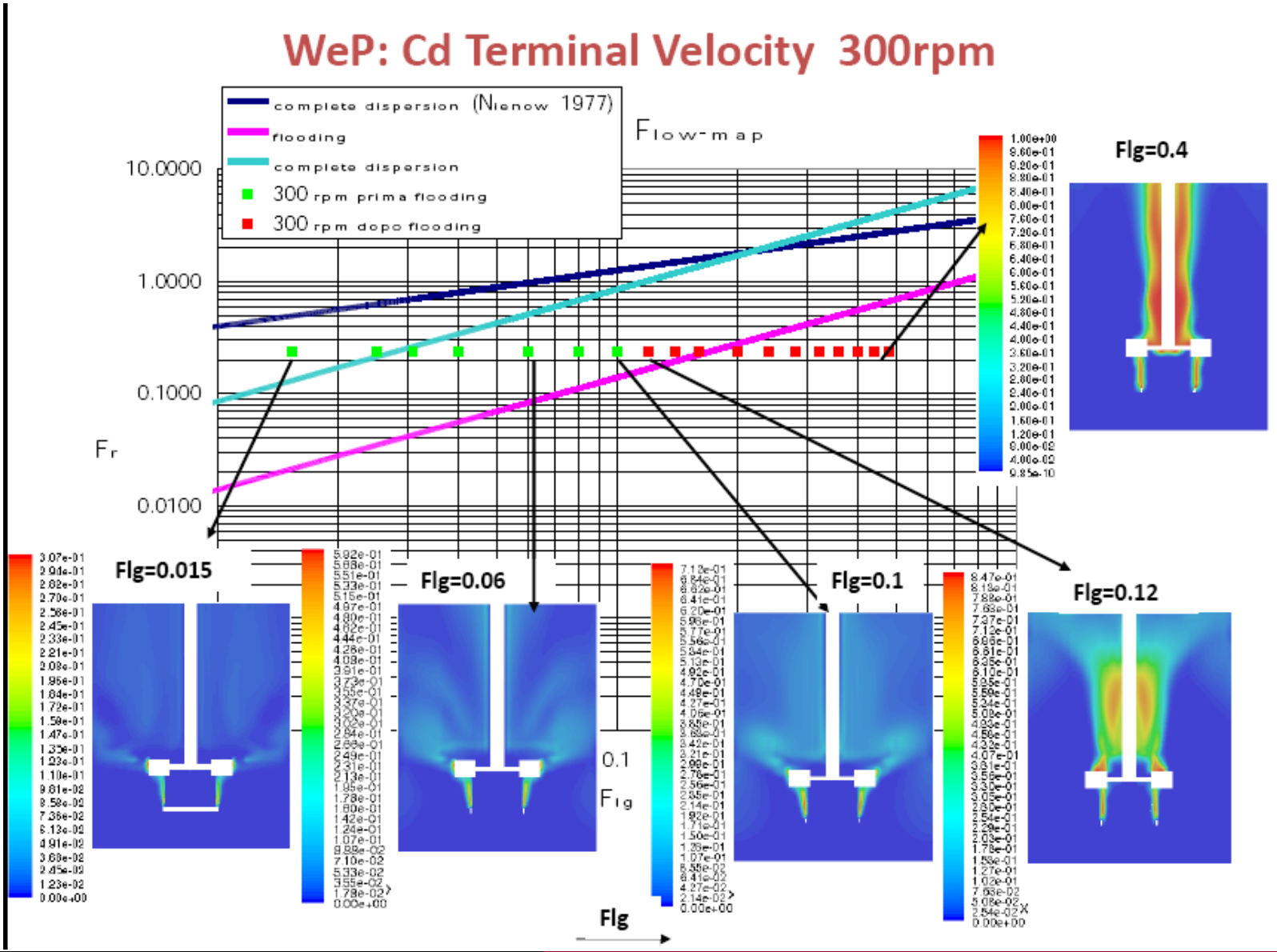


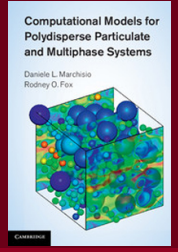
Fluid-Dynamic Regimes in Stirred Tanks





Fluid-Dynamic Regimes in Stirred Tanks

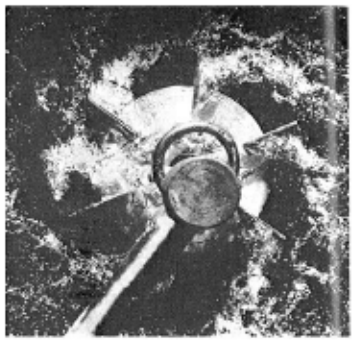




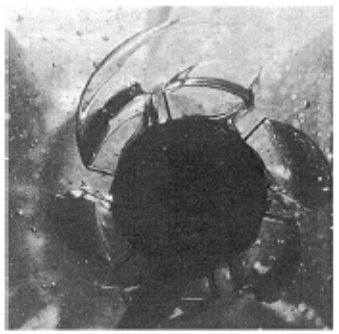
Fluid-Dynamic Regimes for Impeller

	300rpm	500rpm
$F_{ig}\_S33 = 3.8*10^{-3}*((Re^2/Fr)^{0.07})*((D/T)^{-0.5})$ soglia max per avere VC structure (vortex clinging cavities) VC per $0<F_{ig}<F_{ig}\_S33$	0.0319	0.0319
F_{ig}_L33 soglia max per avere S33 structure (3 large+3clinging cavities) S33 per $F_{ig}\_S33 < F_{ig} < F_{ig}\_L33$	0.1000	0.1000
$F_{ig}\_LC = 0.025*(D/T)^{(-0.5)}$ soglia min per F_{ig} per avere large cavities	0.0426	0.0426
$F_{ig}\_L33 = F_{ig}\_Flooding = 30Fr*(D/T)^{3.5}$ soglia max per avere L33 structure (3 smaller+3 larger cavities) L33 per $F_{ig}\_L33 < F_{ig} < F_{ig}\_flooding$	0.1705	0.4737

vortex cavity

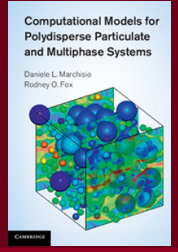


clinging cavity

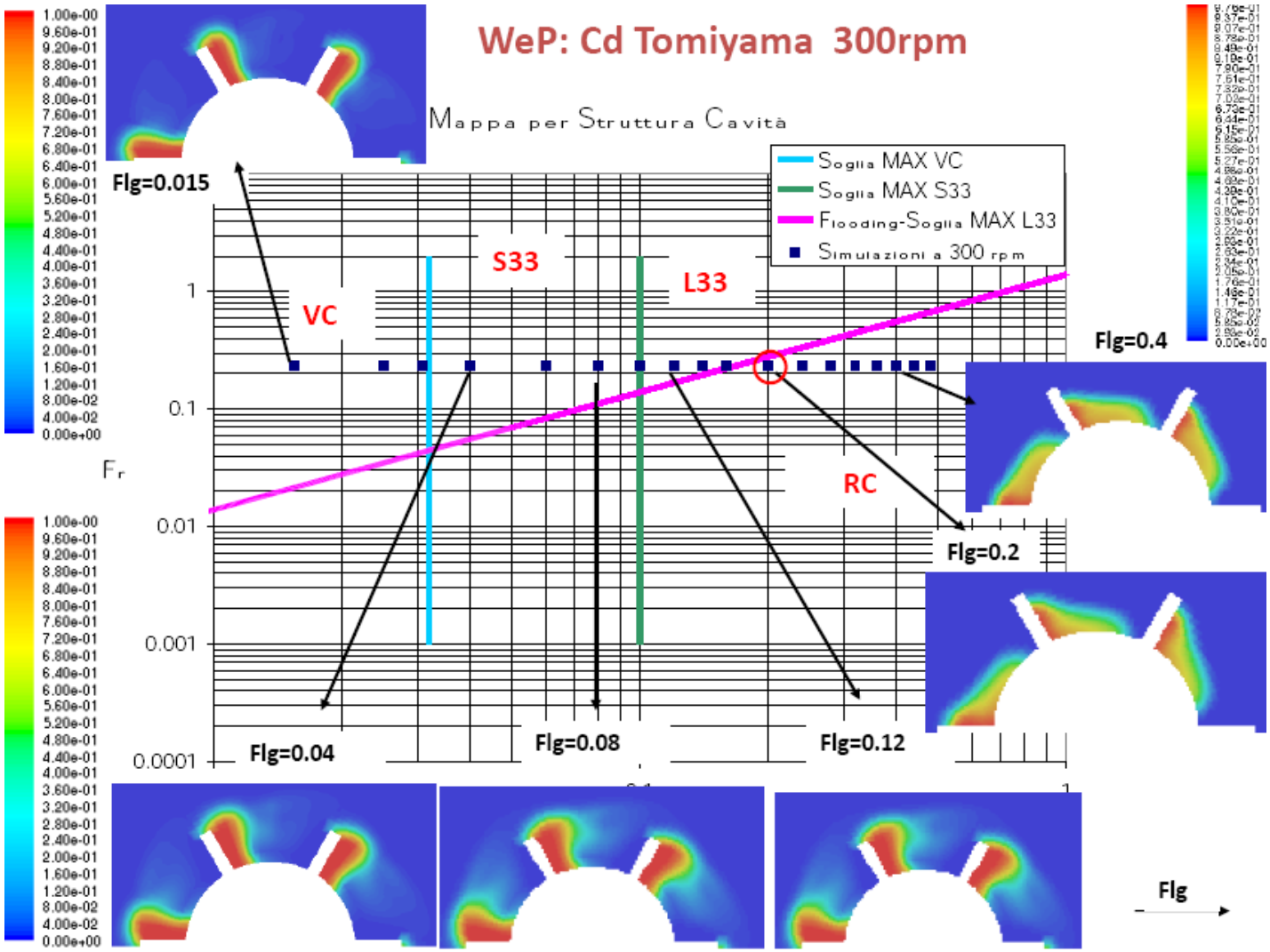


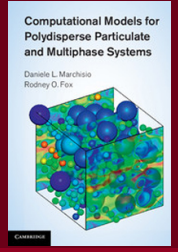
large cavity

After flooding Ragged Cavities (RC)

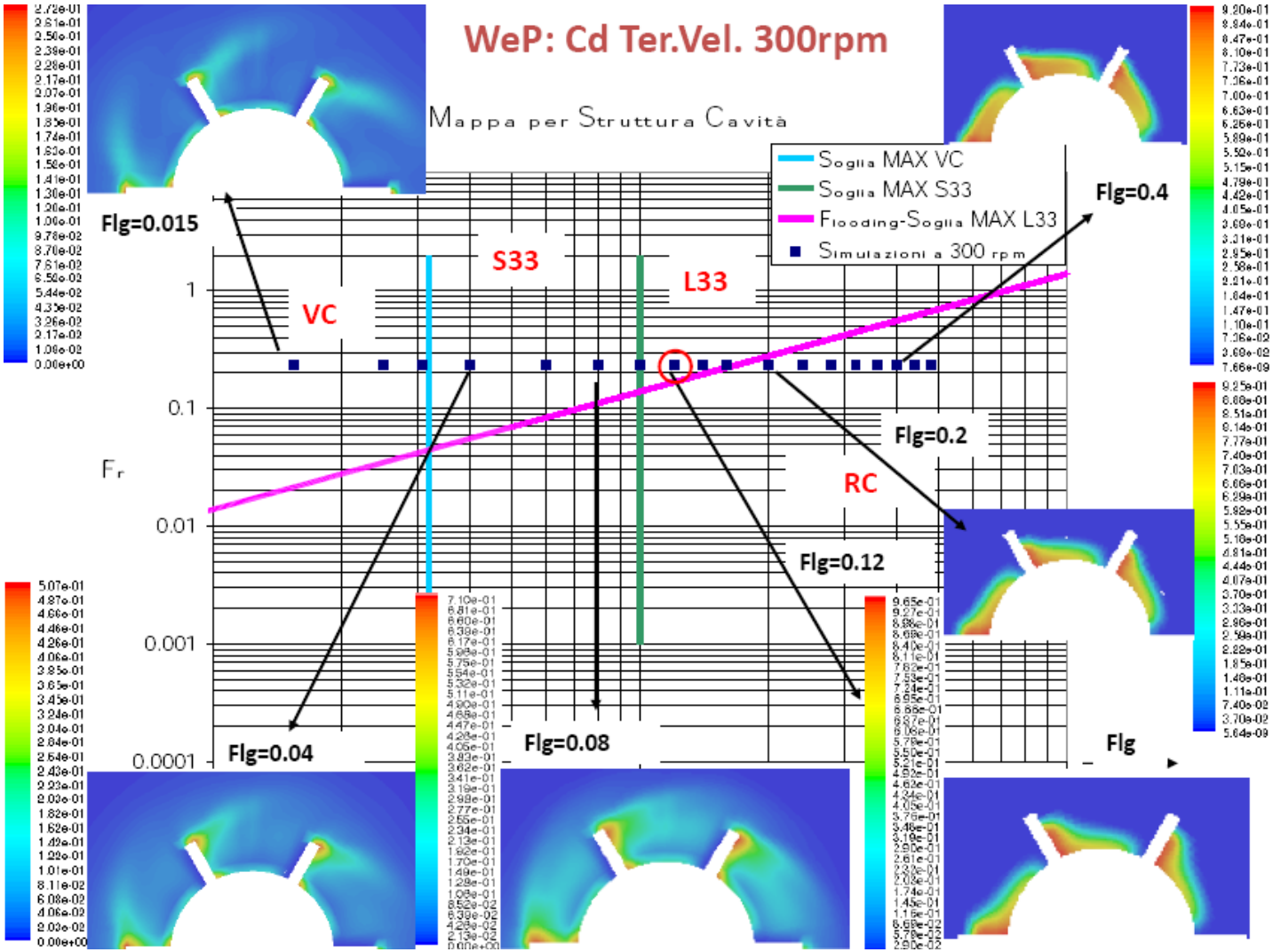


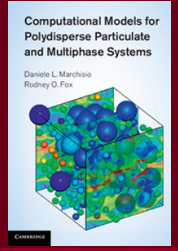
Fluid-Dynamic Regimes for Impeller





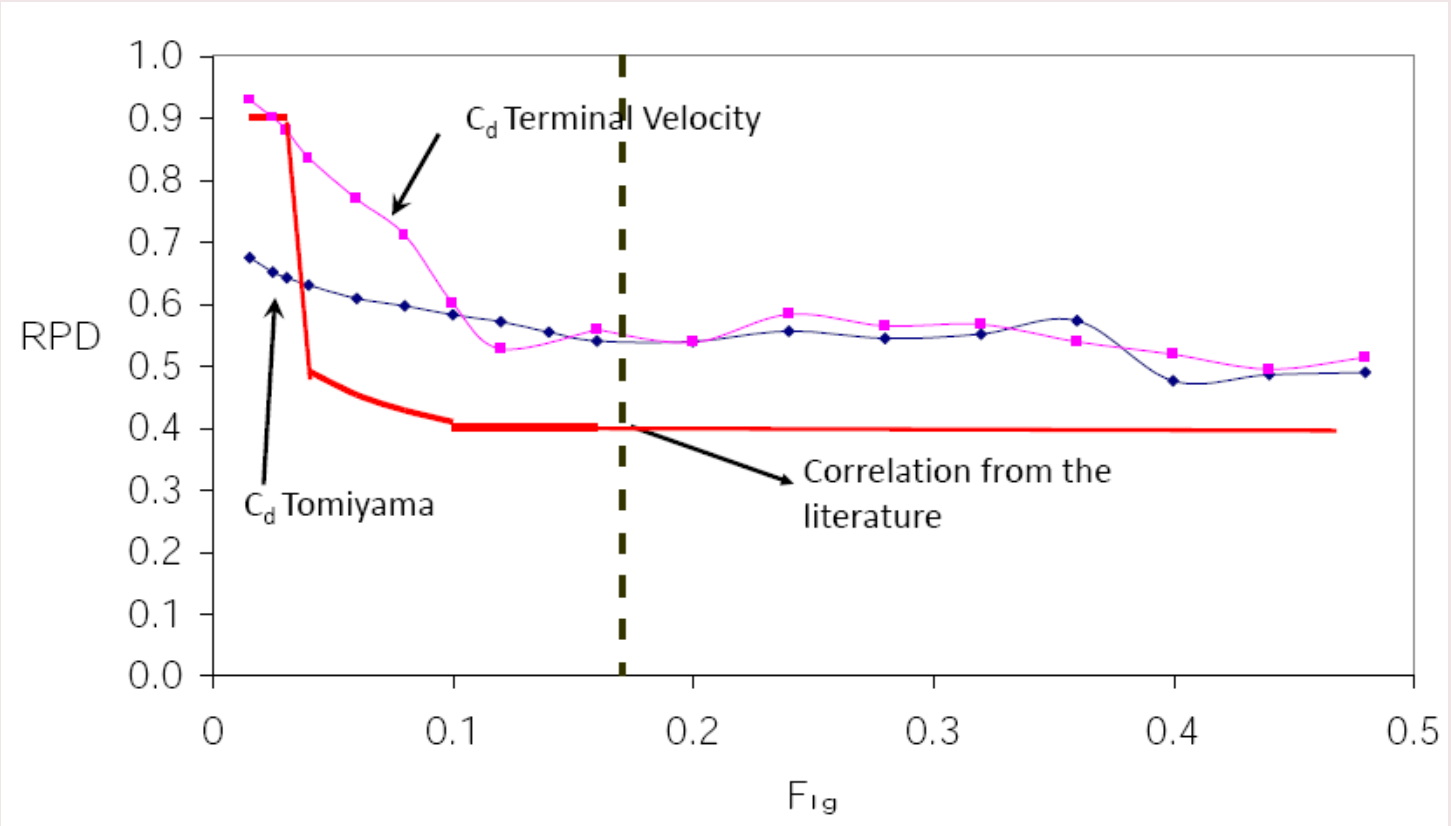
Fluid-Dynamic Regimes for Impeller

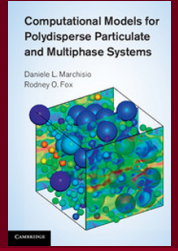




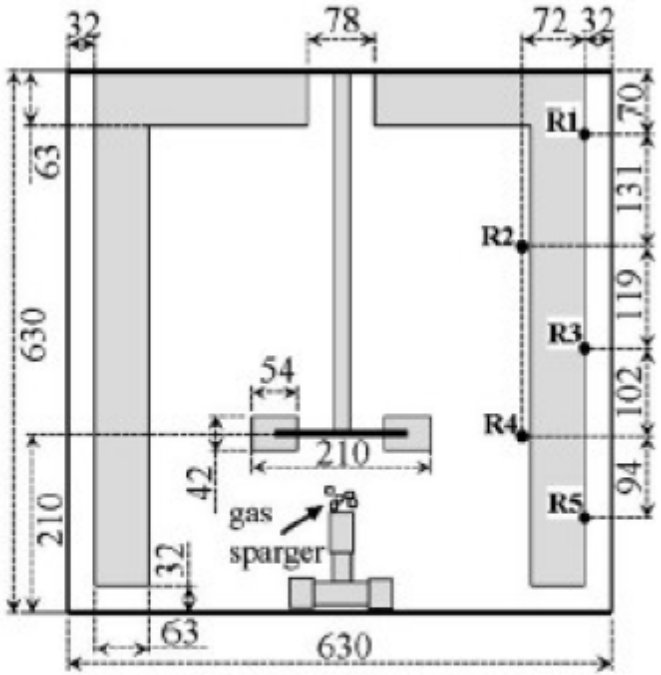
Power-Number Reduction (RPD)

Power-number reduction (RPD) due to presence of liquid



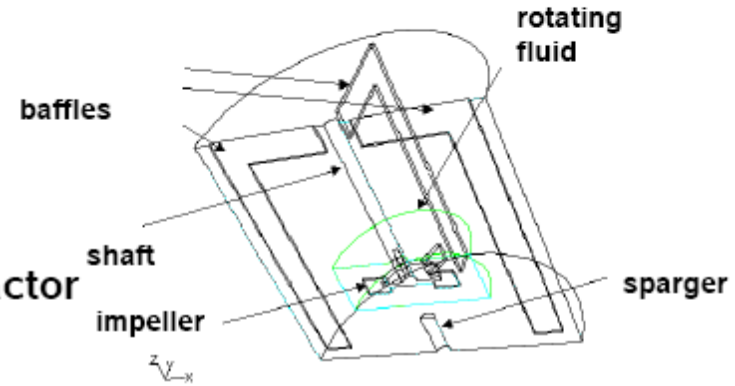


QBMM for Stirred Tanks



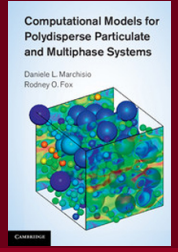
Porous sparger

Standard Rushton turbine
Baffles and blades width: 1.03 cm
Shaft diameter: 3.3 cm
Disk diameter: 13.6 cm
Turbine diameter: 21cm
Disk width: 1.06 cm
Sparger position: z = -10.5 cm; d = 3.3 cm
Sparger width d = 15 mm

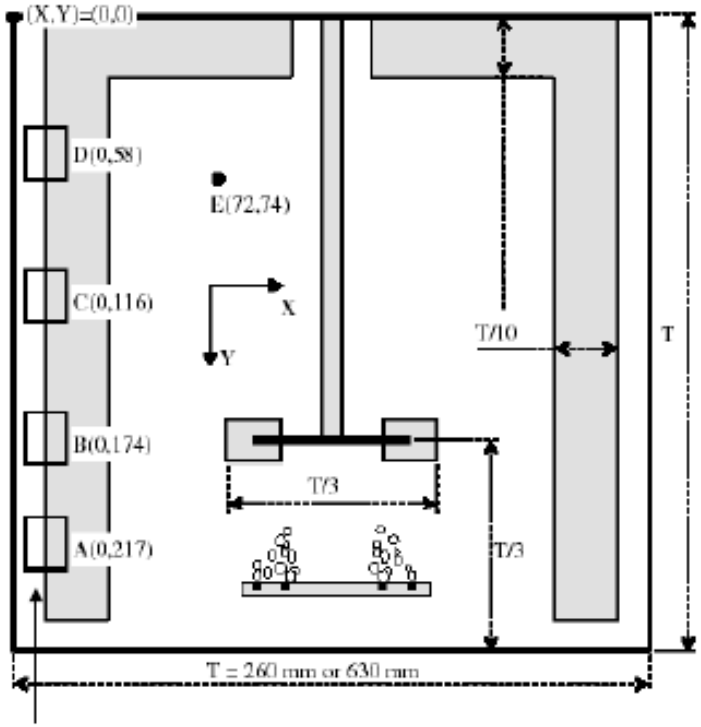


Computational domain on half reactor
→ N_{cells}: 226776

Laakkonen M, Alopaeus V, Aittamaa J. Validation of bubble breakage, coalescence and mass transfer models for gas–liquid dispersion in agitated vessel. Chemical Engineering Science. 2006, 61: 218-228.



QBMM for Stirred Tanks



Ring sparger (12 holes 2 mm of diameter)

Computational domain on half reactor

→ N_{cells}: 226776

Standard Rushton turbine

Baffles and blades width: 1.03 cm

Shaft diameter: 3.3 cm

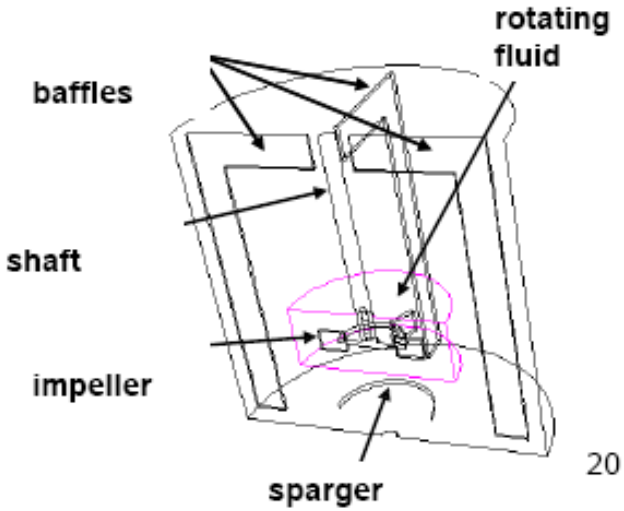
Disk diameter: 13.6 cm

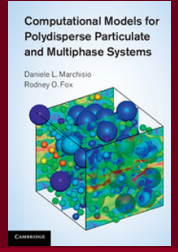
Turbine diameter: 21cm

Disk width: 1.06 cm

Sparger position: z = -10.5 cm; d = 3.3 cm

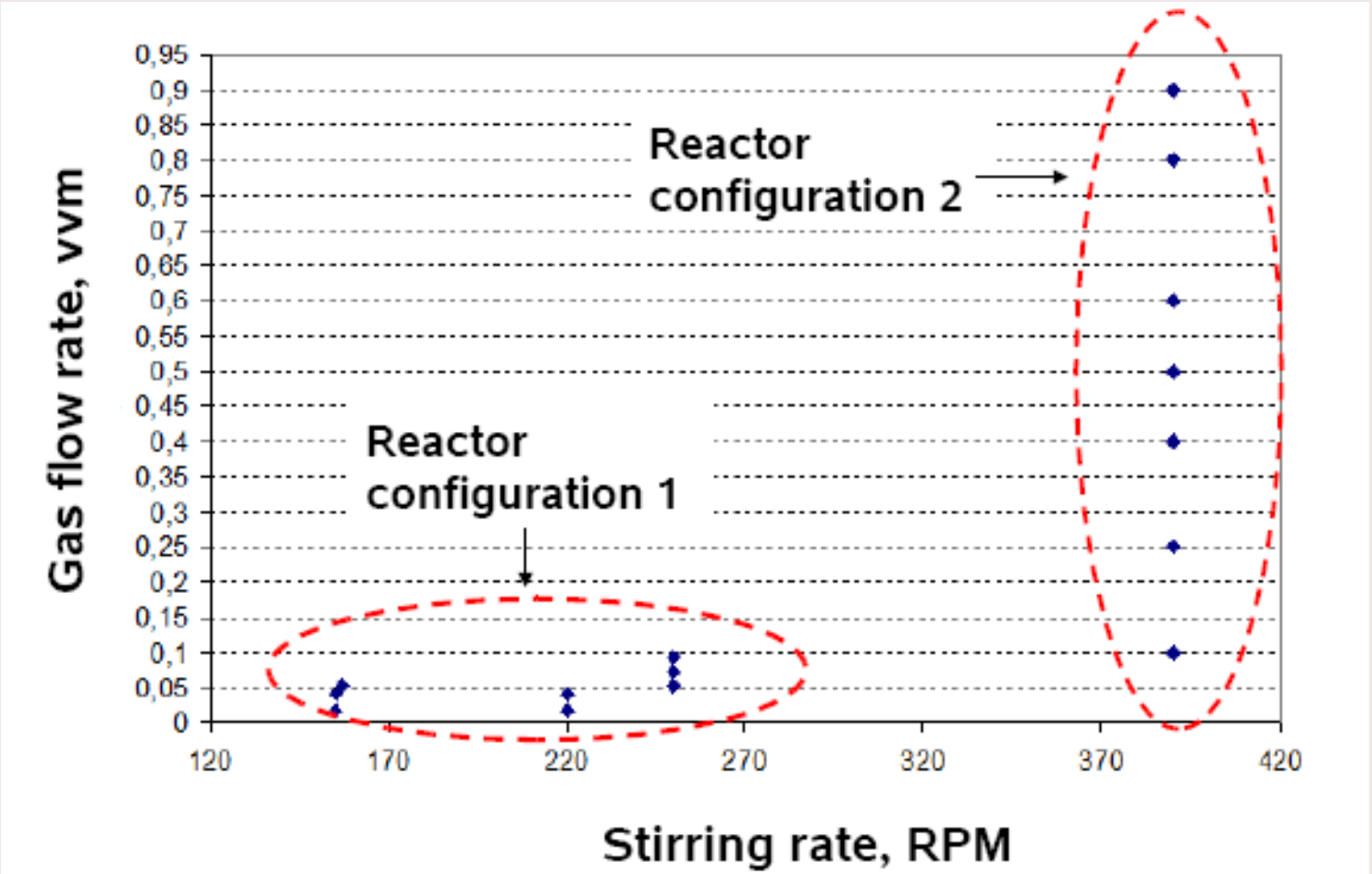
Sparger width d = 15 mm

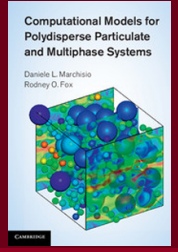




QBMM for Stirred Tanks

Different gassing and stirring rates were investigated





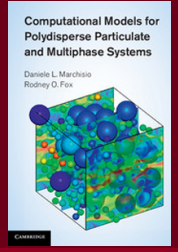
QBMM for Stirred Tanks

CQMOM: 10 additional scalars are now transported: 9 moments and 1 oxygen concentration inside liquid phase

1. 9 moments of distribution are inverted in order to find weights and nodes of quadrature

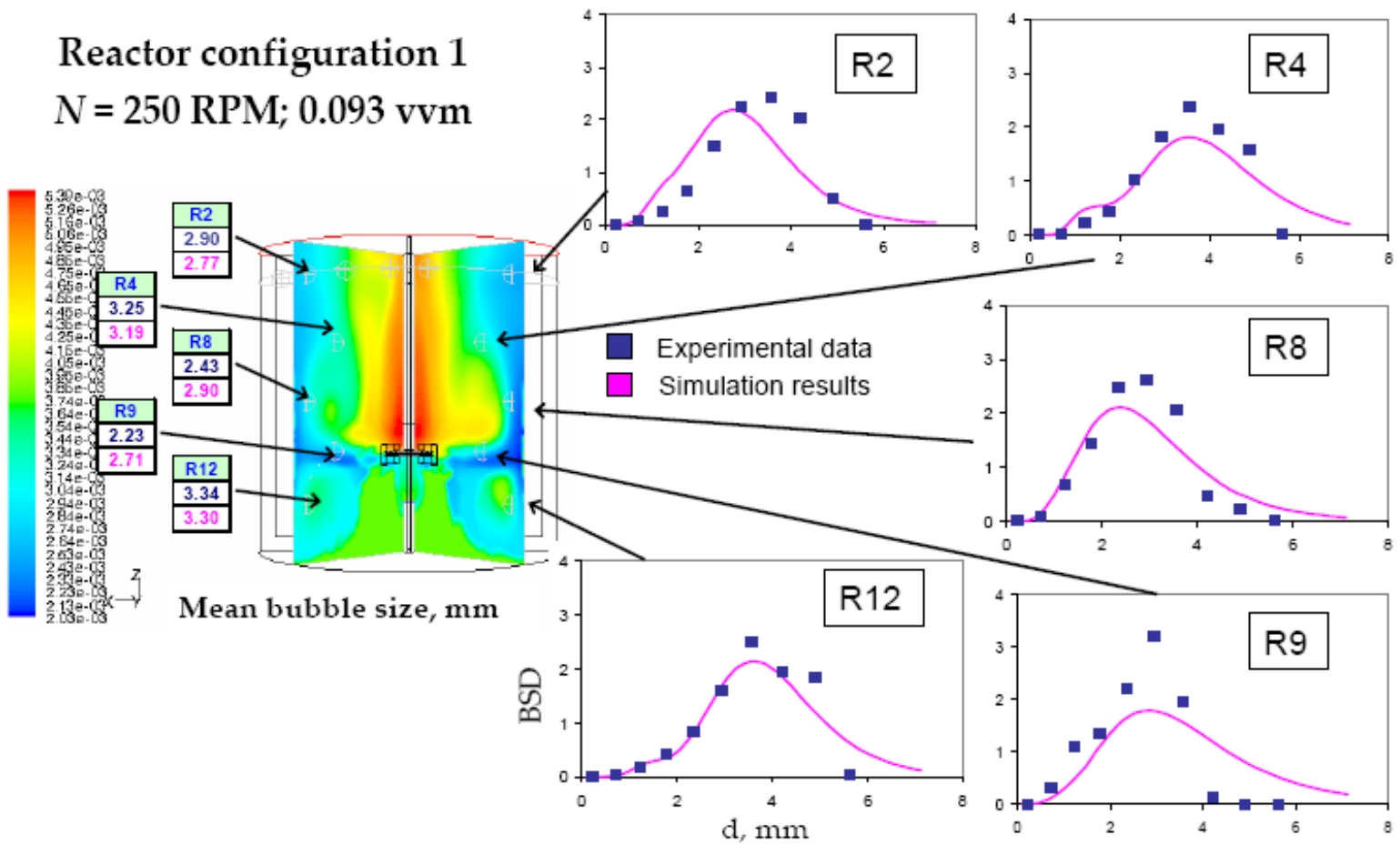
   $M_{0,0}, M_{1,0}, M_{2,0}, M_{3,0}, M_{4,0}, M_{5,0}, M_{0,1}, M_{1,1}, M_{2,1} \rightarrow w_1, w_2, w_3, L_1, L_2, L_3, \phi_1, \phi_2, \phi_3$

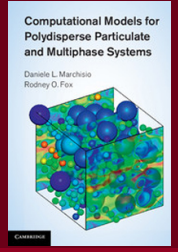
2. Source terms of moments due to coalescence, breakage and mass trasfer are calculated
3. Source term of the chemical species in liquid is calculated
4. To speed up calculation moment equations are solved at each time step whereas flow field is updated (for 2 sec) every 6 sec



QBMM for Stirred Tanks

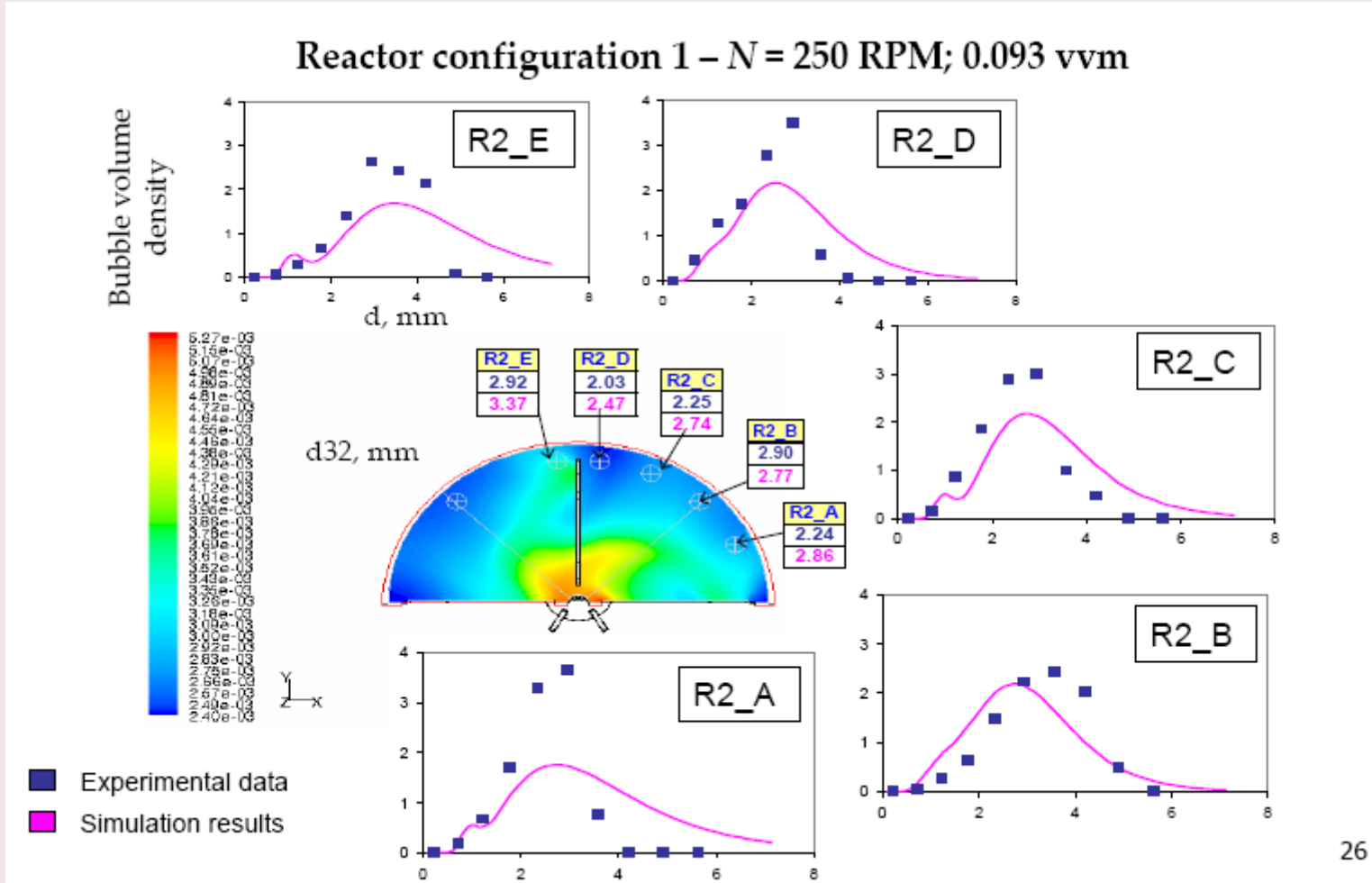
Results and comparison with experimental data

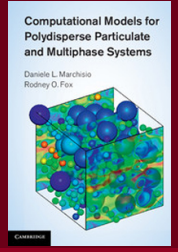




QBMM for Stirred Tanks

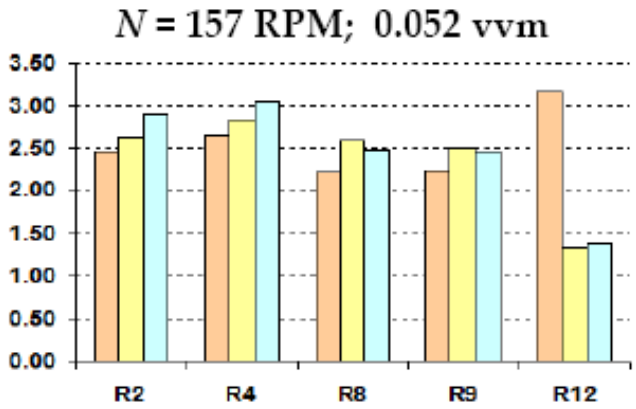
Results and comparison with experimental data

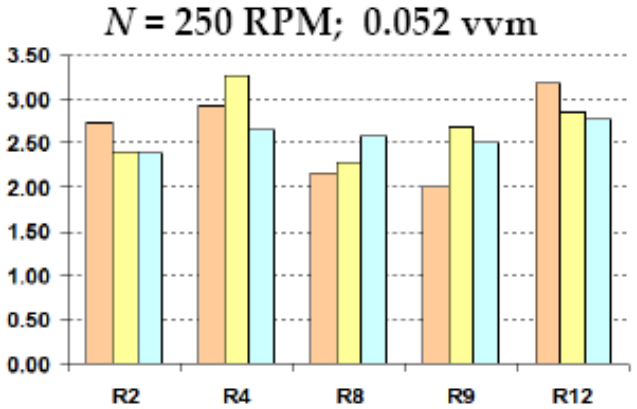


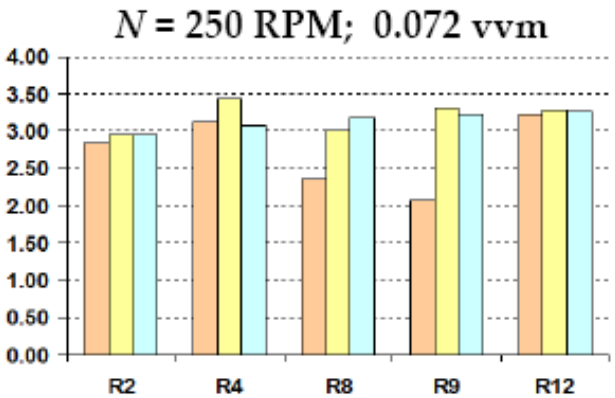


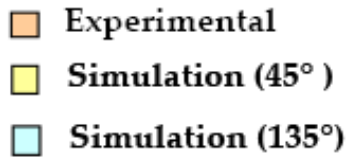
QBMM for Stirred Tanks

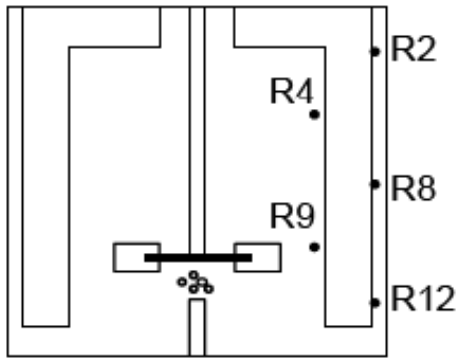
Results and comparison with experimental data for configuration 1



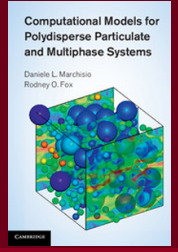








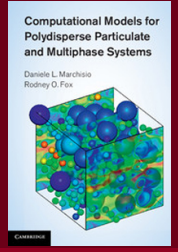
27



QBMM for Stirred Tanks

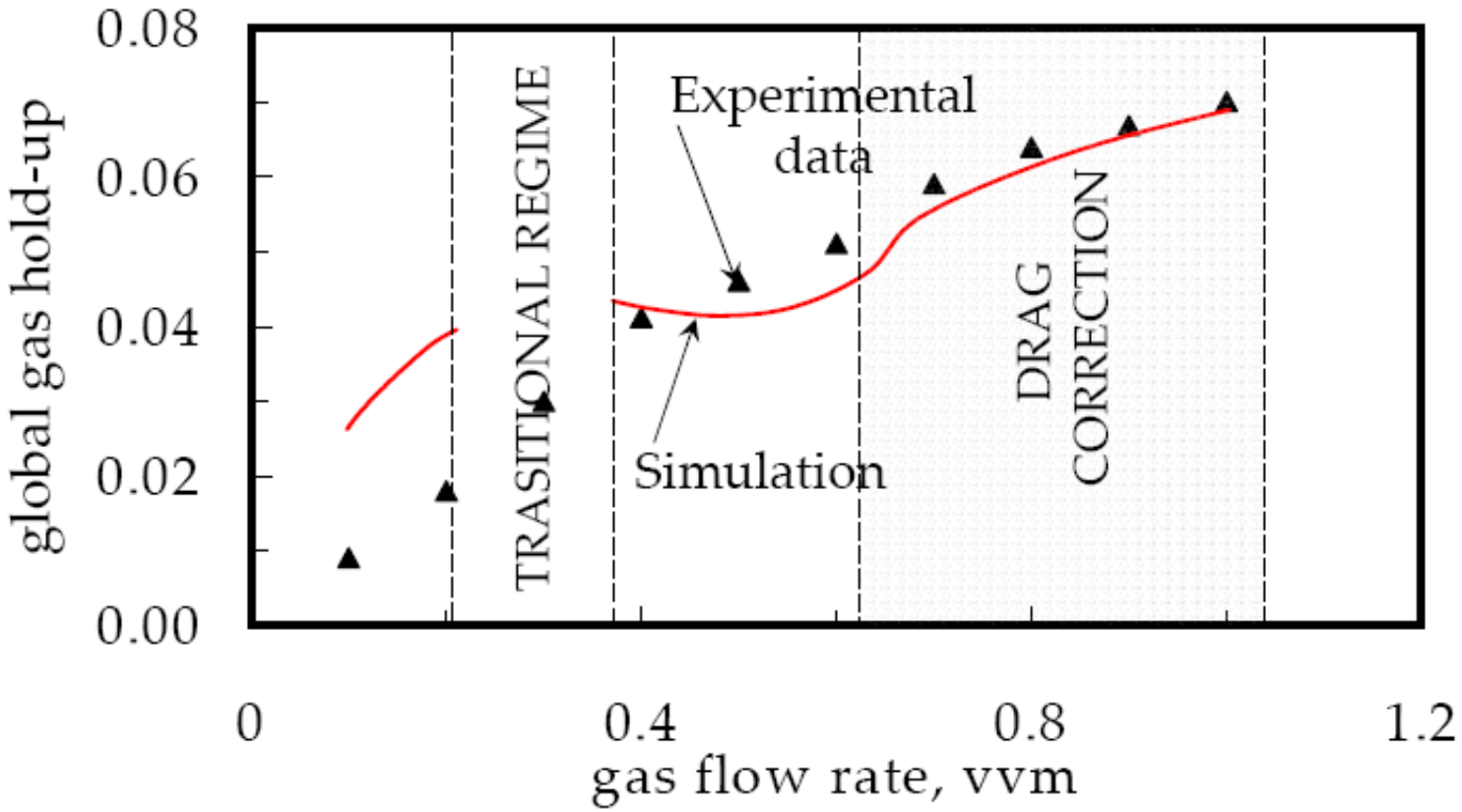
Experimental and simulated mean bubble size (mm) in five different points of the stirred tank for configuration 1

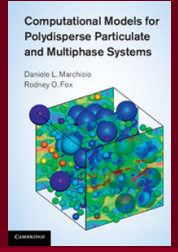
SR, RPM	GR, vvm		R2	R4	R8	R9	R12
155	0.018	Exp.	2.37	2.48	2.29	1.65	3.31
		Sim.	3.10	2.56	2.57	2.63	3.09
220	0.041	Exp.	2.56	3.34	2.57	1.76	3.81
		Sim.	2.66	3.04	2.47	2.50	3.20
220	0.052	Exp.	2.74	2.93	2.17	2.01	3.18
		Sim.	2.45	3.31	2.55	2.65	3.57
250	0.093	Exp.	2.96	3.25	2.43	2.23	3.33
		Sim.	2.56	3.27	2.59	3.05	3.35



QBMM for Stirred Tanks

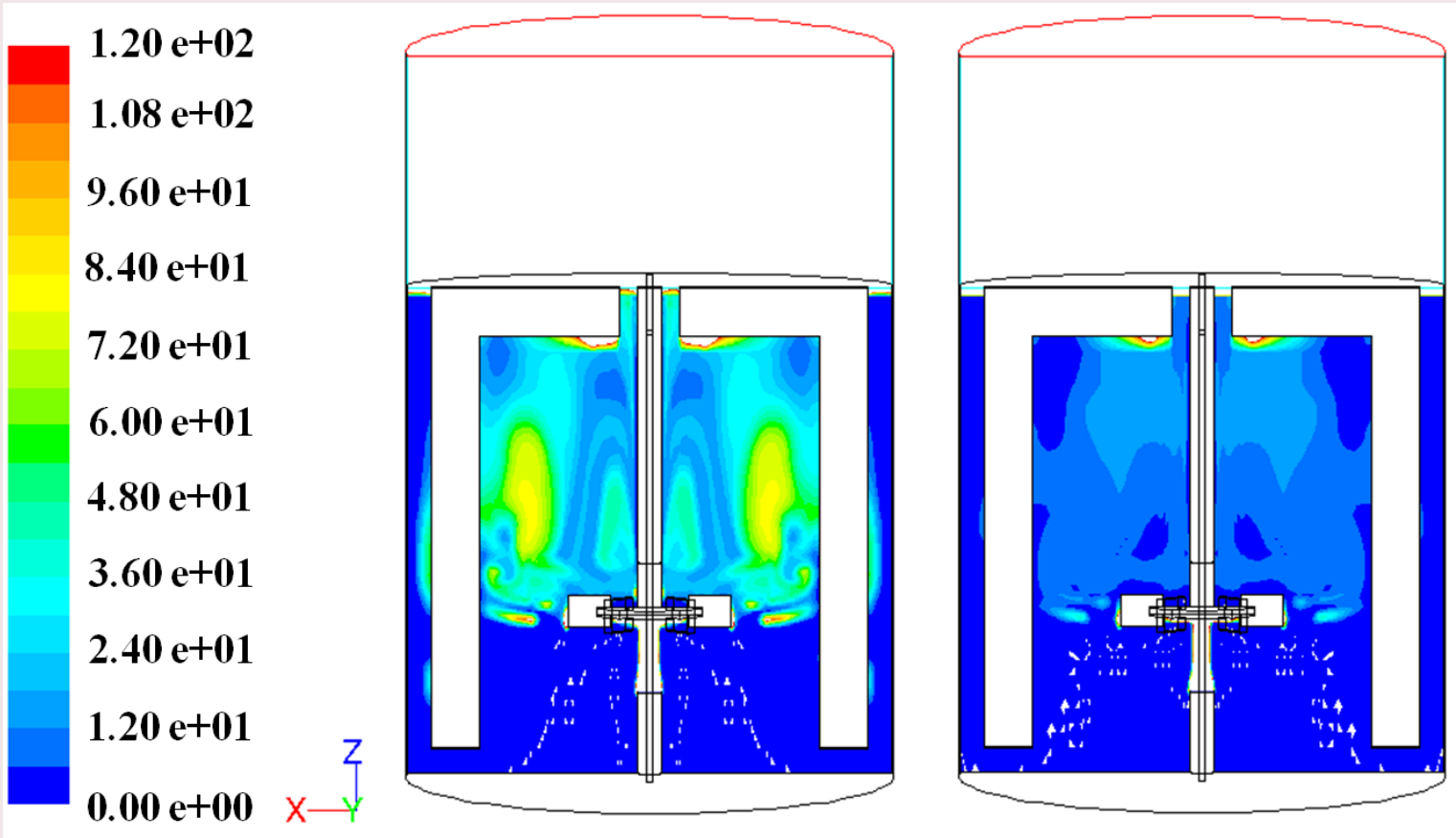
Global gas hold-up for 390 rpm for configuration 2
Drag coefficient with terminal velocity (U_t=13-8 cm/s)

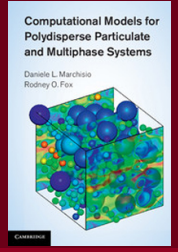




QBMM for Stirred Tanks

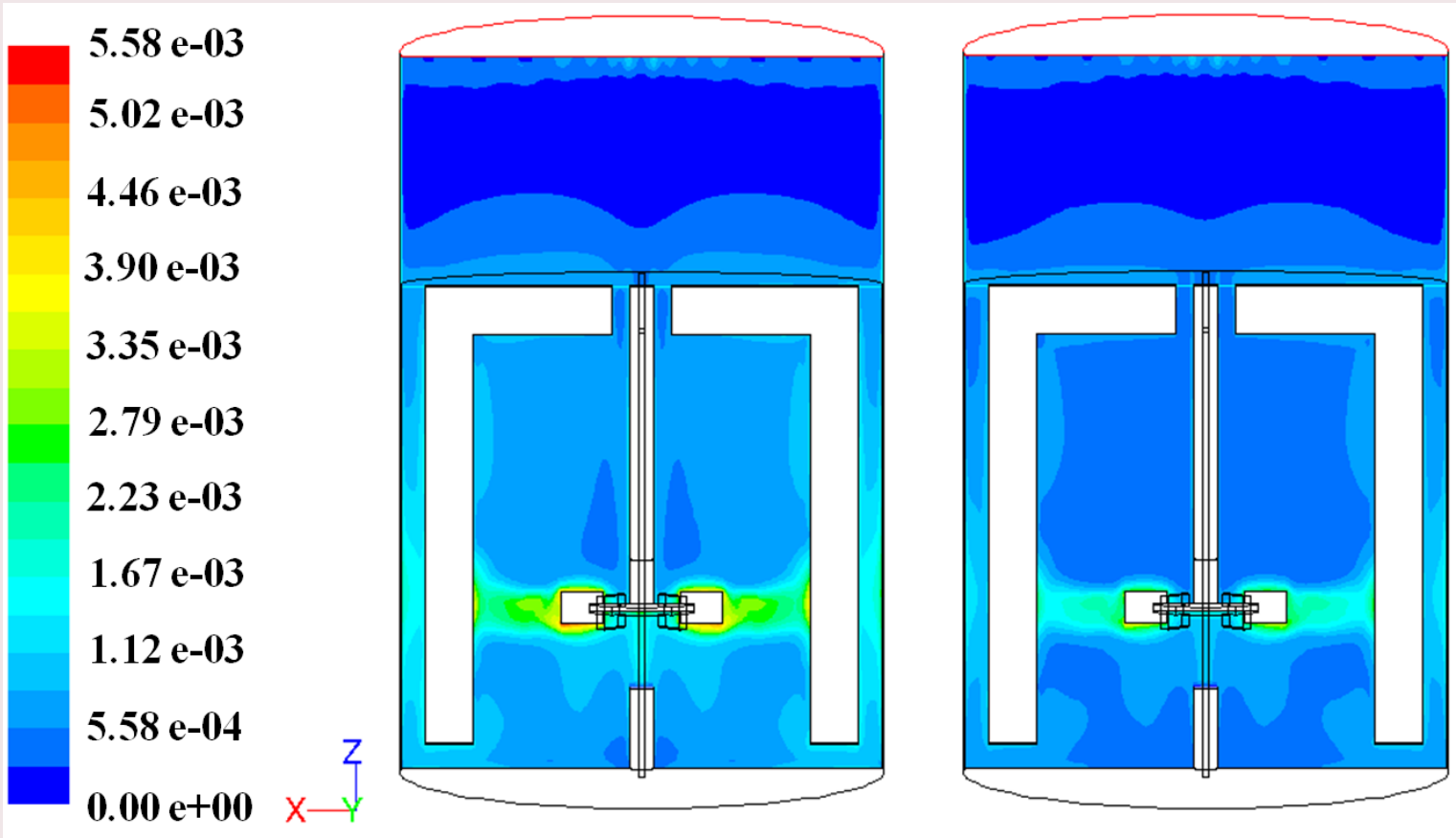
Specific surface area of bubbles (m^{-1}) at 250 rpm and 0.052 vvm (left) and at 155 rpm and 0.018 vvm (right)

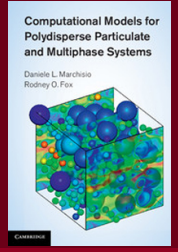




QBMM for Stirred Tanks

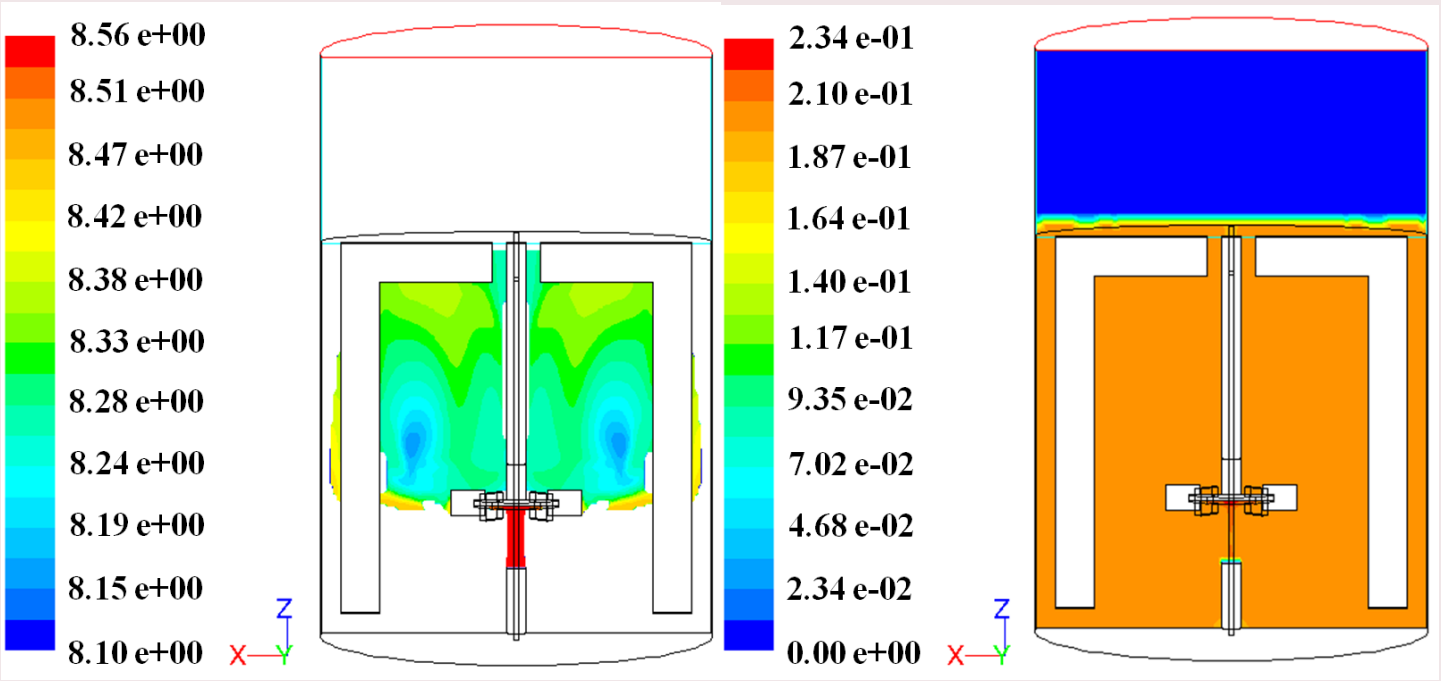
Mass transfer coefficient k_l (m s^{-1}) at 250 rpm and 0.052 vvm (left) and at 155 rpm and 0.018 vvm (right)

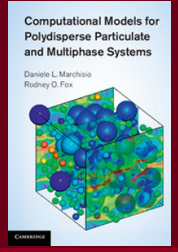




QBMM for Stirred Tanks

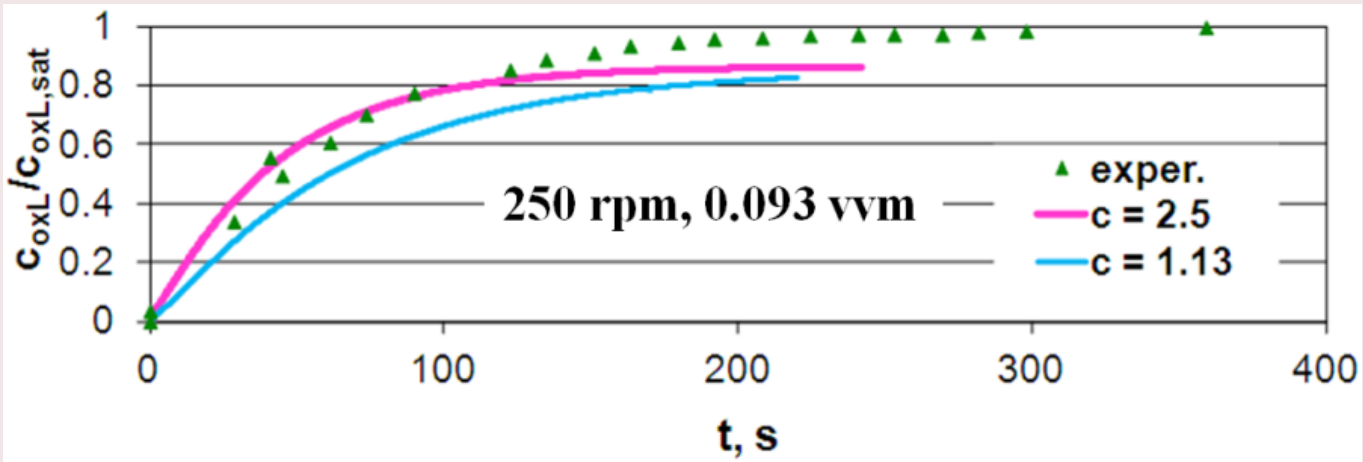
Oxygen concentration (mol m^{-3}) in gas (left) and in liquid (right) at 250 rpm and 0.052 vvm

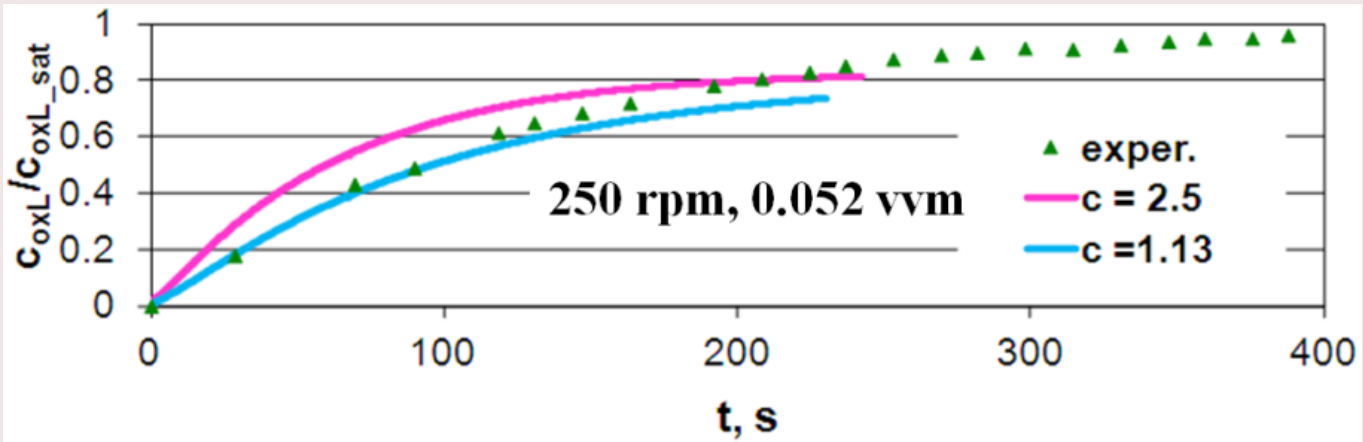


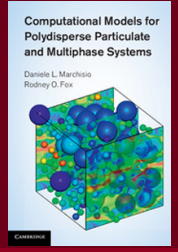


QBMM for Stirred Tanks

Time evolution of oxygen concentration in liquid

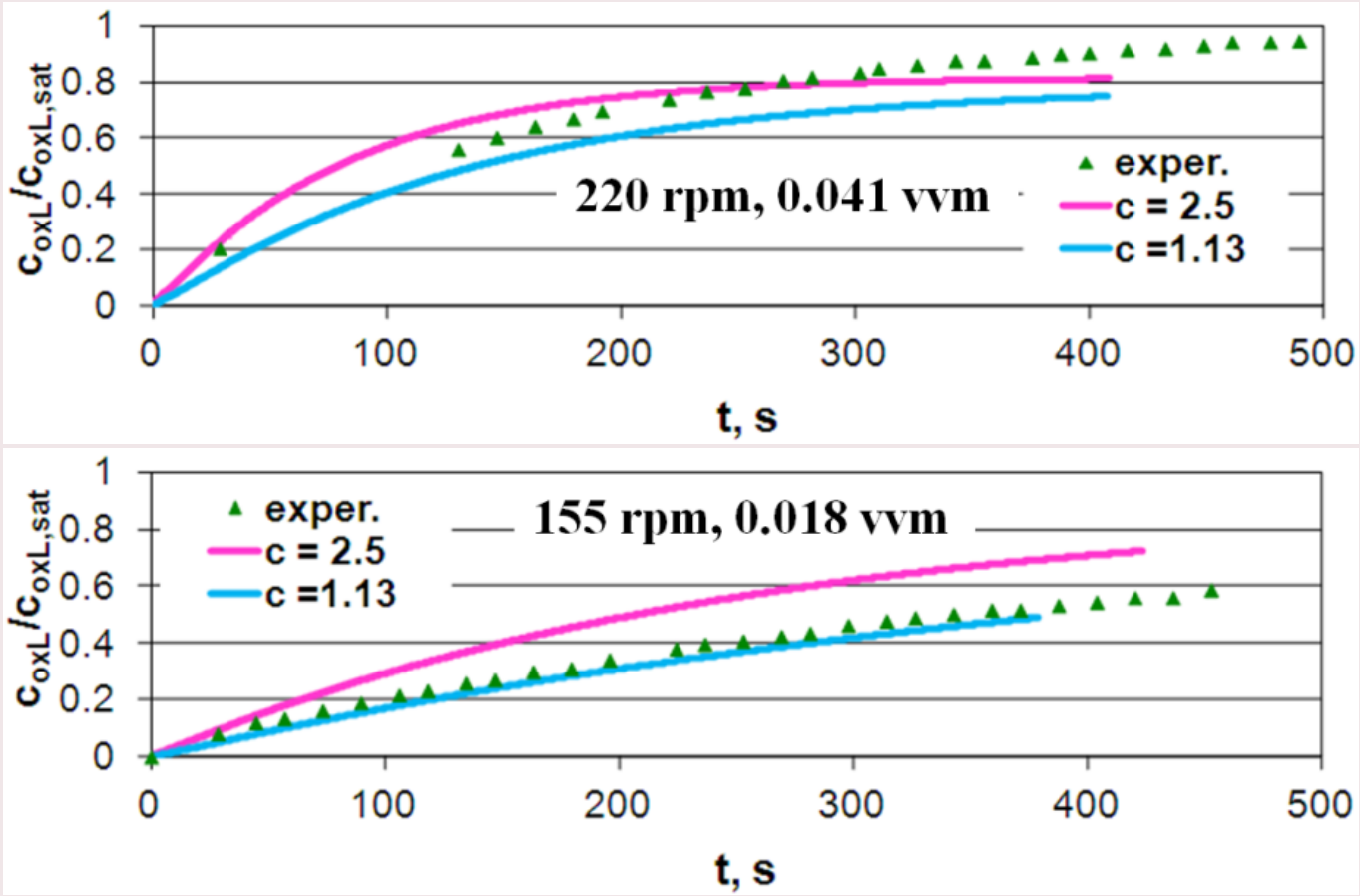


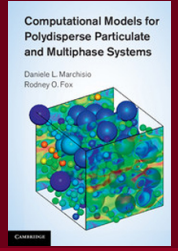




QBMM for Stirred Tanks

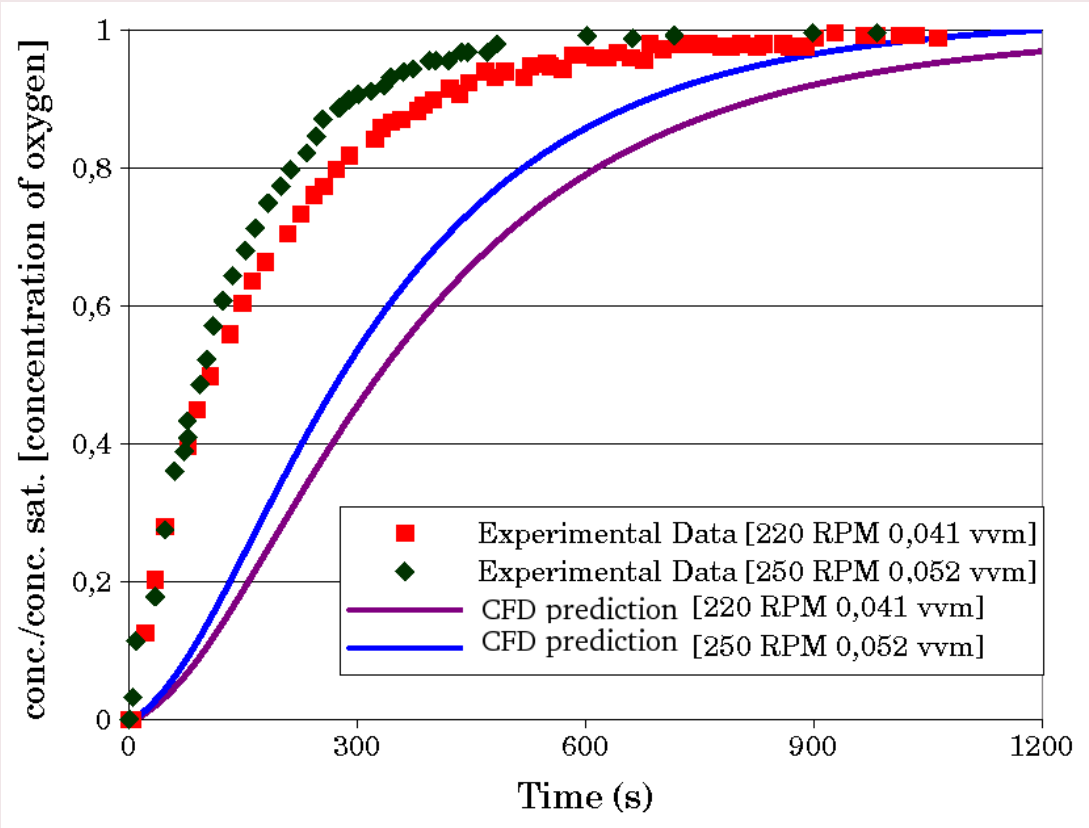
Time evolution of oxygen concentration in liquid

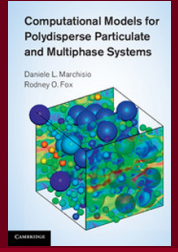




QBMM for Stirred Tanks

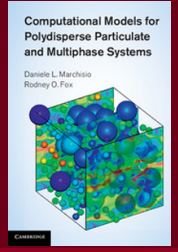
Example of poor performance of fixed-constant bubble 3-D simulation for mass transfer!





Part 6

APPLICATION TO GAS–PARTICLE FLOWS

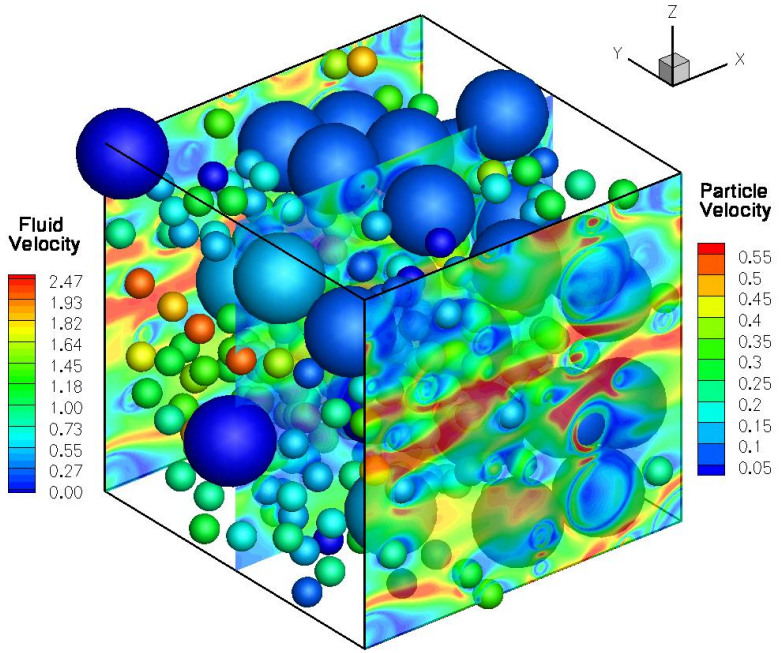


Gas–Particle Flows

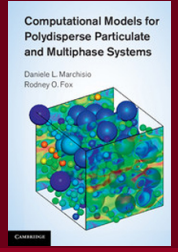
Principal characteristics

- gas phase
- particle phase
- finite particle inertia
- collisions
- variable mass loading
- size distribution
- multiphase turbulence

St_p is not negligible, must account for velocity fluctuations



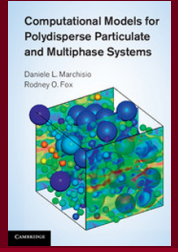
Bidisperse gas–particle flow (PR-DNS of S Subramaniam)



Modeling Challenges in Gas–Particle Flows

- Strong coupling between gas and particle phases
- Wide range of particle volume fractions (even in same flow!)
- Inertial particles with wide range of Stokes numbers
- Collision-dominated to collision-less regimes in same flow
- Granular temperature can be small and large in same flow
- Polydispersity (e.g. size, density, shape) is always present

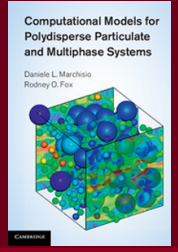
Need a modeling framework that can handle all aspects!



Modeling Challenges in Gas–Particle Flows

- Strong coupling between gas and particle phases
- Wide range of particle volume fractions (even in same flow!)
- Inertial particles with wide range of Stokes numbers
- Collision-dominated to collision-less regimes in same flow
- Granular temperature can be small and large in same flow
- Polydispersity (e.g. size, density, shape) is always present

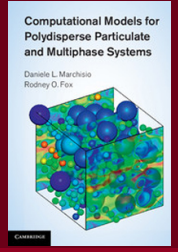
Need a modeling framework that can handle all aspects!



Modeling Challenges in Gas–Particle Flows

- Strong coupling between gas and particle phases
- Wide range of particle volume fractions (even in same flow!)
- Inertial particles with wide range of Stokes numbers
- Collision-dominated to collision-less regimes in same flow
- Granular temperature can be small and large in same flow
- Polydispersity (e.g. size, density, shape) is always present

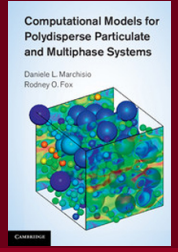
Need a modeling framework that can handle all aspects!



Modeling Challenges in Gas–Particle Flows

- Strong coupling between gas and particle phases
- Wide range of particle volume fractions (even in same flow!)
- Inertial particles with wide range of Stokes numbers
- Collision-dominated to collision-less regimes in same flow
- Granular temperature can be small and large in same flow
- Polydispersity (e.g. size, density, shape) is always present

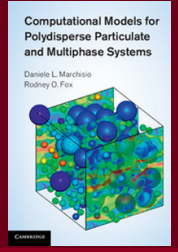
Need a modeling framework that can handle all aspects!



Modeling Challenges in Gas–Particle Flows

- Strong coupling between gas and particle phases
- Wide range of particle volume fractions (even in same flow!)
- Inertial particles with wide range of Stokes numbers
- Collision-dominated to collision-less regimes in same flow
- Granular temperature can be small and large in same flow
- Polydispersity (e.g. size, density, shape) is always present

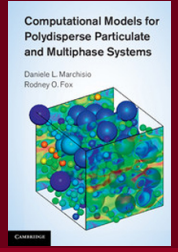
Need a modeling framework that can handle all aspects!



Modeling Challenges in Gas–Particle Flows

- Strong coupling between gas and particle phases
- Wide range of particle volume fractions (even in same flow!)
- Inertial particles with wide range of Stokes numbers
- Collision-dominated to collision-less regimes in same flow
- Granular temperature can be small and large in same flow
- Polydispersity (e.g. size, density, shape) is always present

Need a modeling framework that can handle all aspects!



Mesoscale Kinetic Equation for Particle Phase

Mesoscale equations are closed to reproduce physics from microscale

- Kinetic equation (KE) for **monodisperse** particles: $n(t, \mathbf{x}, \mathbf{v})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x_i}(v_i n) + \frac{\partial}{\partial v_i}[A_i(t, \mathbf{x}, \mathbf{v})n] = \mathbb{C}$$

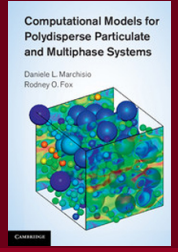
with closed acceleration $\mathbf{A}$ and collision operator $\mathbb{C}$

- Generalized population balance equation (GPBE): $n(t, \mathbf{x}, \mathbf{v}, \xi)$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x_i}(v_i n) + \frac{\partial}{\partial v_i}[A_i(t, \mathbf{x}, \mathbf{v}, \xi)n] + \frac{\partial}{\partial \xi}[G(t, \mathbf{x}, \mathbf{v}, \xi)n] = \mathbb{C}$$

with closed acceleration $\mathbf{A}$, growth $\mathbf{G}$ and collision operator $\mathbb{C}$

KE/GPBE is coupled to Navier-Stokes equation for gas phase



Mesoscale Kinetic Equation for Particle Phase

Mesoscale equations are closed to reproduce physics from microscale

- Kinetic equation (KE) for **monodisperse** particles: $n(t, \mathbf{x}, \mathbf{v})$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x_i}(v_i n) + \frac{\partial}{\partial v_i}[A_i(t, \mathbf{x}, \mathbf{v})n] = \mathbb{C}$$

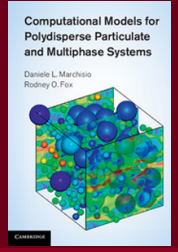
with closed acceleration $\mathbf{A}$ and collision operator $\mathbb{C}$

- Generalized population balance equation (GPBE): $n(t, \mathbf{x}, \mathbf{v}, \xi)$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x_i}(v_i n) + \frac{\partial}{\partial v_i}[A_i(t, \mathbf{x}, \mathbf{v}, \xi)n] + \frac{\partial}{\partial \xi}[G(t, \mathbf{x}, \mathbf{v}, \xi)n] = \mathbb{C}$$

with closed acceleration $\mathbf{A}$, growth $\mathbf{G}$ and collision operator $\mathbb{C}$

KE/GPBE is coupled to Navier-Stokes equation for gas phase



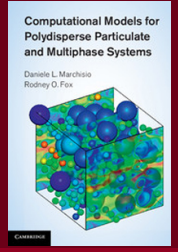
Complexity of Solutions for Gas–Particles Flows

Cluster-induced turbulence

EL simulations: J. S. Capecelatro & O. Desjardins

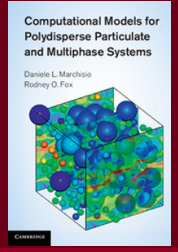
- average volume fraction: $\alpha_p = 0.01$
- $\rho_p/\rho_g = 1500$, $\mathrm{Re}_p = 1$
- elastic collisions
- full 2-way coupling

Eulerian moment model should yield identical results (if closure is accurate)



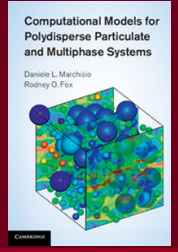
Particle Trajectory Crossing in 2-D

10-moment, 9-node CHyQMOM



Particle Trajectory Crossing in 3-D

16-moment, 27-node CHyQMOM



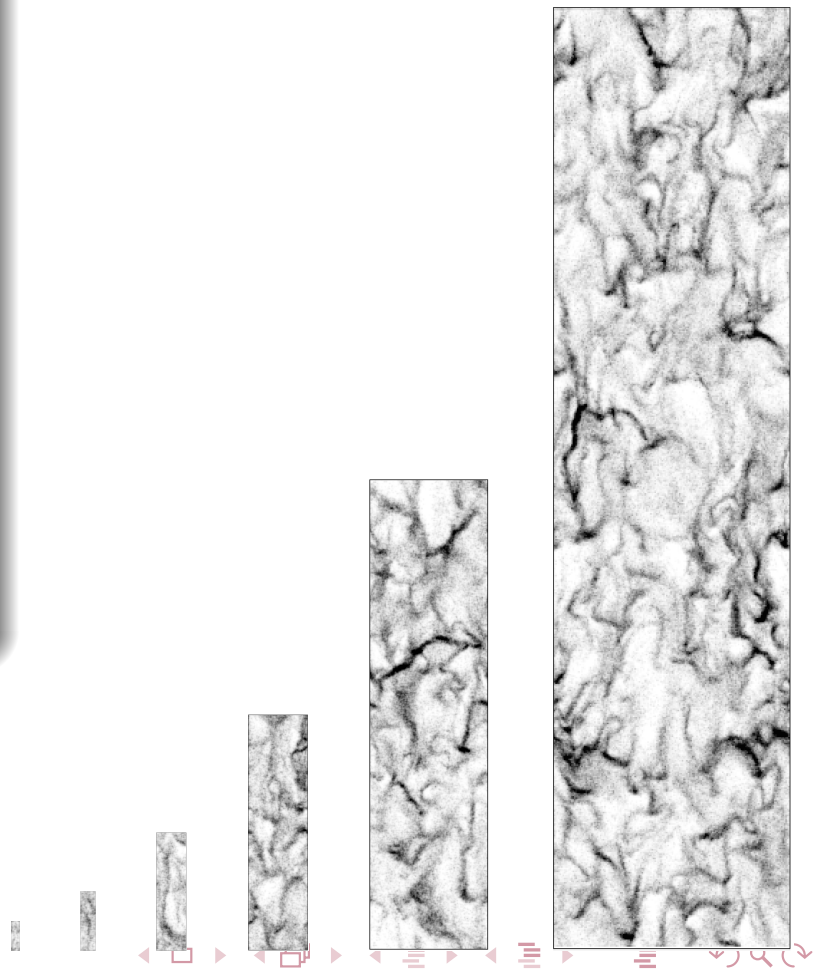
Do Eulerian and Lagrangian Models Agree?

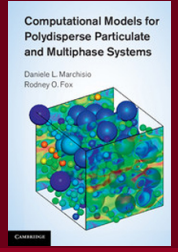
Cluster-induced turbulence

Simulation parameters

- Cases 1–6 in Capecelatro et al. JFE (2015)
- volume fraction: $\alpha_p = 0.01$
- $\rho_p/\rho_g = 1000$, $\mathrm{Re_p} = 0.5$
- terminal velocity: $\mathcal{V} = 0.1$ m/s
- cluster length: $\mathcal{L} = 2.5$ mm
- Case 6: $L_x/\mathcal{L} = 129$ $(2048 \times 512 \times 512)$

Euler-Lagrange and Euler-Euler simulations performed on same grid, but not with same numerical schemes





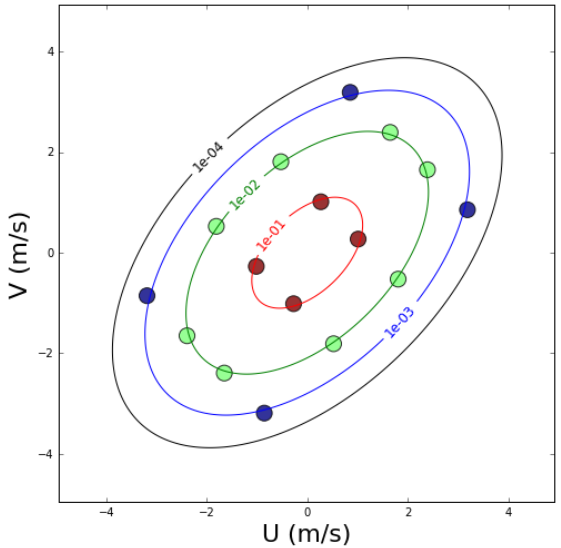
Governing Equations for Eulerian Model

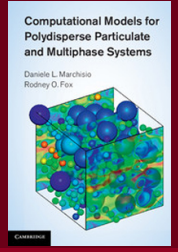
10 velocity moments in 3-D

$$\frac{\partial \mathbf{M}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{A} + \mathbf{C} \qquad M^{\gamma}_{ijk} = \int v_1^i v_2^j v_3^j f(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

$$M^0_{000} = \alpha_p \qquad \begin{bmatrix} M^1_{100} \\ M^1_{010} \\ M^1_{001} \end{bmatrix} = \alpha_p \mathbf{U}_p \qquad \begin{bmatrix} M^2_{200} & M^2_{110} & M^2_{101} \\ M^2_{110} & M^2_{020} & M^2_{011} \\ M^2_{101} & M^2_{011} & M^2_{002} \end{bmatrix} = \alpha_p (\mathbf{U}_p \otimes \mathbf{U}_p + \mathbf{P}_p)$$

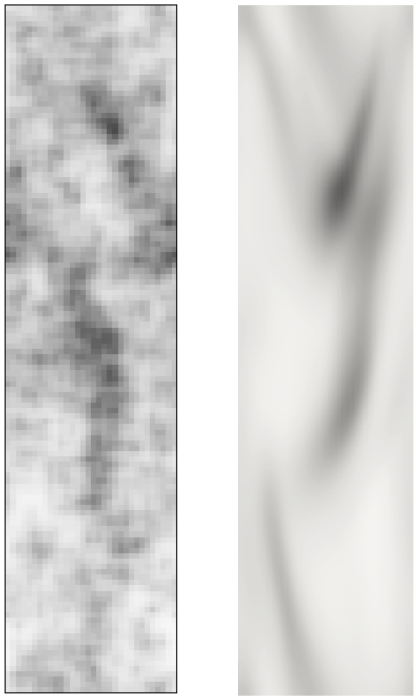
Gauss–Hermite quadrature for free-transport flux $\mathbf{F}$



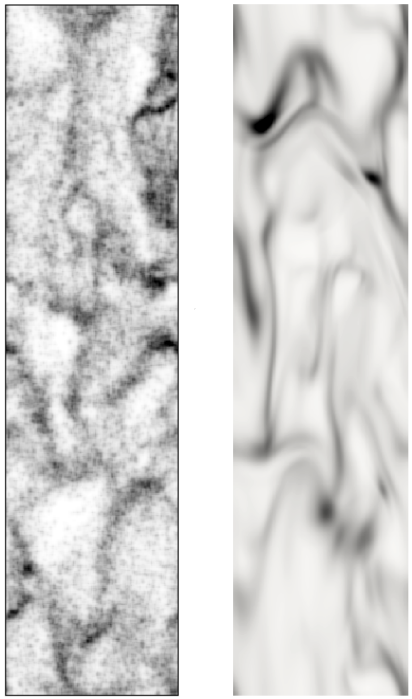


Qualitative Comparison of Clustering

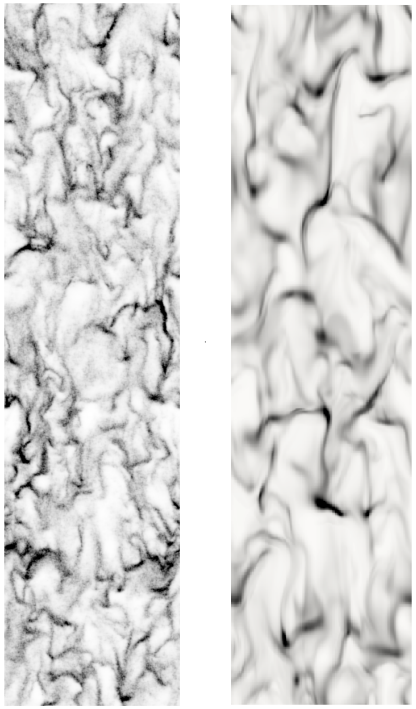
Case 2



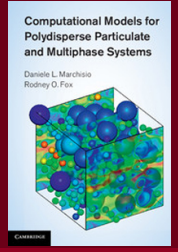
Case 4



Case 6

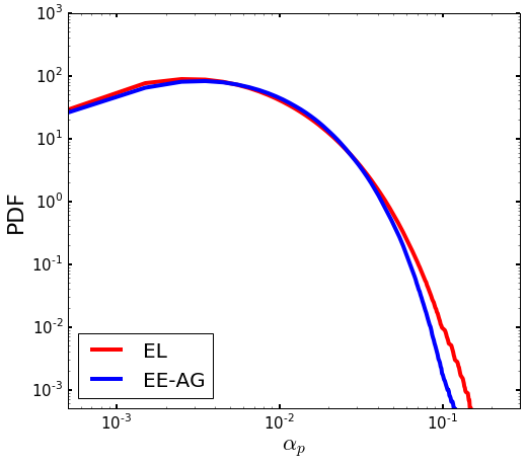


Similar shapes, but EE has slightly longer/wider clusters

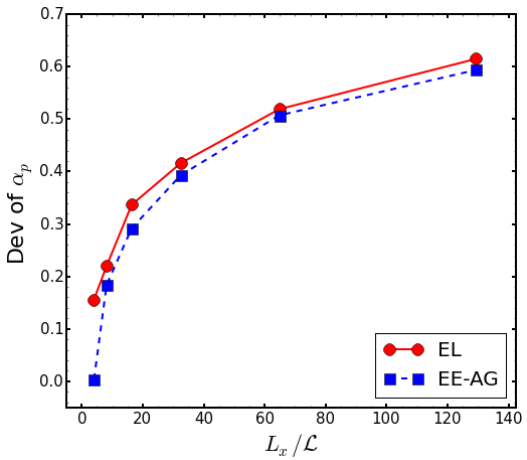


Quantitative Comparison of One-Point Statistics

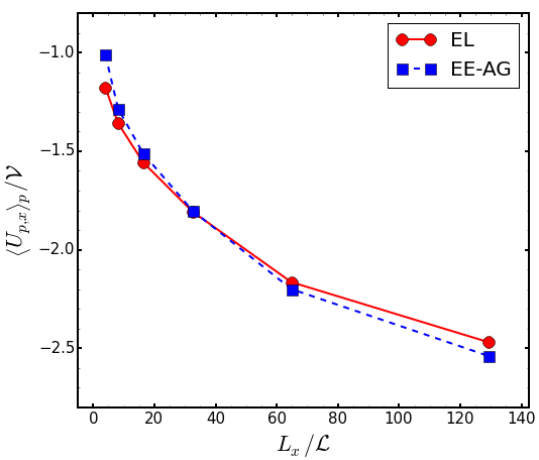
α_p PDF



Inhomogeneity

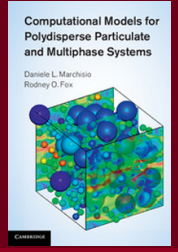


Drift velocity



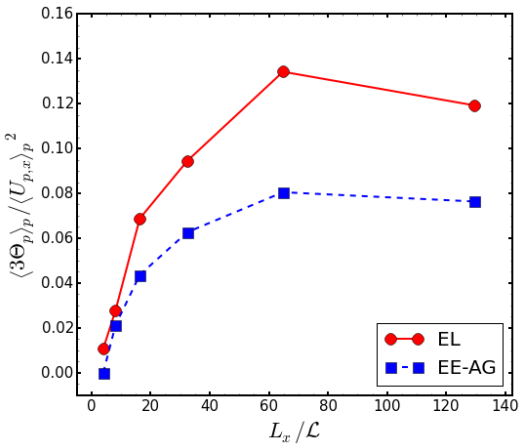
EE has slightly fewer high α_p values (due to numerics?)

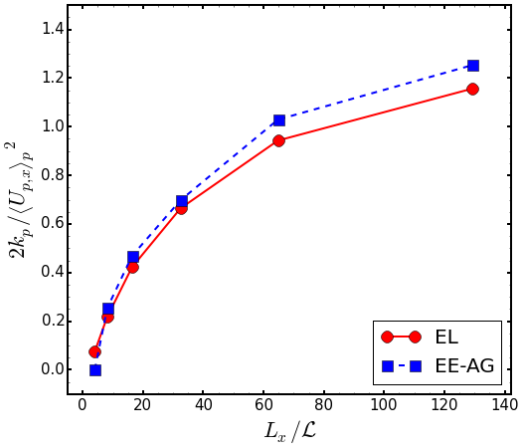
EE has slightly higher drift velocity (due to larger clusters?)

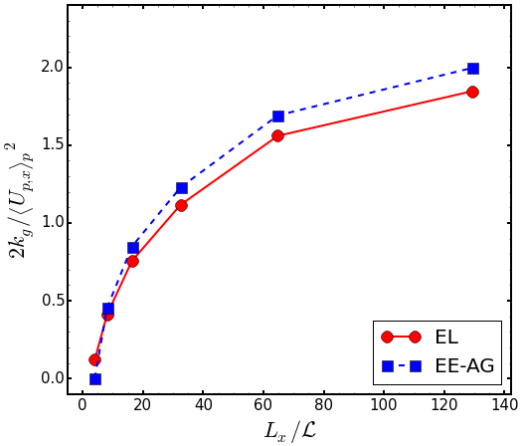


Quantitative Comparison of One-Point TKE Statistics

Θ_p k_p k_f

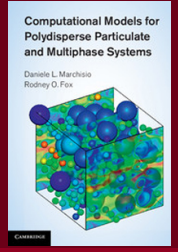






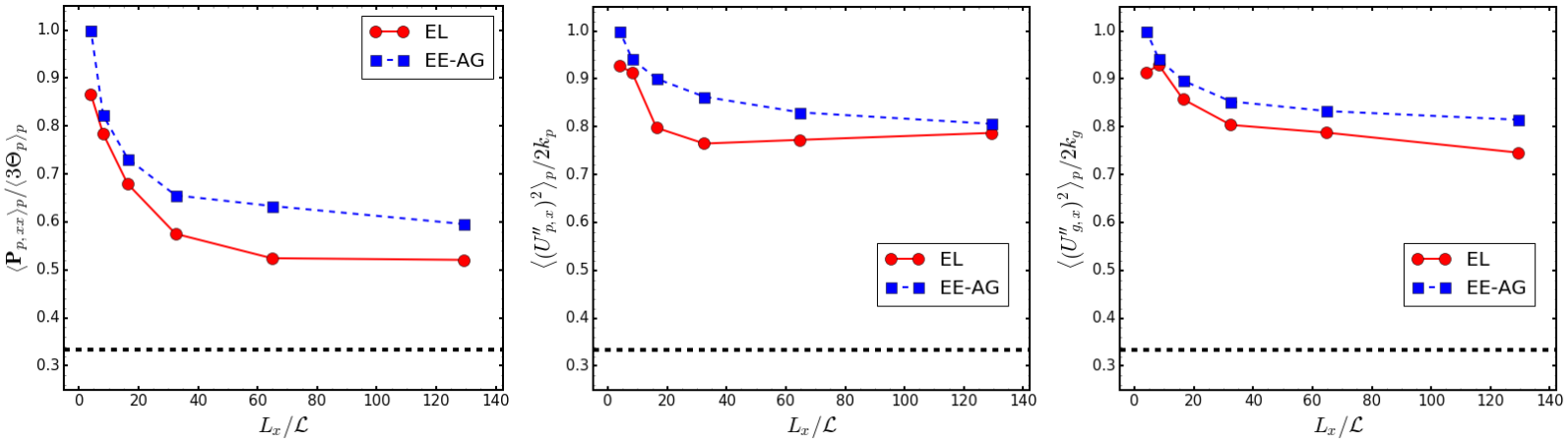
EE has lower uncorrelated TKE (due to EL post-processing?)

EE has higher correlated TKE (due to larger clusters?)



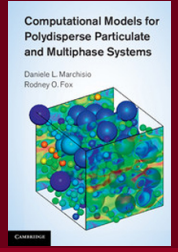
Quantitative Comparison of TKE Anisotropy

$\mathbf{P}_p$ Reynolds stress$_p$ Reynolds stress$_f$



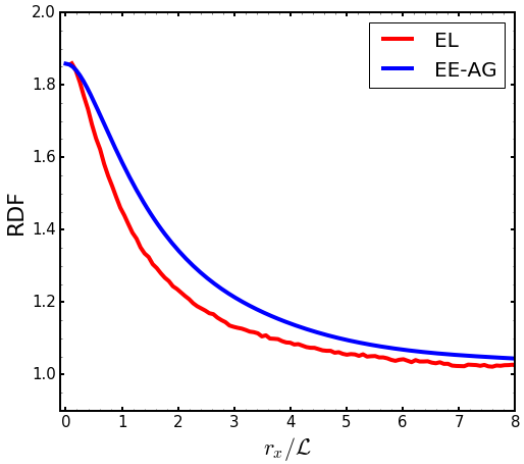
EE has slightly higher anisotropy (due to larger clusters?)

CIT is highly anisotropic $\Longrightarrow$ need full pressure tensor

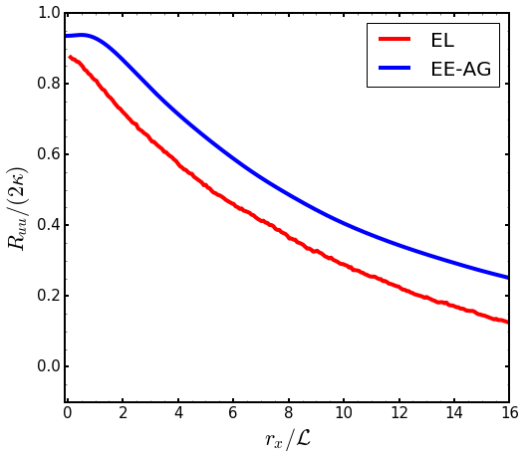


Quantitative Comparison of Two-Point Statistics

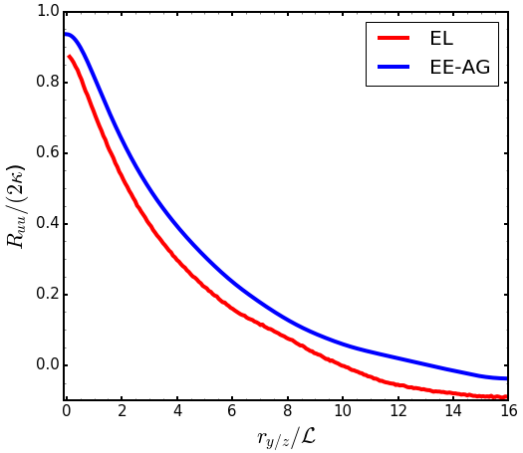
α_p vertical



U_p vertical

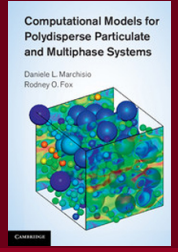


U_p horizontal



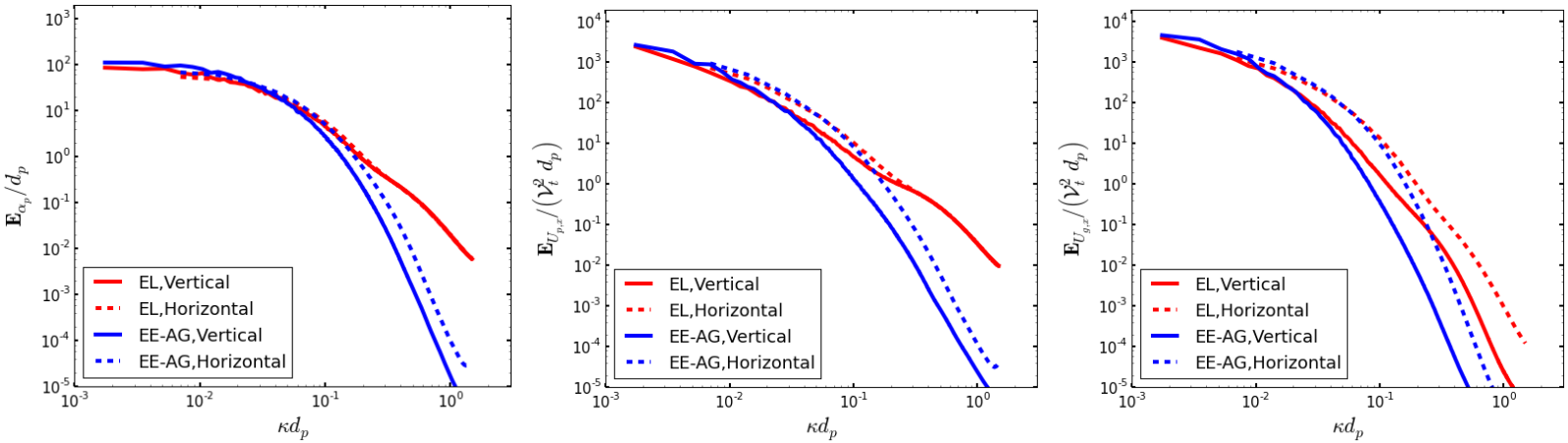
EE has longer clusters (horizontal nearly identical)

EE has larger energetic vortices (due to longer clusters?)



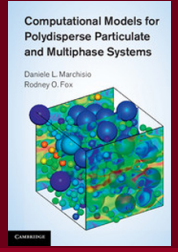
Quantitative Comparison of Energy Spectra

α_p U_p U_f



Good agreement at small wavenumbers ($\kappa d_p < 0.1$)

EE has less energy than EL at large wavenumbers (due to EL filter for coupling, numerics?)



Gas–Particle Model for Dilute to Dense Flows

Particle-phase KE

$$\frac{\partial n}{\partial t} + \mathbf{v} \cdot \frac{\partial n}{\partial \mathbf{x}} + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}n = \mathbb{C}$$

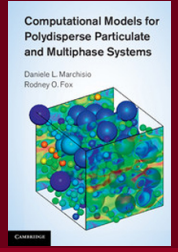
- $n(t, \mathbf{x}, \mathbf{v})$: number density function (NDF)
- $\mathbf{v}$: particle velocity
- $\mathbf{A}$: particle acceleration
- $\mathbb{C}$: rate of change of n due to Boltzmann–Enskog collisions and frictional stresses

Fluid-phase equations

$$\frac{\partial}{\partial t} \alpha_g \rho_g + \nabla \cdot \alpha_g \rho_g \mathbf{U}_g = 0$$

$$\frac{\partial}{\partial t} \alpha_g \rho_g \mathbf{U}_g + \nabla \cdot \alpha_g \rho_g \mathbf{U}_g \mathbf{U}_g = \nabla \cdot \alpha_g \boldsymbol{\tau}_g + \beta_g + \alpha_g \rho_g \mathbf{g}$$

- $\alpha_g = 1 - \alpha_p$: gas volume fraction
- β_g: mean particle drag



Gas–Particle Model for Dilute to Dense Flows

Particle-phase KE

$$\frac{\partial n}{\partial t} + \mathbf{v} \cdot \frac{\partial n}{\partial \mathbf{x}} + \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{A}n = \mathbb{C}$$

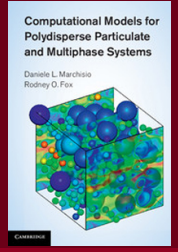
- $n(t, \mathbf{x}, \mathbf{v})$: number density function (NDF)
- $\mathbf{v}$: particle velocity
- $\mathbf{A}$: particle acceleration
- $\mathbb{C}$: rate of change of n due to Boltzmann–Enskog collisions and frictional stresses

Fluid-phase equations

$$\frac{\partial}{\partial t} \alpha_g \rho_g + \nabla \cdot \alpha_g \rho_g \mathbf{U}_g = 0$$

$$\frac{\partial}{\partial t} \alpha_g \rho_g \mathbf{U}_g + \nabla \cdot \alpha_g \rho_g \mathbf{U}_g \mathbf{U}_g = \nabla \cdot \alpha_g \boldsymbol{\tau}_g + \beta_g + \alpha_g \rho_g \mathbf{g}$$

- $\alpha_g = 1 - \alpha_p$: gas volume fraction
- β_g: mean particle drag



Governing Equations with Collisional and Frictional Fluxes

10 velocity moments in 3-D (dilute regime):

$$\frac{\partial \mathbf{M}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S} \qquad M^{\gamma}_{ijk} = \int v_1^i v_2^j v_3^j f(\mathbf{v})\, \mathrm{d}\mathbf{v}$$

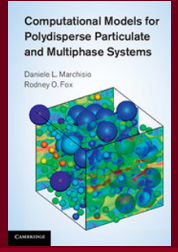
$$M^0_{000} = \alpha_p, \quad \begin{bmatrix} M^1_{100} \\ M^1_{010} \\ M^1_{001} \end{bmatrix} = \alpha_p \mathbf{U}_p, \quad \begin{bmatrix} M^2_{200} & M^2_{110} & M^2_{101} \\ M^2_{110} & M^2_{020} & M^2_{011} \\ M^2_{101} & M^2_{011} & M^2_{002} \end{bmatrix} = \alpha_p (\mathbf{U}_p \otimes \mathbf{U}_p + \mathbf{P}_p)$$

Particle-phase equations (dense regime $\Longrightarrow 3\Theta_p = trace(\mathbf{P}_p)$):

$$\frac{\partial \rho_p \alpha_p}{\partial t} + \nabla \cdot \rho_p \alpha_p \mathbf{U}_p = 0$$

$$\frac{\partial \rho_p \alpha_p \mathbf{U}_p}{\partial t} + \nabla \cdot \rho_p \alpha_p \left(\mathbf{U}_p \otimes \mathbf{U}_p + \mathbf{P}_p + \mathbf{G}_p + \mathbf{Z}_p \right) = \rho_p \alpha_p \mathbf{g} + \rho_p \alpha_p \mathbf{M}_{pg}$$

$$\frac{\partial \rho_p \alpha_p \mathbf{P}_p}{\partial t} + \nabla \cdot \rho_p \alpha_p \left(\mathbf{U}_p \otimes \mathbf{P}_p + \mathbf{Q}_p + \mathbf{H}_p \right) + \rho_p \alpha_p \left[(\mathbf{P}_p + \mathbf{G}_p) \cdot \nabla \mathbf{U}_p + \left(\nabla \mathbf{U}_p \right)^T \cdot (\mathbf{P}_p + \mathbf{G}_p) \right]$$
$$= \rho_p \alpha_p \mathbf{E}_{pg} + \rho_p \alpha_p \mathbf{C}_p$$



Kinetic flux:

$$\mathbf{U}_p \otimes \mathbf{U}_p + \mathbf{P}_p$$

$$\mathbf{P}_p = \Theta_p \mathbf{I} - \boldsymbol{\sigma}_p = \Theta_p \mathbf{I} - 2\nu_{p,k}\mathbf{S}_p$$

$$\mathbf{S}_p = \frac{1}{2}\left[\nabla \mathbf{U}_p + \left(\nabla \mathbf{U}_p\right)^T - \frac{2}{3}\left(\nabla \cdot \mathbf{U}_p\right)\mathbf{I}\right]$$

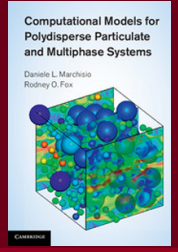
Collisional flux (pressure infinite for finite $\alpha_p \approx 0.63$):

$$\mathbf{G}_p = \frac{p_{p,c}}{\rho_p \alpha_p}\mathbf{I} - 2\nu_{p,c}\mathbf{S}_p$$

Frictional flux (pressure infinite for finite $\alpha_p \approx 0.63$, null when $\alpha_p < 0.55$):

$$\mathbf{Z}_p = \frac{p_{p,f}}{\rho_p \alpha_p}\mathbf{I} - 2\nu_{p,f}\mathbf{S}_p$$

Energy fluxes: $\mathbf{U}_p \otimes \mathbf{P}_p + \mathbf{Q}_p + \mathbf{H}_p = \mathbf{U}_p \otimes \mathbf{P}_p - \frac{2}{3}k_\Theta \nabla \otimes \mathbf{P}_p$

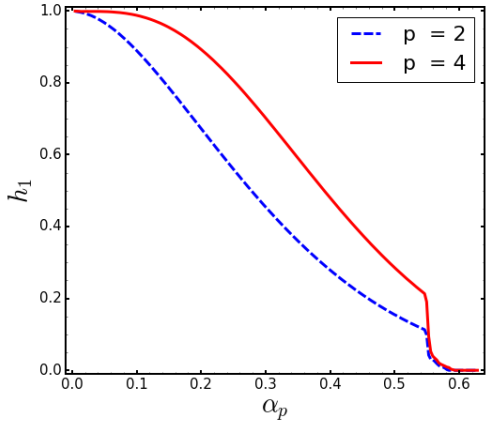


Kinetic Flux-Splitting Scheme for All Flow Regimes

$$\frac{\partial \mathbf{M}}{\partial t} + \nabla \cdot h_1 \mathbf{F} + \nabla \cdot (h_2 \mathbf{F} + \mathbf{G} + \mathbf{Z}) = \mathbf{S}$$

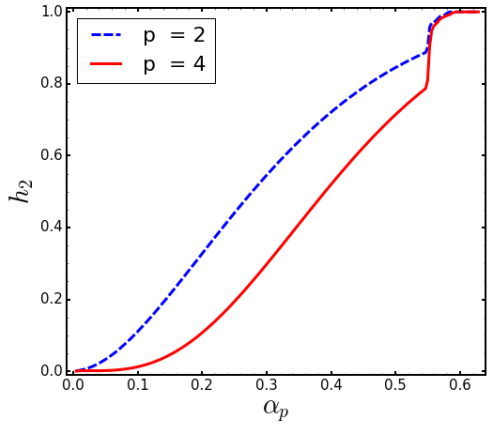
$$h_1 + h_2 = 1$$

$$h_2 = \left(\frac{p_{p,c} + p_{p,f}}{p_{p,k} + p_{p,c} + p_{p,f} + \varepsilon} \right)^{\mathrm{p}}$$



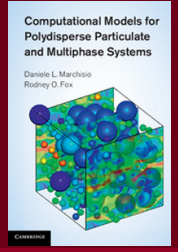
Step 1: KBFVM

$$\frac{\partial \mathbf{M}}{\partial t} + \nabla \cdot h_1 \mathbf{F} = \mathbf{0}$$



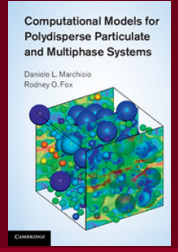
Step 2: Hydrodynamic solver

$$\frac{\partial \mathbf{M}}{\partial t} + \nabla \cdot (h_2 \mathbf{F} + \mathbf{G} + \mathbf{Z}) = \mathbf{S}$$



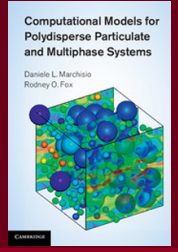
Solution Procedure

1. Initialize all variables $\mathbf{M}$, $\{\alpha_p, \mathbf{U}_p, \Theta_p, \sigma_p\}$, and $\{\alpha_g, \mathbf{U}_g, p_g\}$
2. Calculate h_1 and h_2
3. **Explicit** Free-transport solver:
   - Compute kinetic-based moment fluxes to transport the moments
   - Update $\{\alpha_p, \mathbf{U}_p, \Theta_p, \sigma_p\}$ using moments $\mathbf{M}$
4. **Iterative** Hydrodynamic solver:
   - Solve $\{\alpha_p, \mathbf{U}_p, \Theta_p\}$ hydrodynamic transport equations
   - Solve gas-phase velocity and pressure, $\{\mathbf{U}_g, p_g\}$, equations
5. Solve σ_p transport equation
6. Update moment set $\mathbf{M}$ using $\{\alpha_p, \mathbf{U}_p, \Theta_p, \sigma_p\}$
7. Advance in time by repeating from Step 2 until simulation is complete



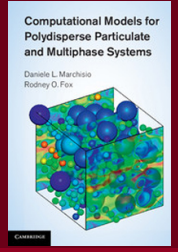
Solution Procedure

1. Initialize all variables $\mathbf{M}$, $\{\alpha_p, \mathbf{U}_p, \Theta_p, \sigma_p\}$, and $\{\alpha_g, \mathbf{U}_g, p_g\}$
2. Calculate h_1 and h_2
3. **Explicit** Free-transport solver:
   - Compute kinetic-based moment fluxes to transport the moments
   - Update $\{\alpha_p, \mathbf{U}_p, \Theta_p, \sigma_p\}$ using moments $\mathbf{M}$
4. **Iterative** Hydrodynamic solver:
   - Solve $\{\alpha_p, \mathbf{U}_p, \Theta_p\}$ hydrodynamic transport equations
   - Solve gas-phase velocity and pressure, $\{\mathbf{U}_g, p_g\}$, equations
5. Solve σ_p transport equation
6. Update moment set $\mathbf{M}$ using $\{\alpha_p, \mathbf{U}_p, \Theta_p, \sigma_p\}$
7. Advance in time by repeating from Step 2 until simulation is complete



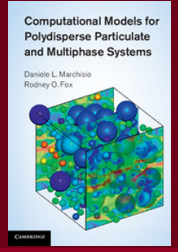
Solution Procedure

1. Initialize all variables $\mathbf{M}$, $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$, and $\{\alpha_g, \mathbf{U}_g, p_g\}$
2. Calculate h_1 and h_2
3. **Explicit** Free-transport solver:
   - Compute kinetic-based moment fluxes to transport the moments
   - Update $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$ using moments $\mathbf{M}$
4. **Iterative** Hydrodynamic solver:
   - Solve $\{\alpha_p, \mathbf{U}_p, \Theta_p\}$ hydrodynamic transport equations
   - Solve gas-phase velocity and pressure, $\{\mathbf{U}_g, p_g\}$, equations
5. Solve σ_p transport equation
6. Update moment set $\mathbf{M}$ using $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$
7. Advance in time by repeating from Step 2 until simulation is complete



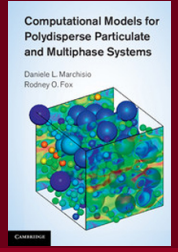
Solution Procedure

1. Initialize all variables $\mathbf{M}$, $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$, and $\{\alpha_g, \mathbf{U}_g, p_g\}$
2. Calculate h_1 and h_2
3. **Explicit** Free-transport solver:
   - Compute kinetic-based moment fluxes to transport the moments
   - Update $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$ using moments $\mathbf{M}$
4. **Iterative** Hydrodynamic solver:
   - Solve $\{\alpha_p, \mathbf{U}_p, \Theta_p\}$ hydrodynamic transport equations
   - Solve gas-phase velocity and pressure, $\{\mathbf{U}_g, p_g\}$, equations
5. Solve σ_p transport equation
6. Update moment set $\mathbf{M}$ using $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$
7. Advance in time by repeating from Step 2 until simulation is complete



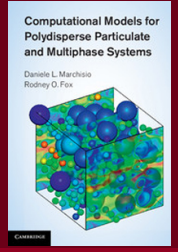
Solution Procedure

1. Initialize all variables $\mathbf{M}$, $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$, and $\{\alpha_g, \mathbf{U}_g, p_g\}$
2. Calculate h_1 and h_2
3. **Explicit** Free-transport solver:
   - Compute kinetic-based moment fluxes to transport the moments
   - Update $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$ using moments $\mathbf{M}$
4. **Iterative** Hydrodynamic solver:
   - Solve $\{\alpha_p, \mathbf{U}_p, \Theta_p\}$ hydrodynamic transport equations
   - Solve gas-phase velocity and pressure, $\{\mathbf{U}_g, p_g\}$, equations
5. Solve $\boldsymbol{\sigma}_p$ transport equation
6. Update moment set $\mathbf{M}$ using $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$
7. Advance in time by repeating from Step 2 until simulation is complete



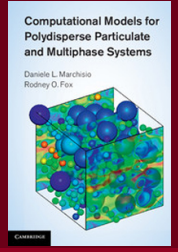
Solution Procedure

1. Initialize all variables $\mathbf{M}$, $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$, and $\{\alpha_g, \mathbf{U}_g, p_g\}$
2. Calculate h_1 and h_2
3. **Explicit** Free-transport solver:
   - Compute kinetic-based moment fluxes to transport the moments
   - Update $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$ using moments $\mathbf{M}$
4. **Iterative** Hydrodynamic solver:
   - Solve $\{\alpha_p, \mathbf{U}_p, \Theta_p\}$ hydrodynamic transport equations
   - Solve gas-phase velocity and pressure, $\{\mathbf{U}_g, p_g\}$, equations
5. Solve $\boldsymbol{\sigma}_p$ transport equation
6. Update moment set $\mathbf{M}$ using $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$
7. Advance in time by repeating from Step 2 until simulation is complete



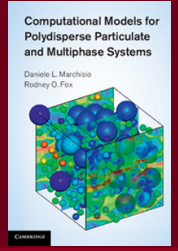
Solution Procedure

1. Initialize all variables $\mathbf{M}$, $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$, and $\{\alpha_g, \mathbf{U}_g, p_g\}$
2. Calculate h_1 and h_2
3. **Explicit** Free-transport solver:
   - Compute kinetic-based moment fluxes to transport the moments
   - Update $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$ using moments $\mathbf{M}$
4. **Iterative** Hydrodynamic solver:
   - Solve $\{\alpha_p, \mathbf{U}_p, \Theta_p\}$ hydrodynamic transport equations
   - Solve gas-phase velocity and pressure, $\{\mathbf{U}_g, p_g\}$, equations
5. Solve $\boldsymbol{\sigma}_p$ transport equation
6. Update moment set $\mathbf{M}$ using $\{\alpha_p, \mathbf{U}_p, \Theta_p, \boldsymbol{\sigma}_p\}$
7. Advance in time by repeating from Step 2 until simulation is complete



2-D Bubbling Fluidized Bed

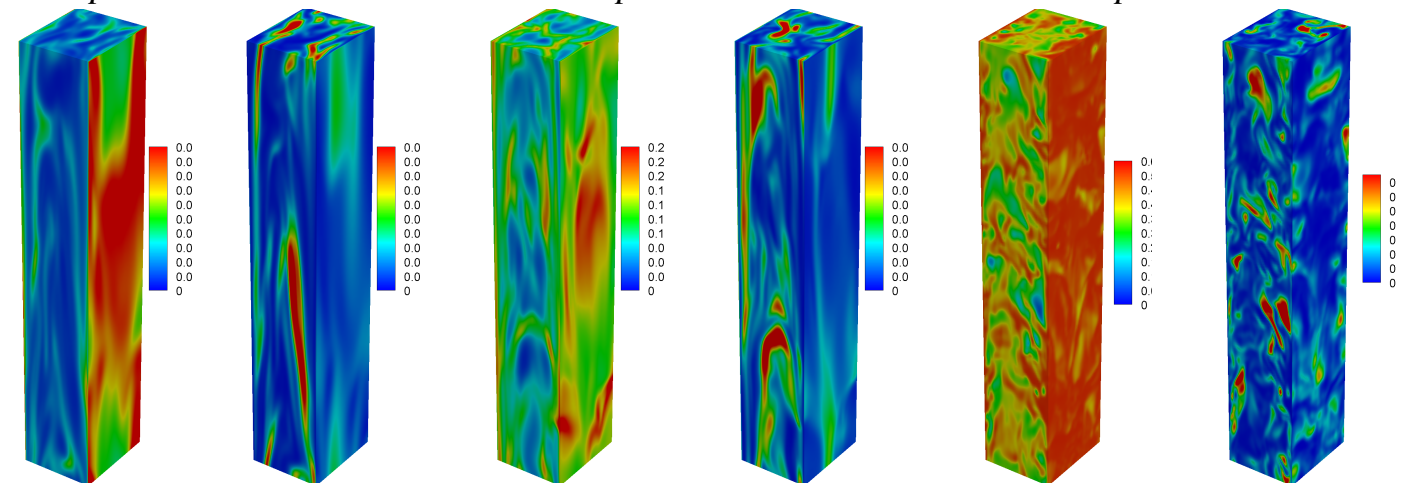
α_p h_2 $U_{p,y}$ Θ_p $\sigma_{p,xy}$



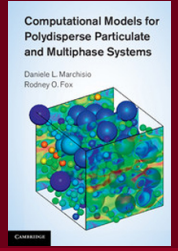
3-D Wall-Bounded Vertical Channel

Instantaneous Fields

$\overline{\alpha}_p = 0.01$ $\overline{\alpha}_p = 0.1$ $\overline{\alpha}_p = 0.4$



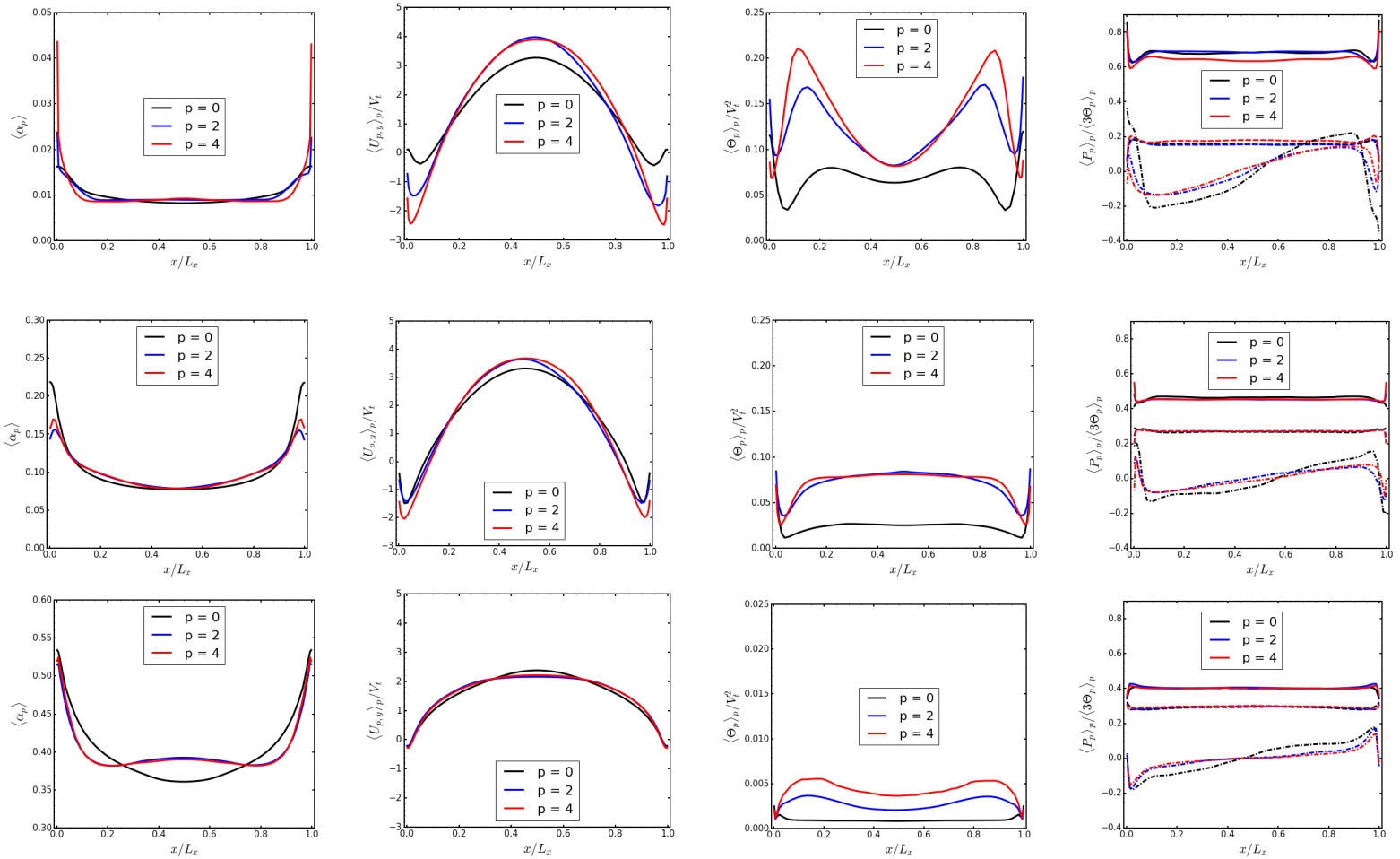
α_p Θ_p α_p Θ_p α_p Θ_p

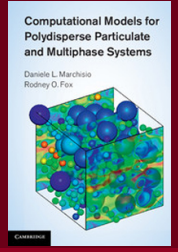


Wall-Bounded Vertical Channel

Statistical Results

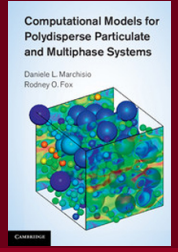
$\langle \alpha_p \rangle$ $\langle U_{p,y} \rangle_p / V_t$ $\langle \Theta_p \rangle_p / V_t^2$ $\langle P_p \rangle_p / \langle 3\Theta_p \rangle_p$





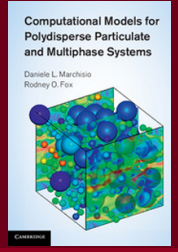
Summary for Gas–Particle Flows

- Quadrature-based moment methods solve kinetic equation by reconstructing velocity distribution function from moments
- For dilute flows, simulations use velocity NDF reconstruction and kinetic-based finite-volume methods
- For dense flows, use KBFVM for kinetic flux and "two-fluid" hydrodynamic solver for collisional/frictional fluxes
- Joint mass-velocity NDF allows for polydisperse particles is used for applications involving particle "size" changes



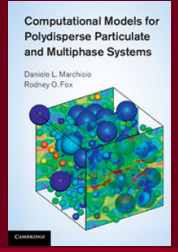
Summary for Gas–Particle Flows

- Quadrature-based moment methods solve kinetic equation by reconstructing velocity distribution function from moments
- For dilute flows, simulations use velocity NDF reconstruction and kinetic-based finite-volume methods
- For dense flows, use KBFVM for kinetic flux and "two-fluid" hydrodynamic solver for collisional/frictional fluxes
- Joint mass-velocity NDF allows for polydisperse particles is used for applications involving particle "size" changes



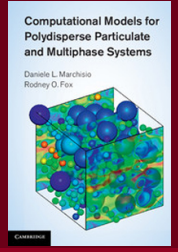
Summary for Gas–Particle Flows

- Quadrature-based moment methods solve kinetic equation by reconstructing velocity distribution function from moments
- For dilute flows, simulations use velocity NDF reconstruction and kinetic-based finite-volume methods
- For dense flows, use KBFVM for kinetic flux and "two-fluid" hydrodynamic solver for collisional/frictional fluxes
- Joint mass-velocity NDF allows for polydisperse particles is used for applications involving particle "size" changes

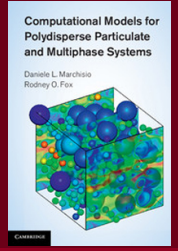


Summary for Gas–Particle Flows

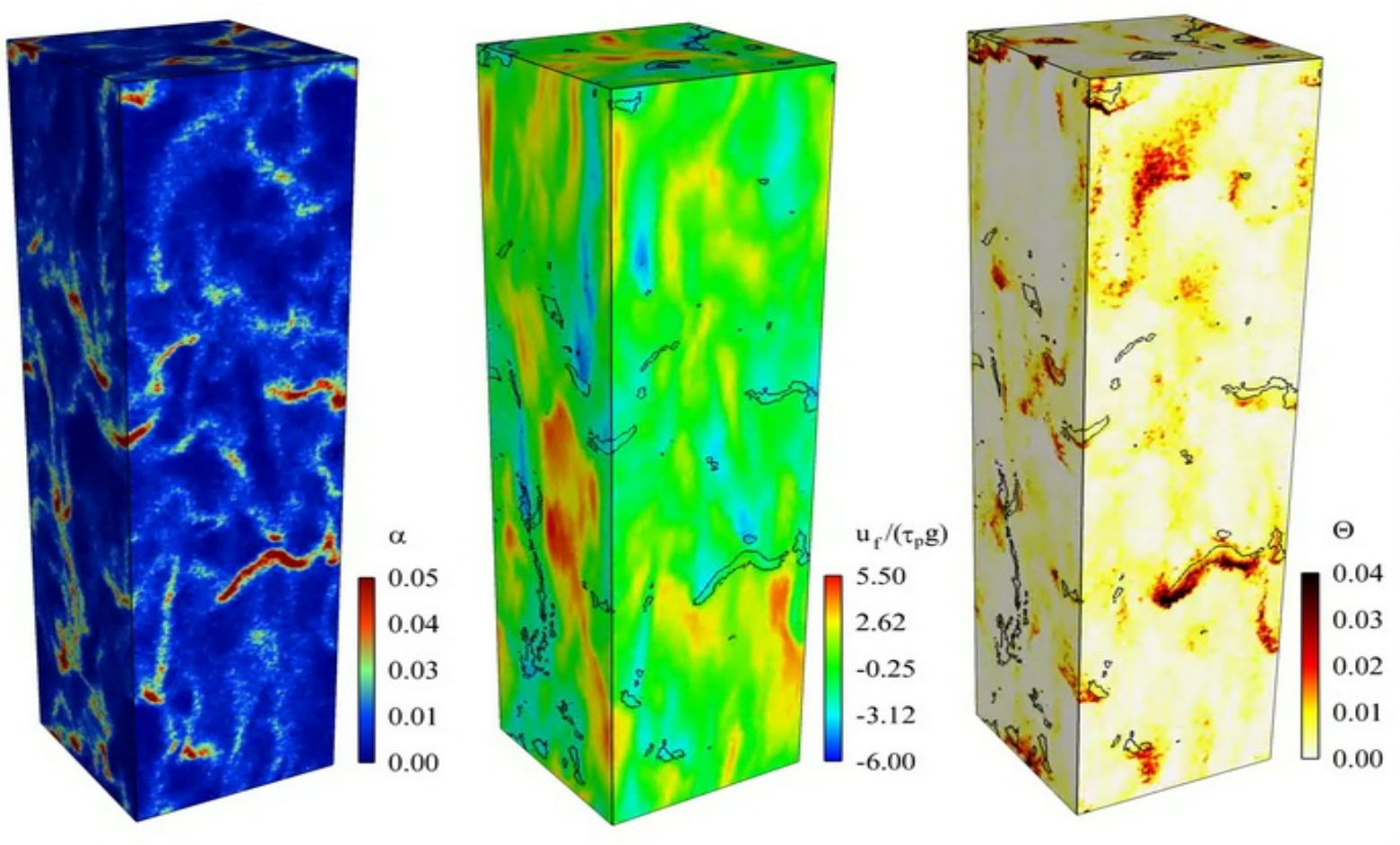
- Quadrature-based moment methods solve kinetic equation by reconstructing velocity distribution function from moments
- For dilute flows, simulations use velocity NDF reconstruction and kinetic-based finite-volume methods
- For dense flows, use KBFVM for kinetic flux and "two-fluid" hydrodynamic solver for collisional/frictional fluxes
- Joint mass-velocity NDF allows for polydisperse particles is used for applications involving particle "size" changes

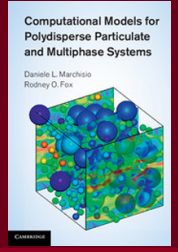


TURBULENCE MODELING FOR DISPERSE MULTIPHASE FLOWS



Example: Cluster-Induced Turbulence



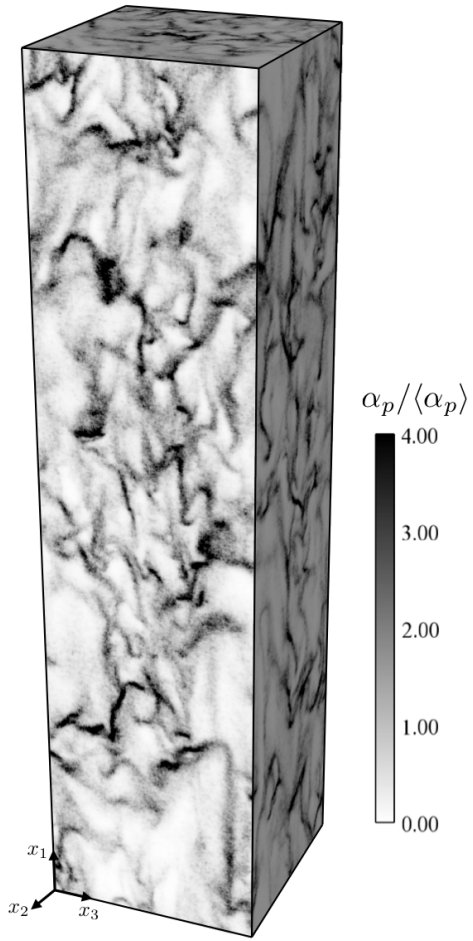


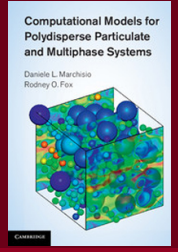
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \frac{\rho_p}{\rho_f}\frac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \frac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \frac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \frac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

CIT exists due to momentum coupling



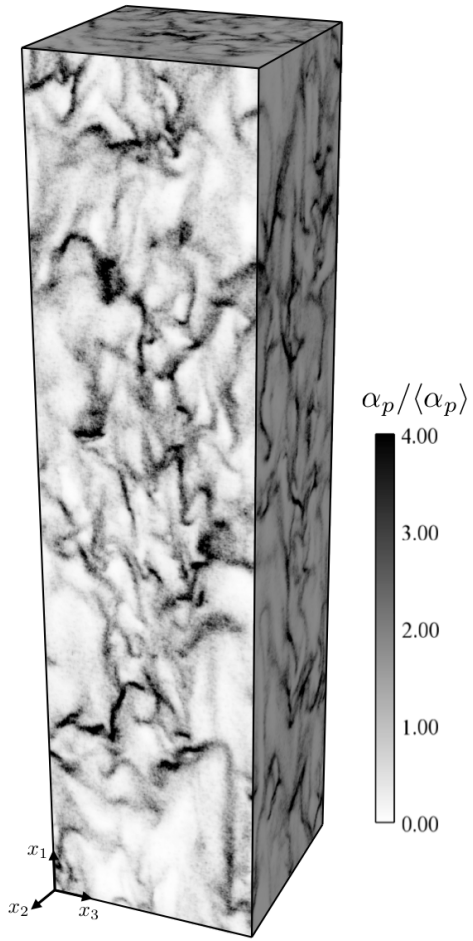


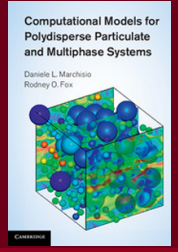
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \dfrac{\rho_p}{\rho_f}\dfrac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \dfrac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \dfrac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \dfrac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

CIT exists due to momentum coupling



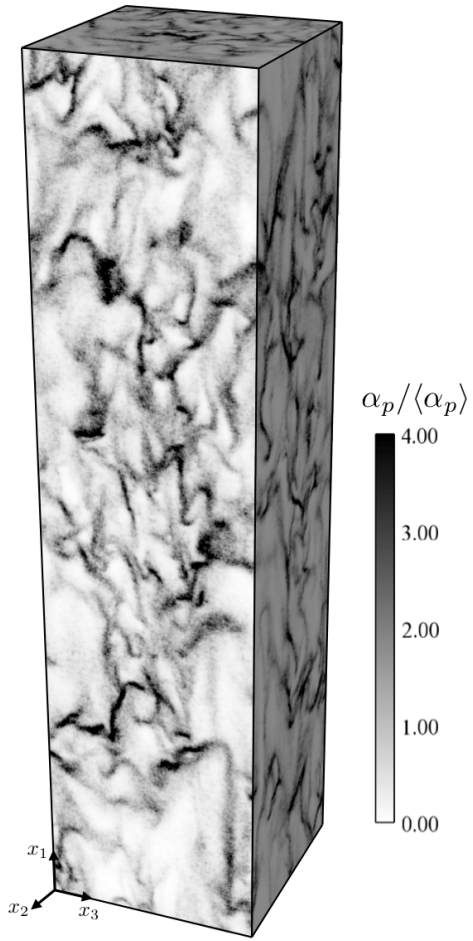


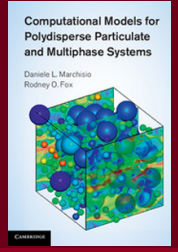
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \dfrac{\rho_p}{\rho_f}\dfrac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \dfrac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \dfrac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \dfrac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

CIT exists due to momentum coupling



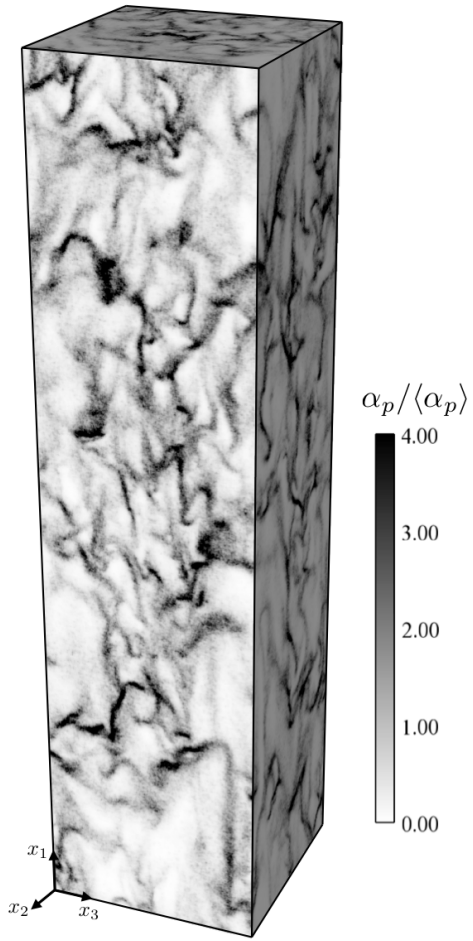


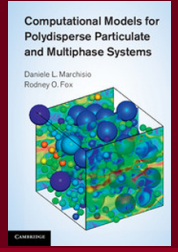
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \dfrac{\rho_p}{\rho_f}\dfrac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \dfrac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \dfrac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \dfrac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

CIT exists due to momentum coupling



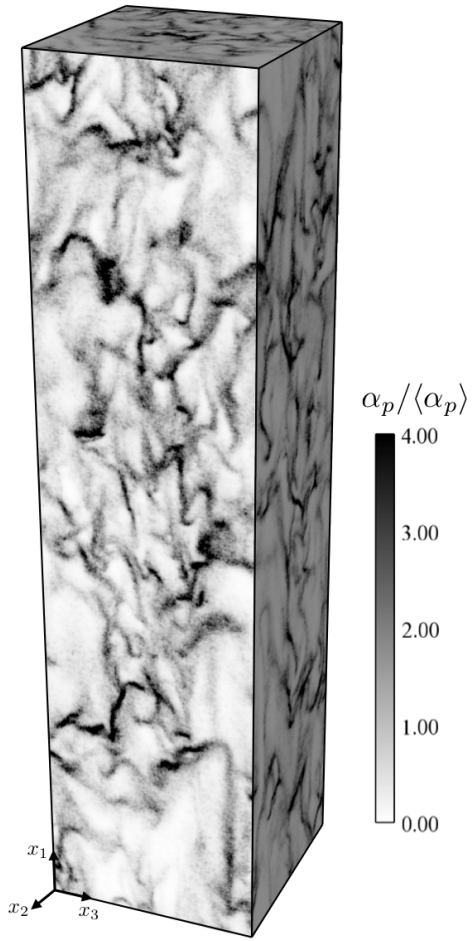


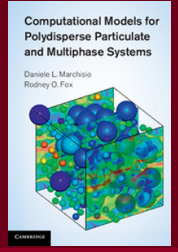
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \dfrac{\rho_p}{\rho_f}\dfrac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \dfrac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \dfrac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \dfrac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

CIT exists due to momentum coupling



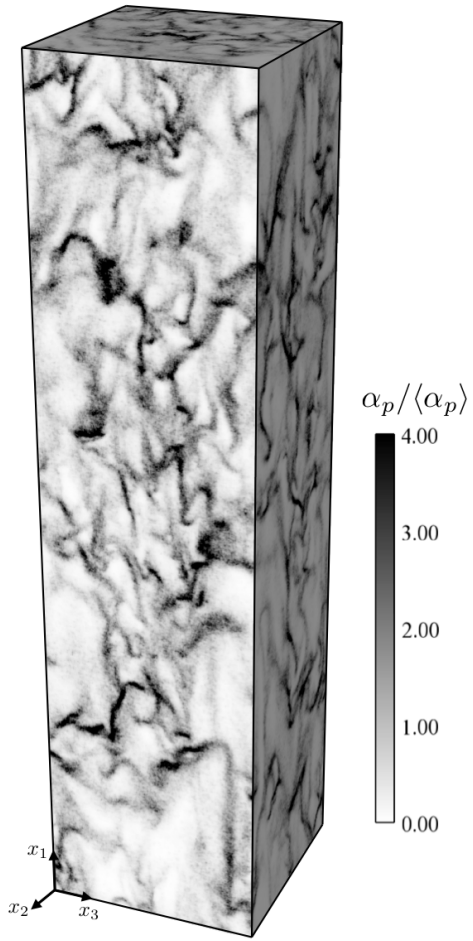


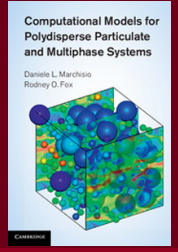
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \frac{\rho_p}{\rho_f}\frac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \frac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \frac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \frac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

CIT exists due to momentum coupling



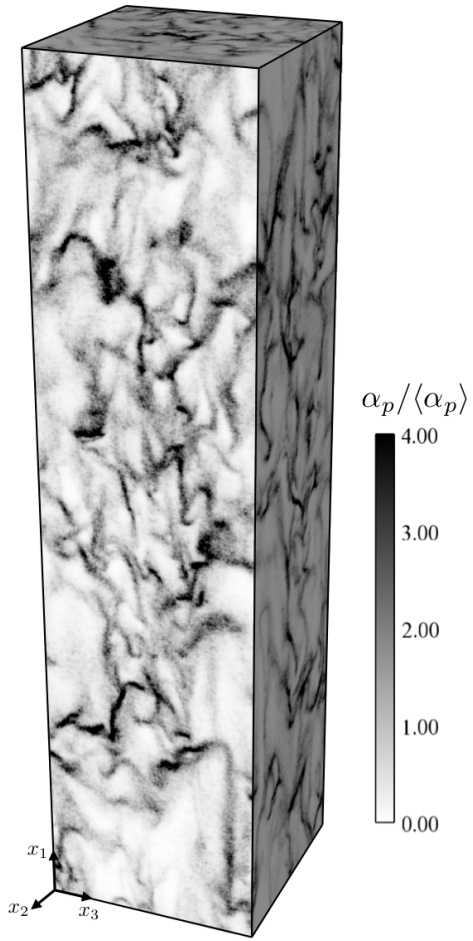


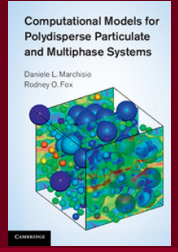
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \dfrac{\rho_p}{\rho_f}\dfrac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \dfrac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \dfrac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \dfrac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

CIT exists due to momentum coupling



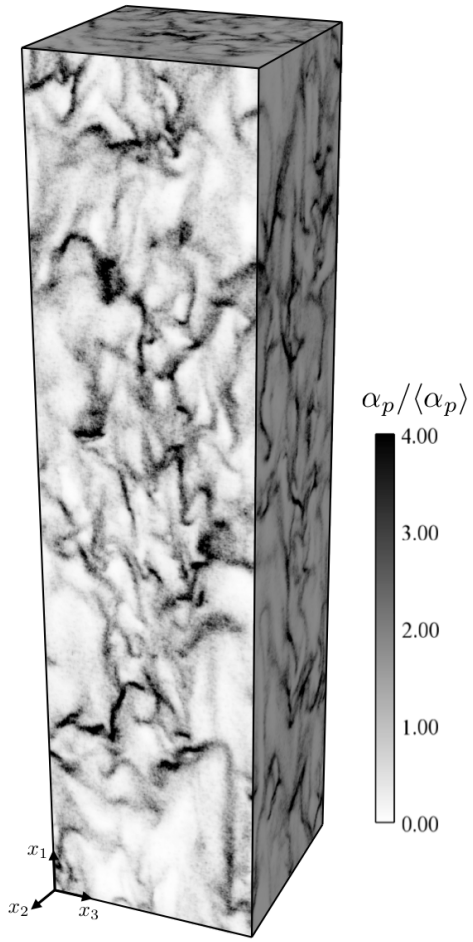


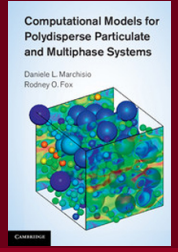
Example: Cluster-Induced Turbulence

Start with homogeneous, laminar flow:

- particles fall under gravity g with $\rho_p \gg \rho_f$
- Stokes drag time $\tau_p = \dfrac{\rho_p}{\rho_f}\dfrac{d_p^2}{18\nu_f}$
  terminal velocity $\propto \mathcal{V} = \tau_p g$
  particle Reynolds number $Re_p = \dfrac{d_p \mathcal{V}}{\nu_f}$
- spontaneous cluster formation
  cluster size $\propto \mathcal{L} = \tau_p \mathcal{V} \gg d_p$
  fluid-phase Reynolds number $Re_f = \dfrac{\mathcal{L}\mathcal{V}}{\nu_f}$
- particle collisions $\langle \alpha_p \rangle \approx 0.01$
- mass loading $\varphi = \dfrac{\rho_p \langle \alpha_p \rangle}{\rho_f \langle \alpha_f \rangle} \geq 1$
- statistically homogeneous, stationary turbulence when $Re_f \gg 1$

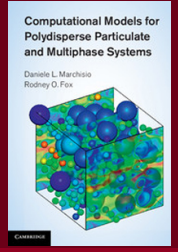
CIT exists due to momentum coupling





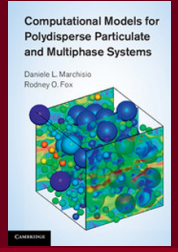
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f} Re_p^2 = \frac{1}{2} St_p Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



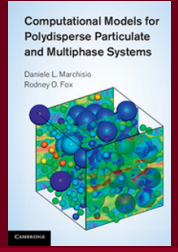
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f} Re_p^2 = \frac{1}{2} St_p Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



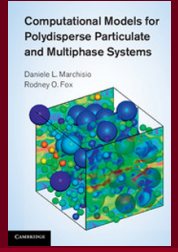
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f} Re_p^2 = \frac{1}{2} St_p Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



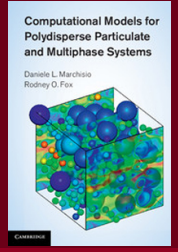
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f} Re_p^2 = \frac{1}{2} St_p Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



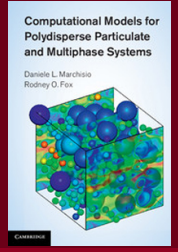
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f} Re_p^2 = \frac{1}{2} St_p \, Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



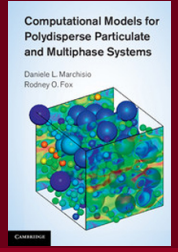
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f}Re_p^2 = \frac{1}{2}St_p\,Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



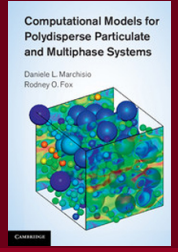
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f}Re_p^2 = \frac{1}{2}St_p\,Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



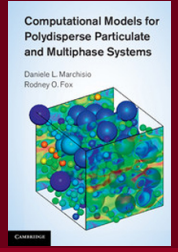
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f}Re_p^2 = \frac{1}{2}St_p\,Re_p$
   - mass loading φ
   - mean volume fraction $\langle\alpha_p\rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



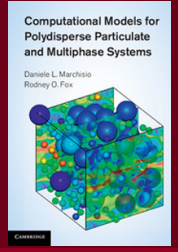
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f} Re_p^2 = \frac{1}{2} St_p \, Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



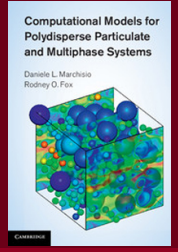
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f}Re_p^2 = \frac{1}{2}St_p\,Re_p$
   - mass loading φ
   - mean volume fraction $\langle\alpha_p\rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



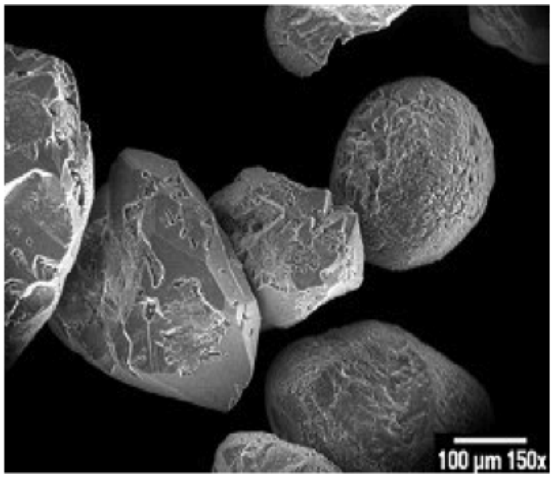
CIT: Important Questions

1. What are the turbulence characteristics of fully developed CIT?
   - integral scales
   - dissipation scales
   - energy spectra for α_p, $\mathbf{u}_p$, $\mathbf{u}_f$, ...
2. How do these characteristics depend on flow parameters?
   - fluid-phase Reynolds number $Re_f = \frac{1}{18}\frac{\rho_p}{\rho_f} Re_p^2 = \frac{1}{2} St_p \, Re_p$
   - mass loading φ
   - mean volume fraction $\langle \alpha_p \rangle$
3. How is turbulent kinetic energy (TKE) produced and transferred between phases?
4. What are roles of correlated vs. uncorrelated components of TKE?
5. How do we derive a one-point turbulence model for CIT?



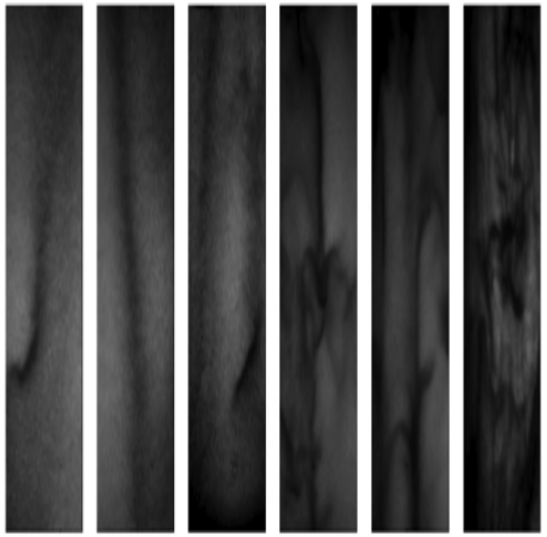
From Microscale to Mesoscale to Macroscale

microscale DNS



- Resolve flow around particles
- Particle collisions resolved
- Fluid stresses on particles

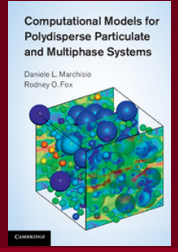
mesoscale simulation



- Drag model
- Particle collisions resolved
- Clusters are resolved

macroscale simulation

- Turbulence model
- Phase-average variables
- Effect of clusters is modeled



Mesoscale Computations
(Capecelatro & Desjardins 2013)

Fluid phase

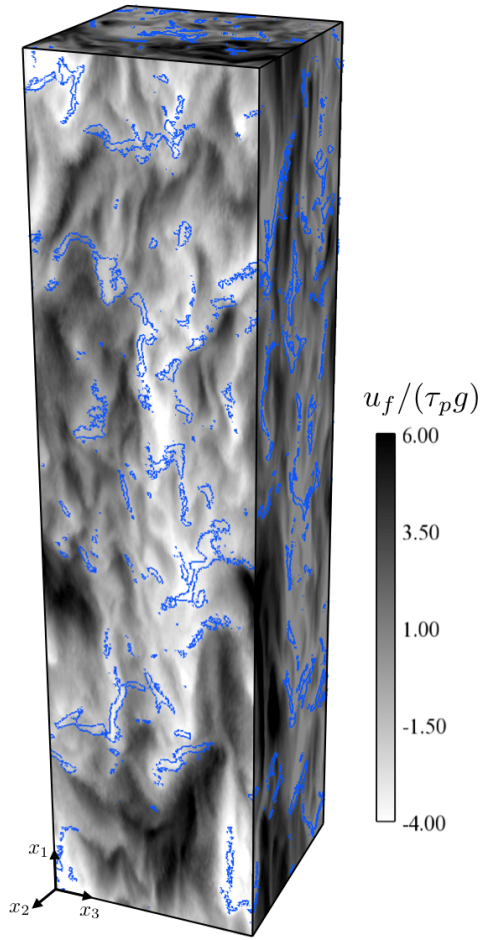
- NGA arbitrarily high-order DNS/LES code
- Massively parallel
- Conservation of mass, momentum, and kinetic energy

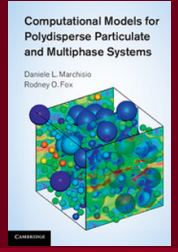
Particle phase

- Lagrangian particle tracking
- Newton's 2nd law with Runge-Kutta ODE solver
- Soft-sphere collision model

Interphase exchange terms

- Transfer particle volume and momentum to fluid
- Fully conservative and consistent filtering approach
- Transferred data converges under mesh refinement





Mesoscale Transport Equations

(Capecelatro & Desjardins 2013)

Gas phase

$$\frac{\partial \rho_f (1-\hat{\alpha})}{\partial t} + \nabla \cdot \rho_f (1-\hat{\alpha}) \mathbf{u}_f = 0$$

$$\frac{\partial \rho_f (1-\hat{\alpha}) \mathbf{u}_f}{\partial t} + \nabla \cdot \rho_f (1-\hat{\alpha}) \mathbf{u}_f \mathbf{u}_f = -\nabla p_f + \nabla \cdot \boldsymbol{\sigma}_f - \rho_p \hat{\alpha} \hat{\mathcal{A}} + \rho_f (1-\hat{\alpha}) \mathbf{g}$$

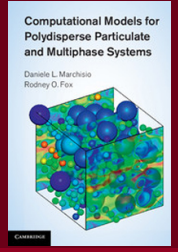
Particles $(1 \le p \le N_p)$

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{v}_p \qquad \frac{\mathrm{d}\mathbf{v}_p}{\mathrm{d}t} = \mathcal{A}_p + \mathbf{F}_p^{\mathrm{col}} + \mathbf{g}$$

Two-way coupling

$$\hat{\alpha}(t,\mathbf{x}) \approx \sum_{p=1}^{N_p} G(|\mathbf{x} - \mathbf{x}_p(t)|) V_p \qquad \hat{\alpha}\hat{\mathcal{A}}(t,\mathbf{x}) \approx \sum_{p=1}^{N_p} \mathcal{A}_p G(|\mathbf{x} - \mathbf{x}_p(t)|) V_p$$

$G(x)$ is an isotropic Gaussian filter with width $\delta_f = 8 d_p$



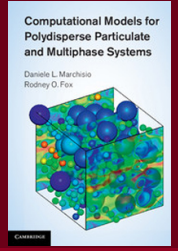
Simulation Parameters for CIT

Physical parameters

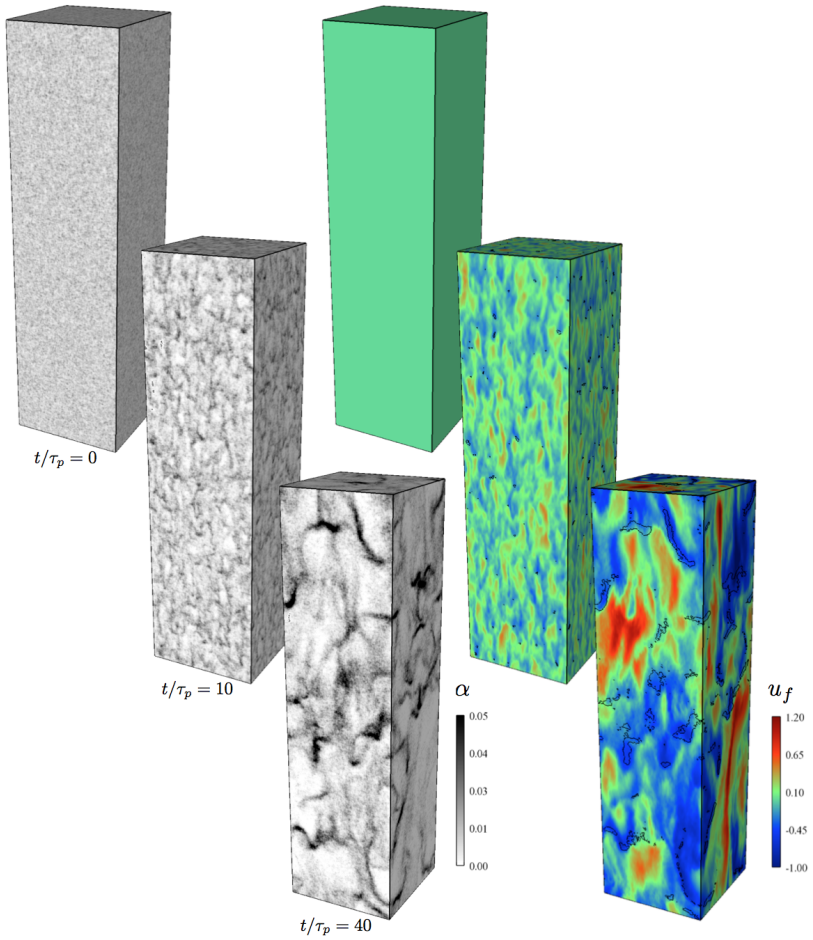
d_p	Particle diameter	0.09			mm
ρ_p	Particle density	1000			kg/m^3
ρ_f	Fluid density	1			kg/m^3
ν_f	Fluid kinematic viscosity	1.8×10^{-5}			m^2/s
e	Coefficient of restitution	0.9			
N_p	Number of particles	55×10^6			
g	Gravity magnitude	2.0002	4.0004	8.0008	m/s^2
$\mathcal{V}$	Cluster velocity	0.05	0.1	0.2	m/s
$\mathcal{L}$	Cluster length	1.25	2.5	5	mm

Non-dimensional parameters

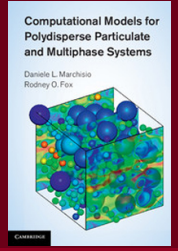
$\langle\alpha_p\rangle$	Mean particle volume fraction	0.01		
φ	Mean mass loading	10.1		
Re_p	Particle Reynolds number	0.25	0.5	1
Re_f	Fluid Reynolds number	3.5	14	56
$L_x/\mathcal{L}$	Domain length	256	128	64



Time Evolution of Cluster Formation

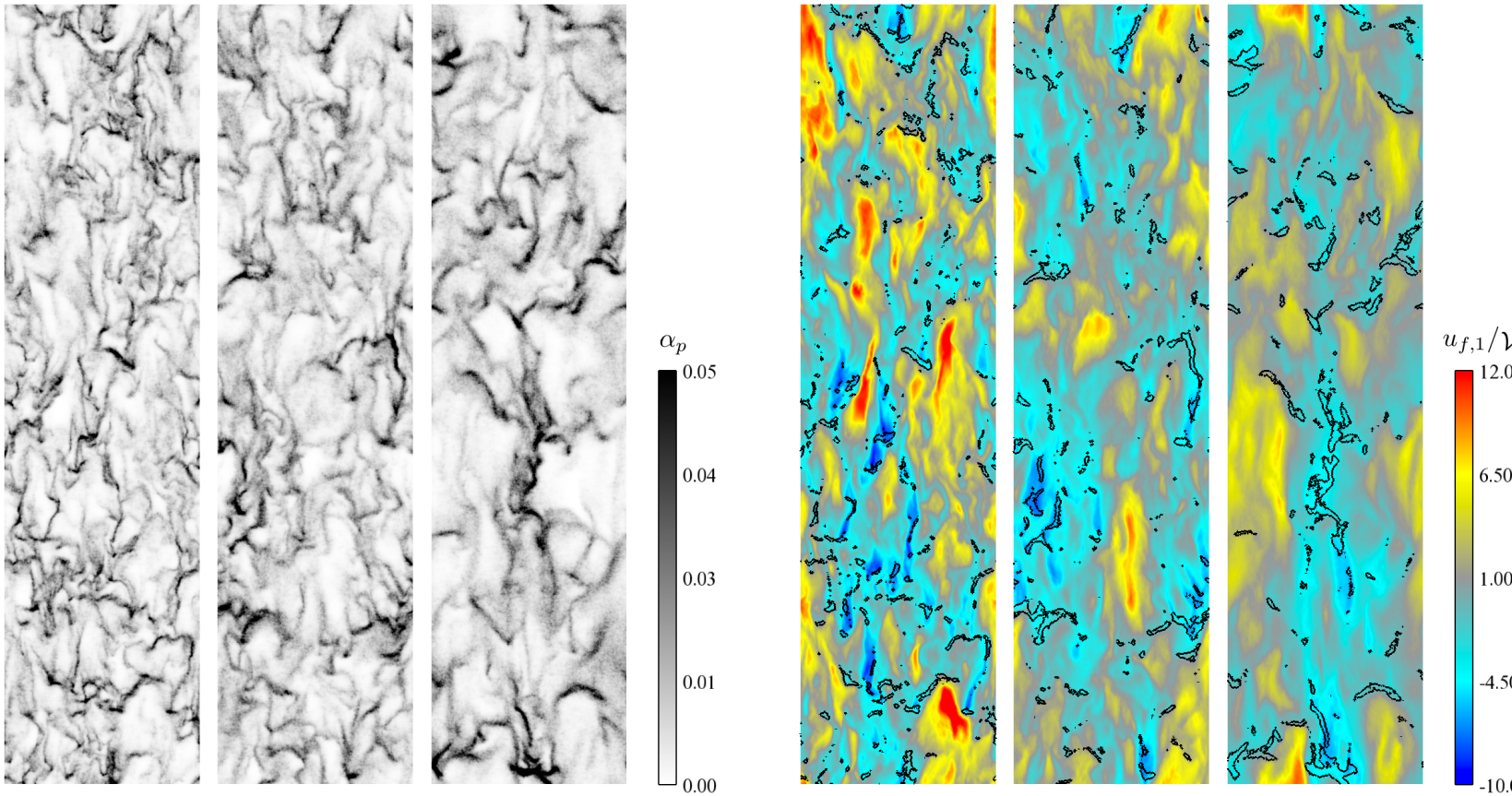


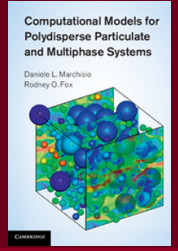
- Clusters form due to fluid instability
- Positive feedback due to cluster-fluid-velocity correlation
- Steady state reached when fluid turbulence breaks up clusters
- Fluid velocity seen by particles is negatively correlated with α_p
- Box size must allow fluid turbulence to fully develop at large scales!



Particle Volume Fraction and Fluid Velocity vs. Re_p

Re_p 0.25 0.5 1 Re_p 0.25 0.5 1





Spatially Correlated and Uncorrelated Velocity

$$\mathbf{v}_p^{(i)}(t) = \mathbf{u}_p(\mathbf{x}_p^{(i)}, t) + \delta\mathbf{v}_p^{(i)}$$

correlated

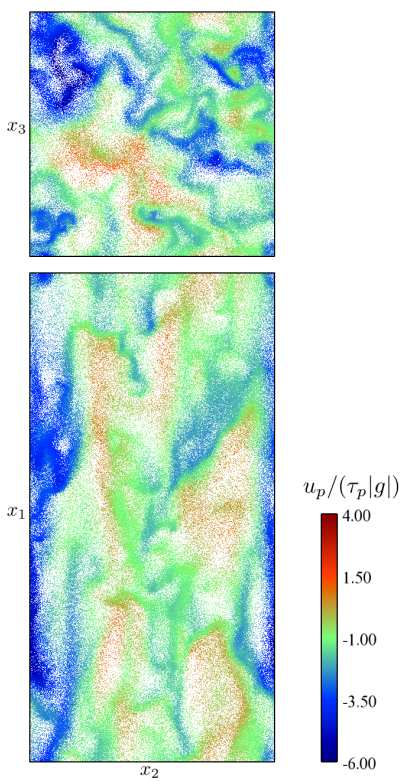
$$\mathbf{u}_p(\mathbf{x}, t)$$

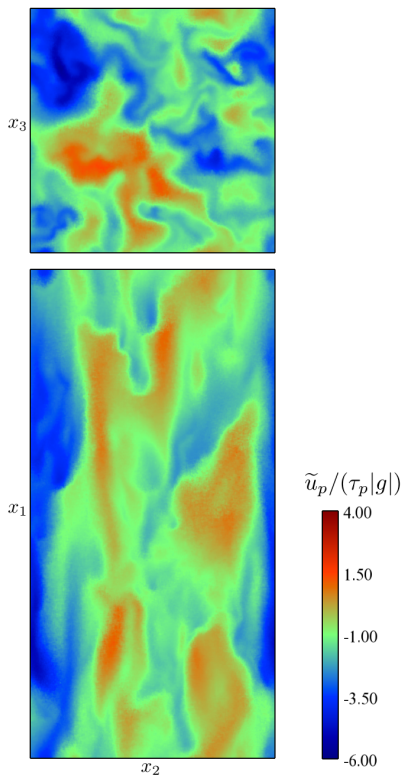
correlated TKE

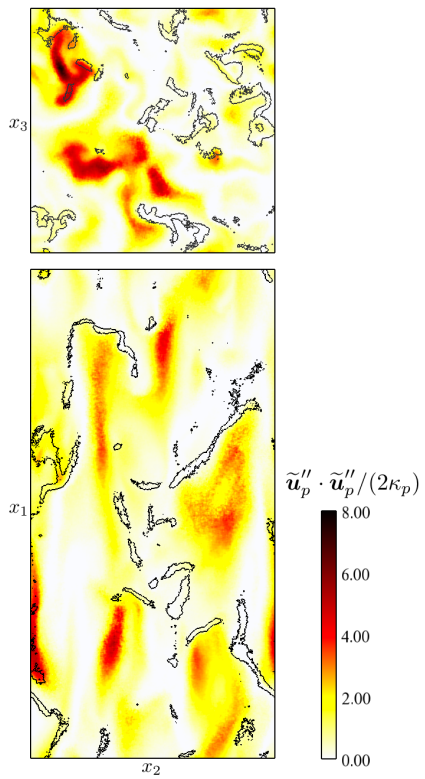
$$k_p^* = \tfrac{1}{2}\mathbf{u}_p \cdot \mathbf{u}_p$$

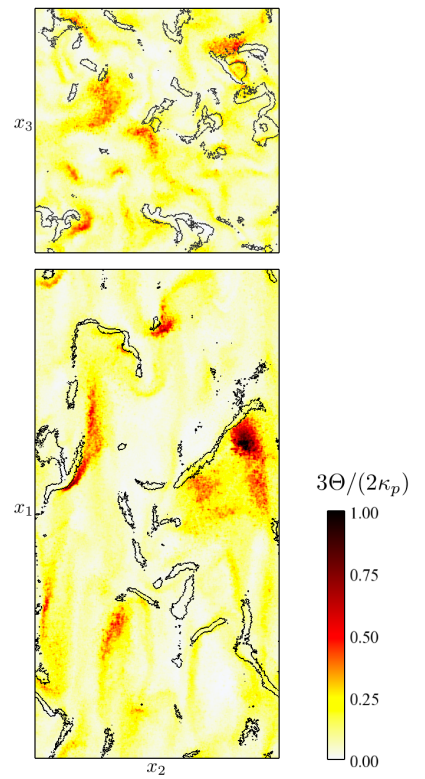
uncorrelated TKE

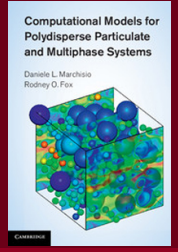
$$\Theta_p^* = \tfrac{1}{3}\delta\mathbf{v}_p \cdot \delta\mathbf{v}_p$$





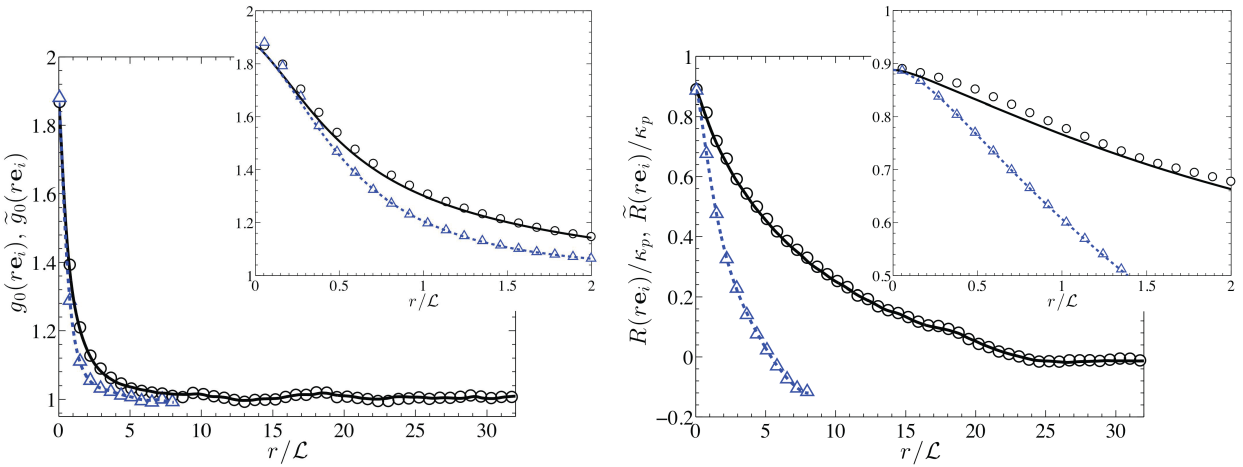


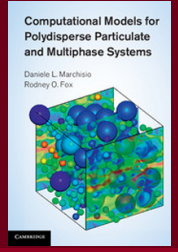




Filter for Decomposing k_p^* and Θ_p^*

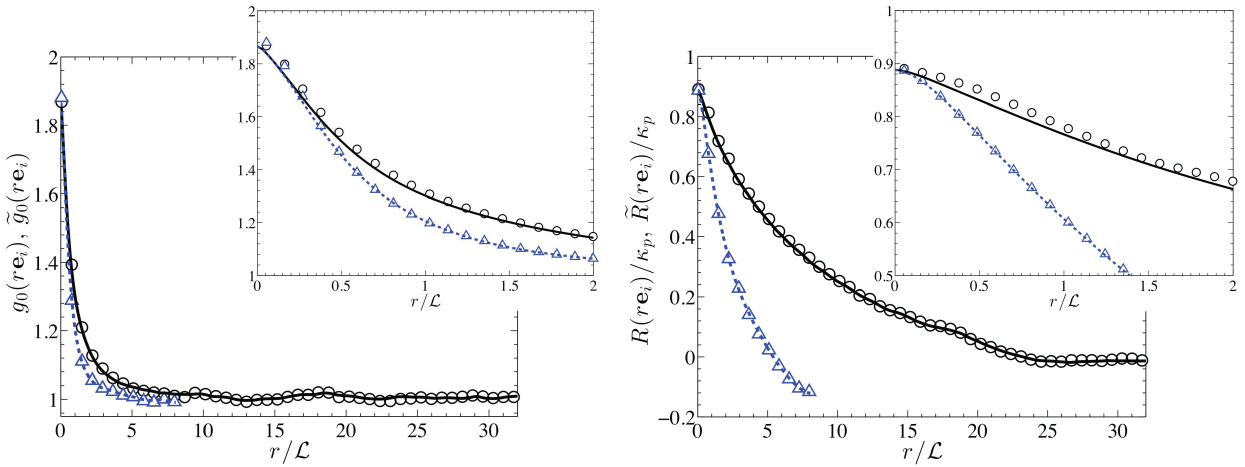
- $\kappa_p^* = \frac{1}{2}\mathbf{v}_p \cdot \mathbf{v}_p = k_p^* + \frac{3}{2}\Theta_p^*$ is particle property (independent of filter)
- Partition between k_p^* and Θ_p^* depends on filter!
- Use variable filter width $\delta_f(\alpha_p) = \left(\frac{\mathcal{N}_p d_p^3}{\alpha_p}\right)^{1/3}$ with $\mathcal{N}_p = 10$
- Validate by comparing Lagrangian and Eulerian two-point statistics:

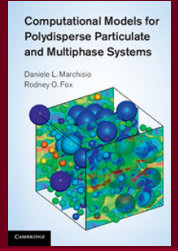




Filter for Decomposing k_p^* and Θ_p^*

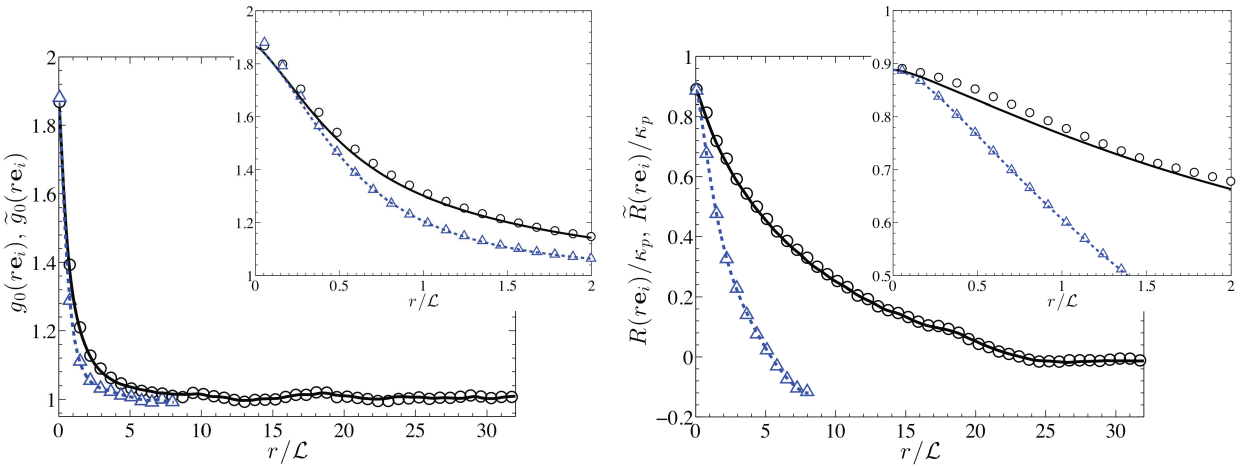
- $\kappa_p^* = \frac{1}{2}\mathbf{v}_p \cdot \mathbf{v}_p = k_p^* + \frac{3}{2}\Theta_p^*$ is particle property (independent of filter)
- Partition between k_p^* and Θ_p^* depends on filter!
- Use variable filter width $\delta_f(\alpha_p) = \left(\frac{\mathcal{N}_p d_p^3}{\alpha_p}\right)^{1/3}$ with $\mathcal{N}_p = 10$
- Validate by comparing Lagrangian and Eulerian two-point statistics:

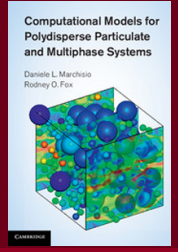




Filter for Decomposing k_p^* and Θ_p^*

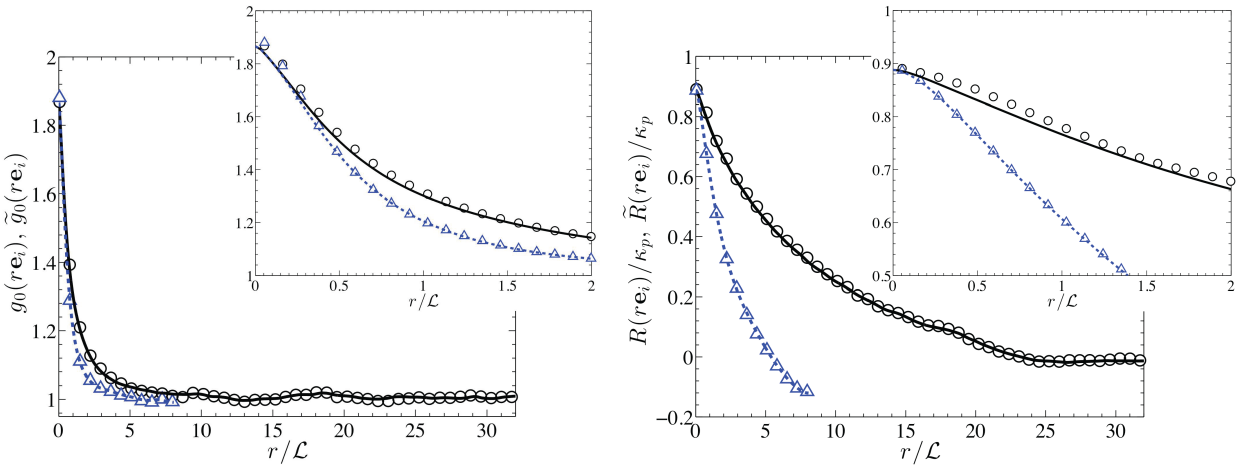
- $\kappa_p^* = \frac{1}{2}\mathbf{v}_p \cdot \mathbf{v}_p = k_p^* + \frac{3}{2}\Theta_p^*$ is particle property (independent of filter)
- Partition between k_p^* and Θ_p^* depends on filter!
- Use variable filter width $\delta_f(\alpha_p) = \left(\frac{\mathcal{N}_p d_p^3}{\alpha_p}\right)^{1/3}$ with $\mathcal{N}_p = 10$
- Validate by comparing Lagrangian and Eulerian two-point statistics:

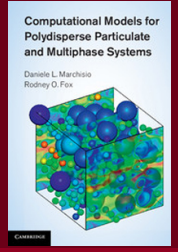




Filter for Decomposing k_p^* and Θ_p^*

- $\kappa_p^* = \frac{1}{2}\mathbf{v}_p \cdot \mathbf{v}_p = k_p^* + \frac{3}{2}\Theta_p^*$ is particle property (independent of filter)
- Partition between k_p^* and Θ_p^* depends on filter!
- Use variable filter width $\delta_f(\alpha_p) = \left(\frac{\mathcal{N}_p d_p^3}{\alpha_p}\right)^{1/3}$ with $\mathcal{N}_p = 10$
- Validate by comparing Lagrangian and Eulerian two-point statistics:

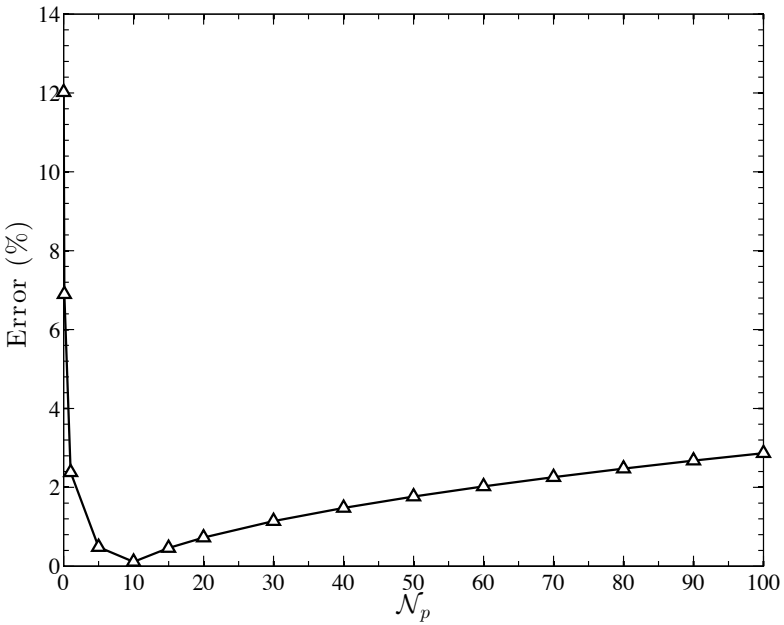




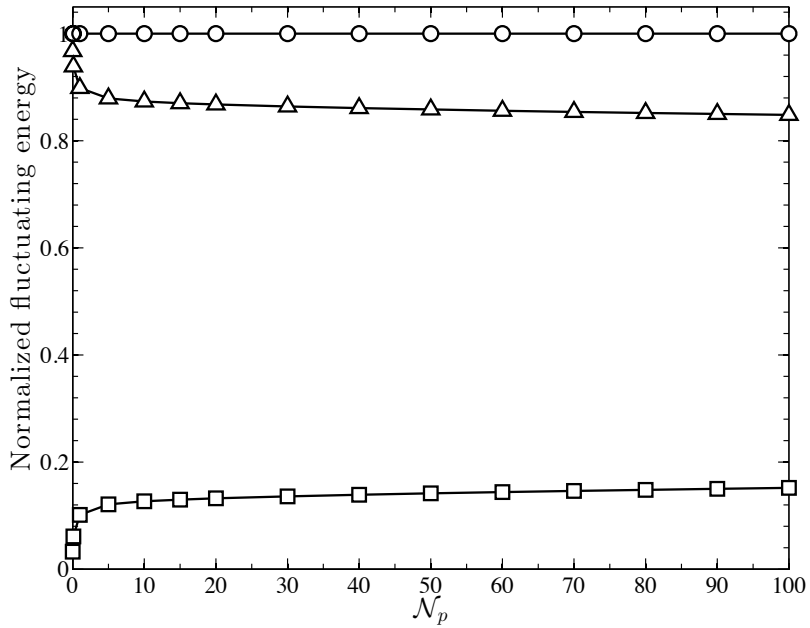
Filter Error for Energy Components

Variable filter width yields good decomposition of κ_p^* to find k_p^* and Θ_p^*

k_p^* error vs. $\mathcal{N}_p$

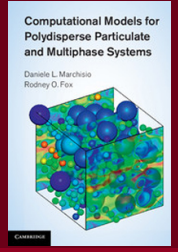


Components of κ_p^* vs. $\mathcal{N}_p$



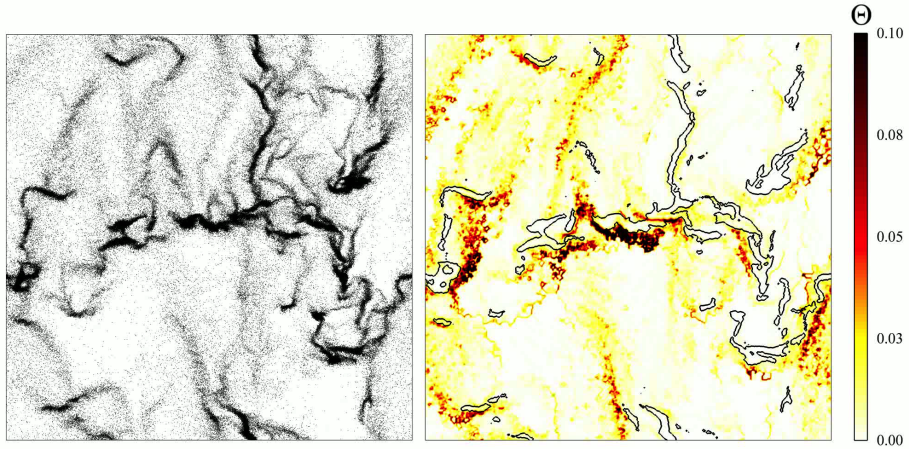
$\mathcal{N}_p \approx 10$ agrees with Lagrangian 2-pt statistics

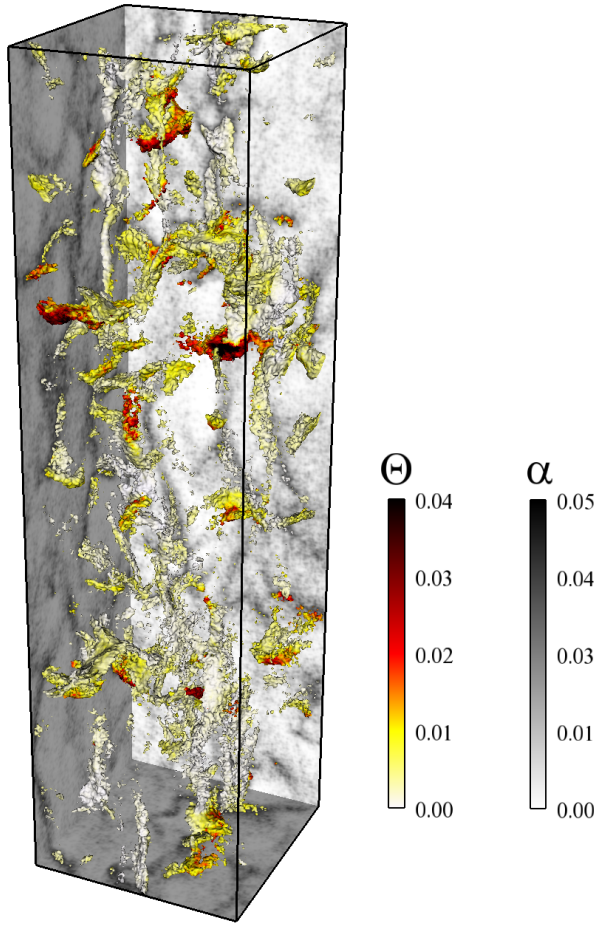
We use adaptive filter to study instantaneous local fields of $\mathbf{u}_p$ and $\delta\mathbf{v}_p$

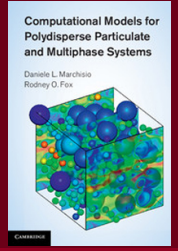


Dynamics of Granular Temperature Field

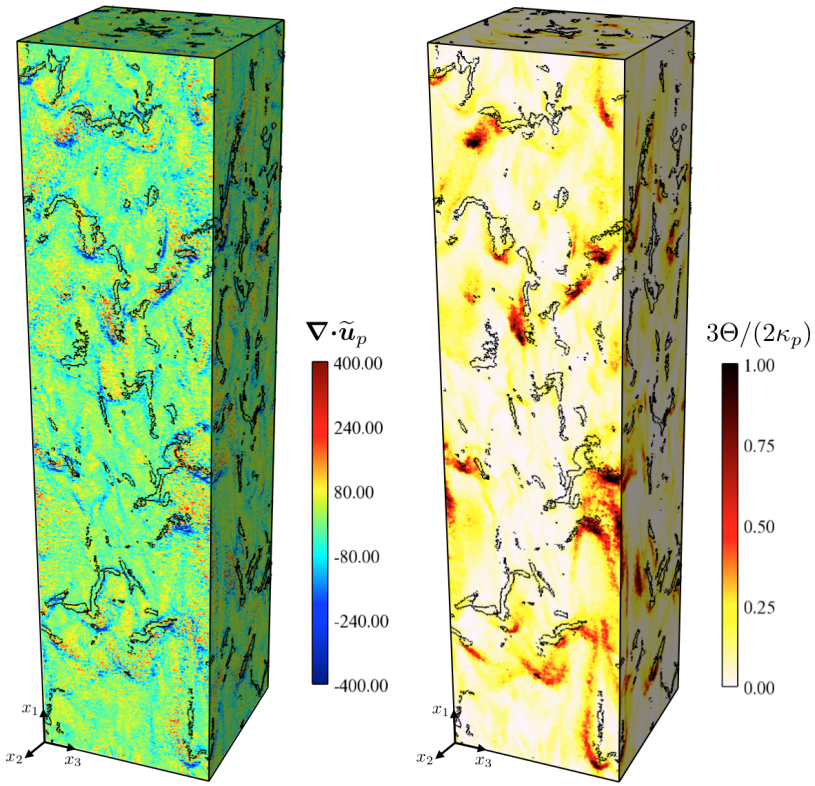
Maximum Θ^* in front of clusters



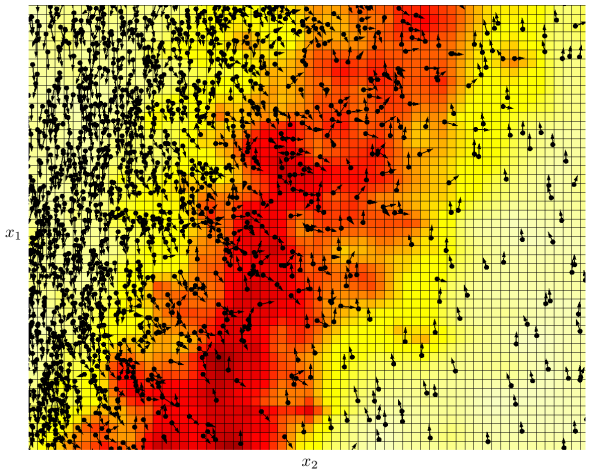


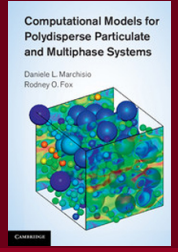


Compressive "Heating" of Particle Phase



- Maximum compression ($-\nabla\cdot\mathbf{u}_p$) at same location at maximum Θ_p^*
- Cluster fall velocity reduced by granular pressure $p = \alpha_p\Theta_p^*$





Eulerian Mesoscale Model for Particle Phase

Kinetic equation for particle number density function (NDF): $f(t, \mathbf{x}, \mathbf{v}_p)$

accumulation	+free transport	=acceleration	+collisions
$\frac{\partial f}{\partial t}$	$+\mathbf{v}_p \cdot \frac{\partial f}{\partial \mathbf{x}}$	$= -\frac{\partial}{\partial \mathbf{v}_p} \cdot [(\mathcal{A}_p + \mathbf{g})f]$	$+C[f,f]$

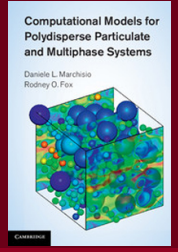
where fluid coupling is modeled as (drag dominant when $\rho_p \gg \rho_f$)

$$\mathcal{A}_p(t, \mathbf{x}, \mathbf{v}_p) = \frac{1}{\tau_p}(\mathbf{u}_f - \mathbf{v}_p) - \frac{1}{\rho_p}\nabla p_f + \frac{1}{\rho_p}\nabla \cdot \boldsymbol{\sigma}_f$$

Define particle-phase mass, momentum and pressure tensor ($trace(\mathbf{P}) = 3\Theta_p$):

$$\alpha_p = \int f \, d\mathbf{v}_p \quad \alpha_p \mathbf{u}_p = \int \mathbf{v}_p f \, d\mathbf{v}_p \quad \alpha_p \mathbf{P} = \int \mathbf{v}_p'' \otimes \mathbf{v}_p'' f \, d\mathbf{v}_p$$

closure with these variables yields anisotropic Gaussian (AG) model



Eulerian Mesoscale Model for Particle Phase

Kinetic equation for particle number density function (NDF): $f(t, \mathbf{x}, \mathbf{v}_p)$

accumulation +free transport =acceleration +collisions

$$\frac{\partial f}{\partial t} + \mathbf{v}_p \cdot \frac{\partial f}{\partial \mathbf{x}} = -\frac{\partial}{\partial \mathbf{v}_p} \cdot [(\mathcal{A}_p + \mathbf{g})f] + C[f,f]$$

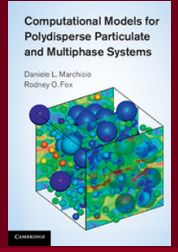
where fluid coupling is modeled as (drag dominant when $\rho_p \gg \rho_f$)

$$\mathcal{A}_p(t, \mathbf{x}, \mathbf{v}_p) = \frac{1}{\tau_p}(\mathbf{u}_f - \mathbf{v}_p) - \frac{1}{\rho_p}\nabla p_f + \frac{1}{\rho_p}\nabla \cdot \boldsymbol{\sigma}_f$$

Define particle-phase mass, momentum and pressure tensor ($trace(\mathbf{P}) = 3\Theta_p$):

$$\alpha_p = \int f \, \mathrm{d}\mathbf{v}_p \quad \alpha_p \mathbf{u}_p = \int \mathbf{v}_p f \, \mathrm{d}\mathbf{v}_p \quad \alpha_p \mathbf{P} = \int \mathbf{v}_p'' \otimes \mathbf{v}_p'' f \, \mathrm{d}\mathbf{v}_p$$

closure with these variables yields anisotropic Gaussian (AG) model



Eulerian Transport Equations

Gas phase

$$\frac{\partial \rho_f \alpha_f}{\partial t} + \nabla \cdot \rho_f \alpha_f \mathbf{u}_f = 0$$

$$\frac{\partial \rho_f \alpha_f \mathbf{u}_f}{\partial t} + \nabla \cdot \rho_f \alpha_f \mathbf{u}_f \otimes \mathbf{u}_f = -\nabla p_f + \nabla \cdot \boldsymbol{\sigma}_f - \rho_p \alpha_p \mathcal{A}_p + \rho_f \alpha_f \mathbf{g}$$

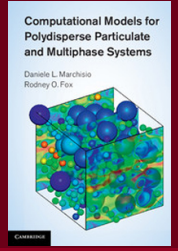
Particle phase

$$\frac{\partial \rho_p \alpha_p}{\partial t} + \nabla \cdot \rho_p \alpha_p \mathbf{u}_p = 0$$

$$\frac{\partial \rho_p \alpha_p \mathbf{u}_p}{\partial t} + \nabla \cdot \rho_p \alpha_p (\mathbf{u}_p \otimes \mathbf{u}_p + \mathbf{P}) = \rho_p \alpha_p (\mathcal{A}_p + \mathbf{g})$$

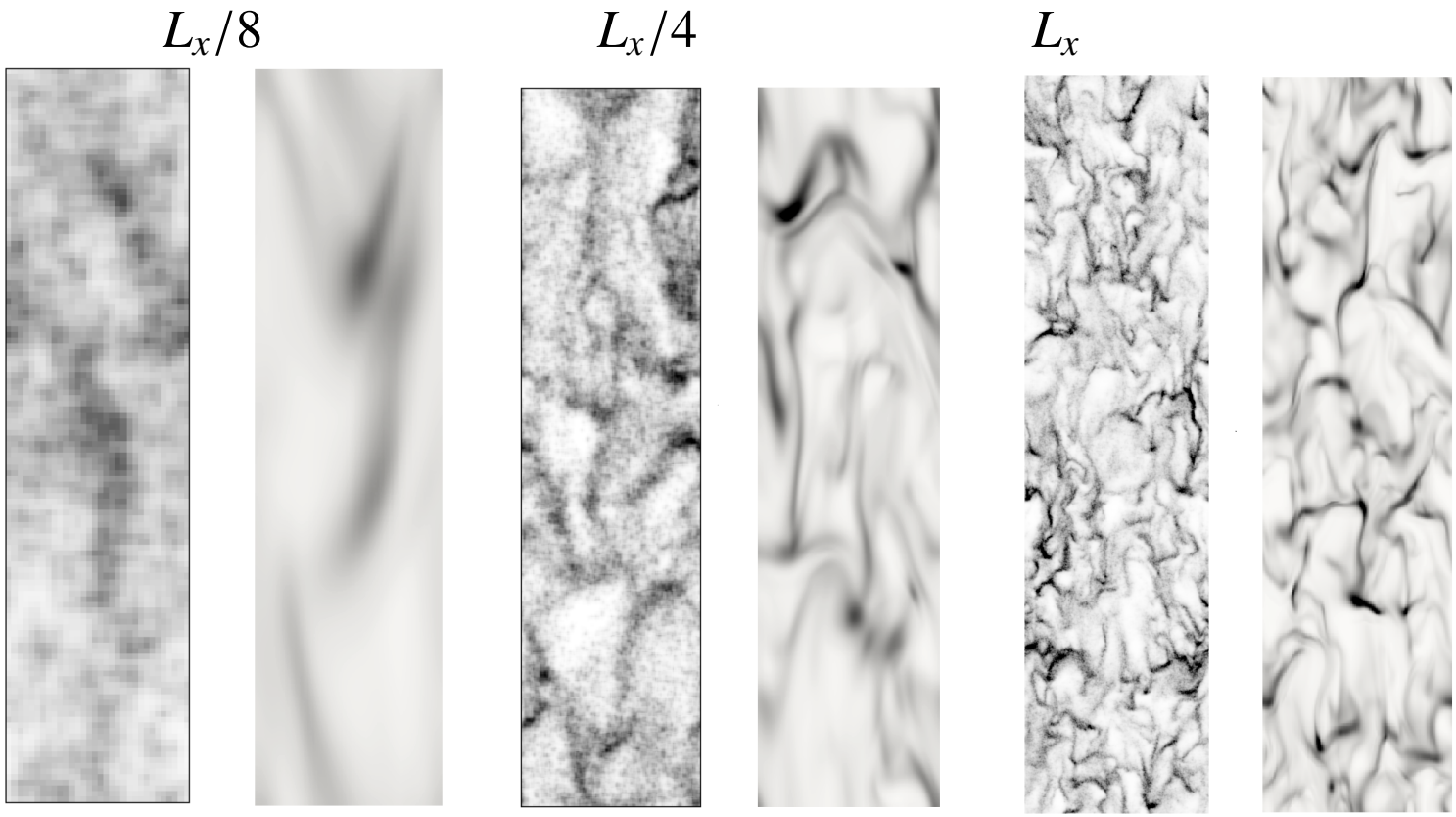
$$\frac{\partial \rho_p \alpha_p \mathbf{P}}{\partial t} + \nabla \cdot \rho_p \alpha_p (\mathbf{u}_p \otimes \mathbf{P} + \mathbf{Q}) = -\rho_p \alpha_p (\mathbf{P} \cdot \nabla \mathbf{u}_p)^\dagger - \rho_p \alpha_p \frac{2}{\tau_p} \mathbf{P} + \rho_p \alpha_p \frac{2}{\tau_c} (\Delta^* - \mathbf{P})$$

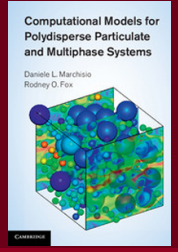
coupling term $\mathcal{A}_p$ is closed, but energy flux $\mathbf{Q}$ requires kinetic theory closure



Particle Volume Fraction

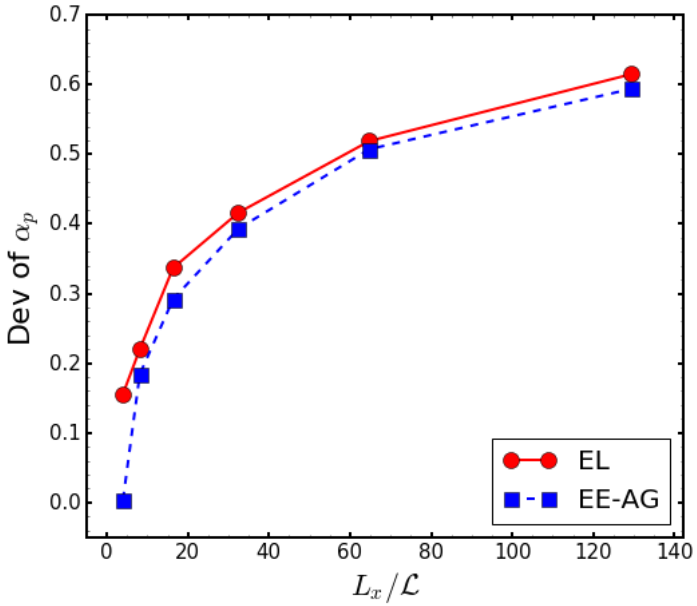
EL vs. EE-AG for $Re_p = 0.5$



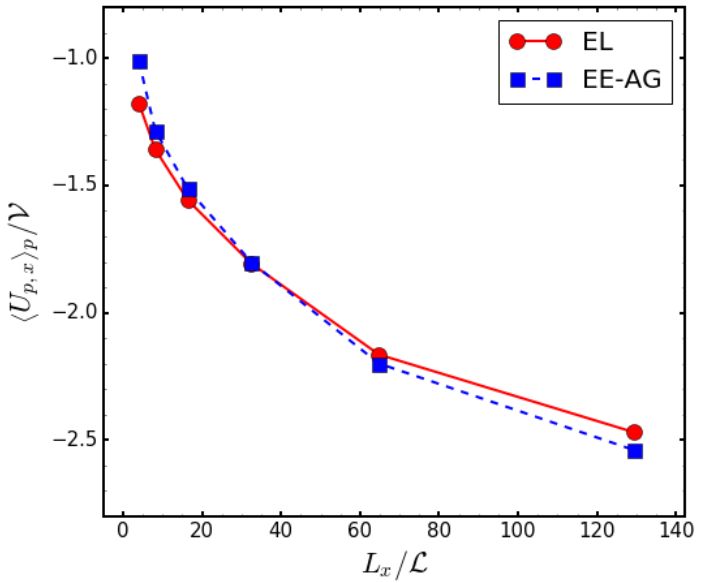


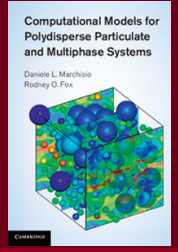
Comparison Between EL and EE-AG

Volume fraction fluctuations



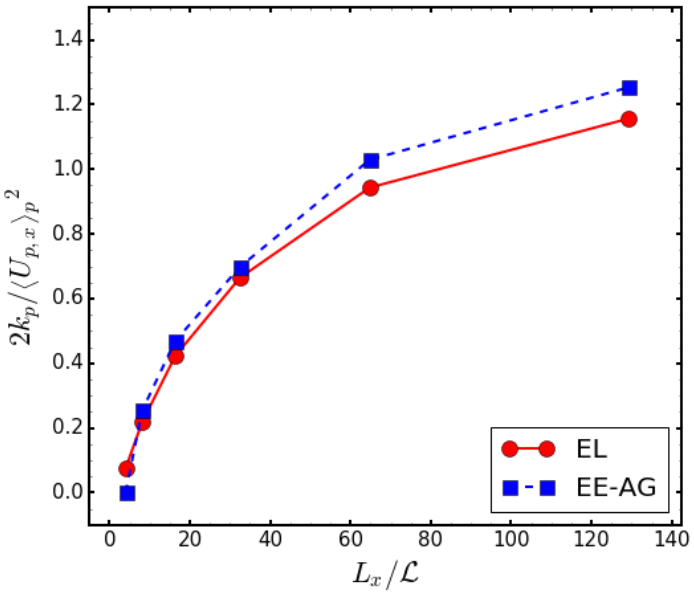
Particle settling velocity



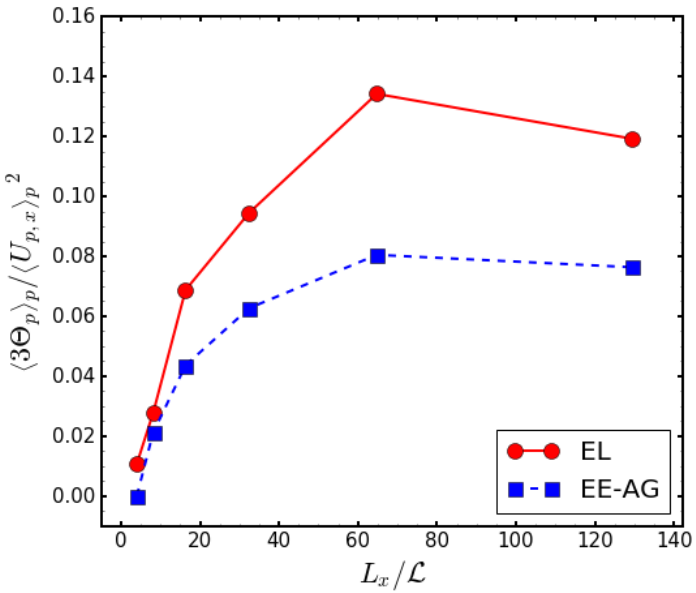


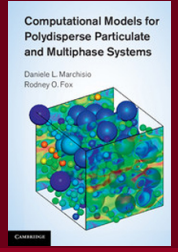
Comparison Between EL and EE-AG

Correlated particle energy

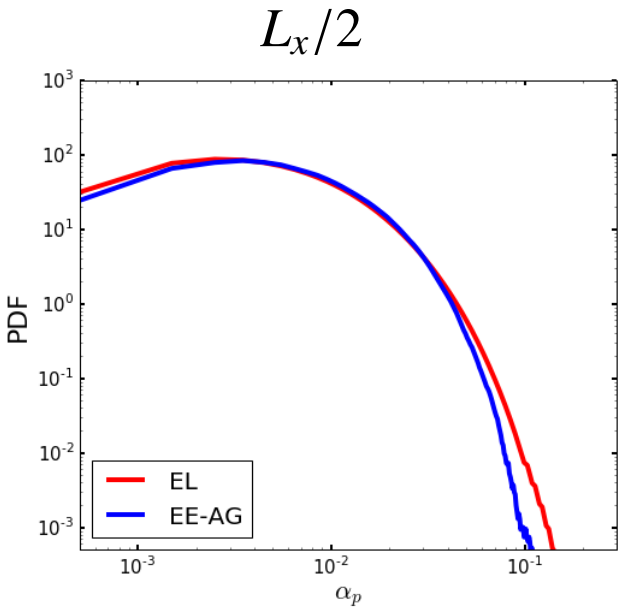


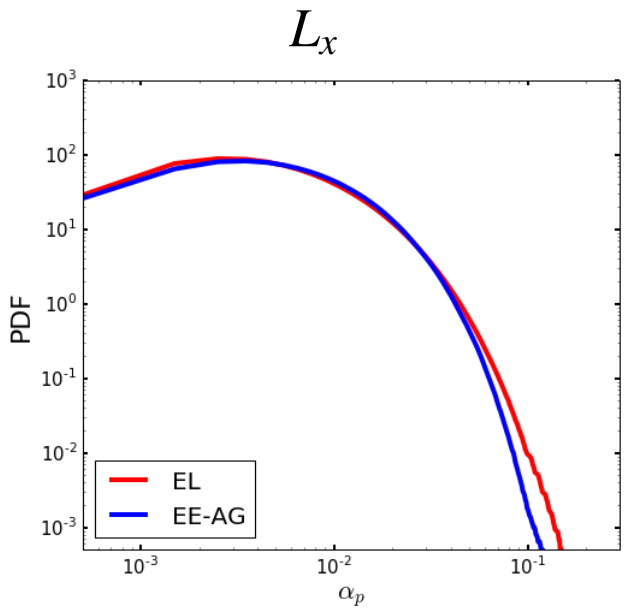
Granular energy contribution



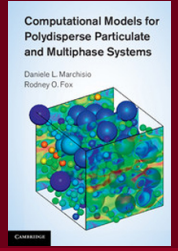


Particle Volume Fraction Distribution





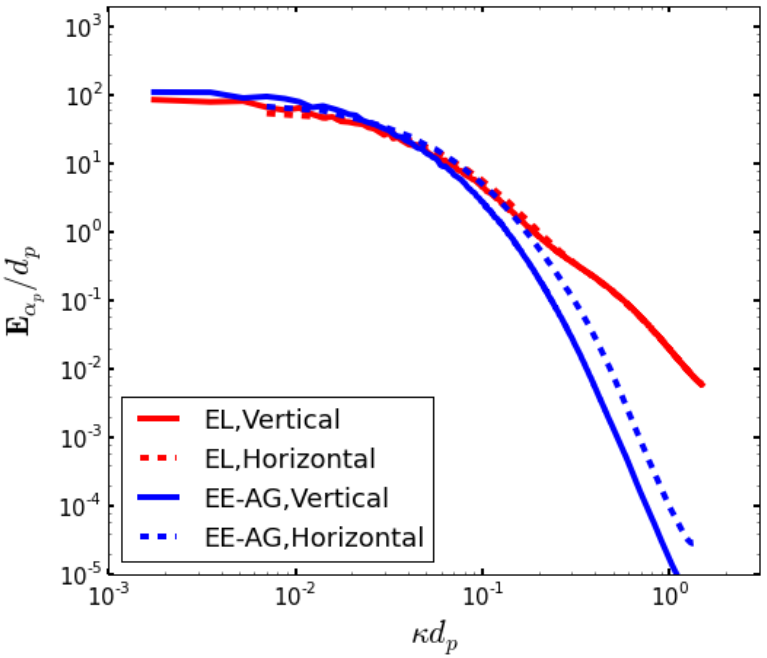
L_x	$\frac{\langle \alpha_p'^2 \rangle}{\langle \alpha_p \rangle^2}$	$\frac{\langle \alpha_p'^3 \rangle}{\langle \alpha_p'^2 \rangle^{3/2}}$	$\frac{\langle \alpha_p'^4 \rangle}{\langle \alpha_p'^2 \rangle^2}$
EL	0.837	2.37	12.19
EE-AG	0.744	2.09	9.91



Energy Spectra

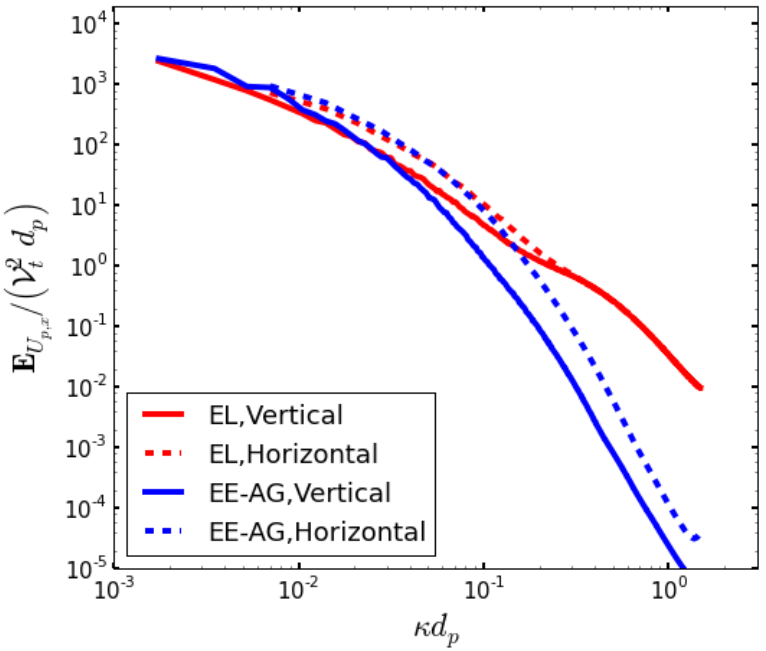
$$g_0(r) = \frac{\langle \alpha_p(x,t)\,\alpha_p(x+r,t)\rangle}{\langle \alpha_p(x,t)\rangle\langle \alpha_p(x+r,t)\rangle}$$

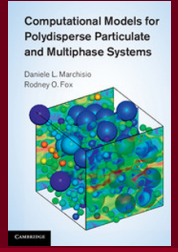
Particle volume fraction



$$R(r) = \frac{\langle (\alpha_p^{1/2}u_p)(x,t)\,(\alpha_p^{1/2}u_p)(x+r,t)\rangle}{\langle \alpha_p^{1/2}(x,t)\,\alpha_p^{1/2}(x+r,t)\rangle}$$

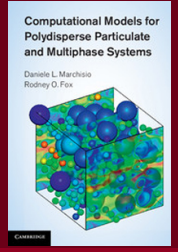
Particle velocity





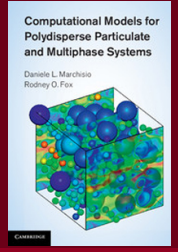
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



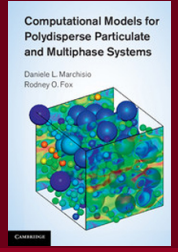
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



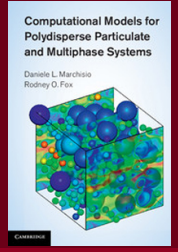
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



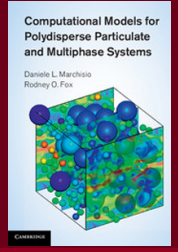
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



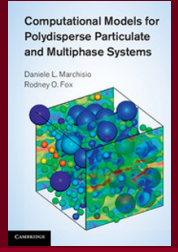
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



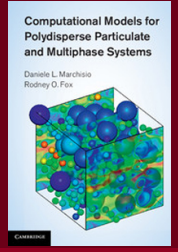
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



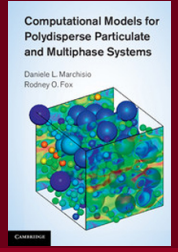
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



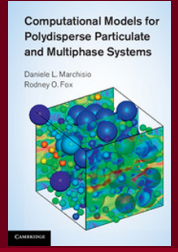
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



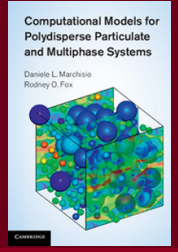
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



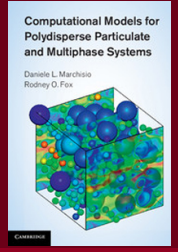
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



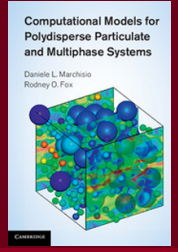
Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)



Observations Relevant to CIT

- Euler–Lagrange simulation
  - Collisions are treated exactly (i.e. no kinetic theory model)
  - Coupling to fluid uses spatial filter ($\delta_f \approx d_p$)
  - Provides most accurate approximation of flow physics
- Euler–Euler simulation
  - Collisions are not treated exactly (anisotropic BGK closure)
  - Coupling to fluid is exact
  - Agreement with EL simulation is satisfactory for collisional flow
- Turbulence model for homogeneous CIT
  - Rigorous derivation starting from Eulerian transport equations
  - Unclosed terms can be found from EL or EE simulations
  - Multiphase Reynolds-stress model (RSM)

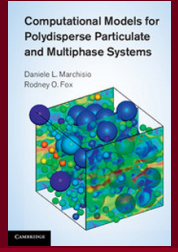


Glossary of Terms for CIT

Due to spatial homogeneity, the turbulence model has production, exchange, dissipation and reorientation terms

Key	Physical meaning
CD	Dissipation/reorientation of $\mathbf{P}$ due to collisions
DE	Exchange of momentum/Reynolds stresses/$\mathbf{P}$ due to fluid drag
DP	Production of $\langle u_{f,1}'''^2 \rangle_f$ by fluid drag due to clusters
PE	Exchange of momentum/Reynolds stresses due to fluid pressure
PS	Pressure-strain reorientation of Reynolds stresses
VD	Viscous dissipation of Reynolds stresses
VE	Exchange of momentum/Reynolds stresses due to viscous stresses

Note: CIT is nearly 1-D turbulence with $\langle u_{f,1}'''^2 \rangle_f \gg \langle u_{f,2}'''^2 \rangle_f = \langle u_{f,3}'''^2 \rangle_f$



Reynolds-Averaged Equations for CIT I

Statistically Homogeneous

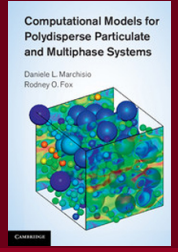
- **Particle phase:** phase average $\langle\cdot\rangle_p = \langle\alpha_p\cdot\rangle/\langle\alpha_p\rangle$

$$\frac{\mathrm{d}\langle u_{p,1}\rangle_p}{\mathrm{d}t} = \frac{1}{\tau_p}\left(\langle u_{f,1}\rangle_p - \langle u_{p,1}\rangle_p\right) \qquad \text{DE}$$

$$+ \frac{1}{\rho_p}\left(\left\langle\frac{\partial\sigma_{f,1i}}{\partial x_i}\right\rangle_p - \left\langle\frac{\partial p_f}{\partial x_1}\right\rangle_p\right) \qquad \text{VE/PE}$$

$$+ g$$

Re_p	$\frac{\langle u_{p,1}\rangle_p}{\mathcal{V}}$	$\frac{\langle u_{f,1}\rangle_p}{\mathcal{V}}$	DE	PE	VE	g
0.25	-3.009	-1.891	2.236	0.007	10^{-7}	-2
0.5	-2.476	-1.420	4.224	0.013	10^{-7}	-4
1.0	-2.278	-1.246	8.255	0.029	10^{-7}	-8



Reynolds-Averaged Equations for CIT II

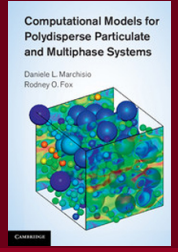
Statistically Homogeneous

$$\frac{1}{2}\frac{\mathrm{d}\langle u''^2_{p,1}\rangle_p}{\mathrm{d}t} = \frac{1}{\tau_p}\left(\langle u'''_{f,1}u''_{p,1}\rangle_p - \langle u''^2_{p,1}\rangle_p\right) \qquad \text{DE}$$

$$+\left\langle \Theta_p \frac{\partial u''_{p,1}}{\partial x_1}\right\rangle_p - \left\langle \sigma_{p,1i}\frac{\partial u''_{p,1}}{\partial x_i}\right\rangle_p \qquad \text{PS/VD}$$

$$+\frac{1}{\rho_p}\left(\left\langle u''_{p,1}\frac{\partial \sigma'_{f,1i}}{\partial x_i}\right\rangle_p - \left\langle u''_{p,1}\frac{\partial p'_f}{\partial x_1}\right\rangle_p\right) \qquad \text{VE/PE}$$

Re_p	$\frac{\langle u''^2_{p,1}\rangle_p}{2k_p}$	PS	VD	DE	PE	VE
0.25	0.809	-0.022	-0.043	0.103	-0.001	10^{-9}
0.5	0.788	-0.075	-0.141	0.274	-0.001	10^{-9}
1.0	0.814	-0.275	-0.539	0.879	-0.005	10^{-9}



Reynolds-Averaged Equations for CIT III

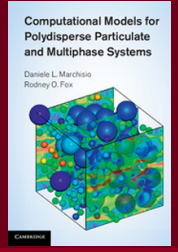
Statistically Homogeneous

$$\frac{1}{2}\frac{\mathrm{d}\langle u''^2_{p,2}\rangle_p}{\mathrm{d}t} = \frac{1}{\tau_p}\left(\langle u'''_{f,2}u''_{p,2}\rangle_p - \langle u''^2_{p,2}\rangle_p\right) \quad \text{DE}$$

$$+\left\langle \Theta_p \frac{\partial u''_{p,2}}{\partial x_2}\right\rangle_p - \left\langle \sigma_{p,2i}\frac{\partial u''_{p,2}}{\partial x_i}\right\rangle_p \quad \text{PS/VD}$$

$$+\frac{1}{\rho_p}\left(\left\langle u''_{p,2}\frac{\partial \sigma'_{f,2i}}{\partial x_i}\right\rangle_p - \left\langle u''_{p,2}\frac{\partial p'_f}{\partial x_2}\right\rangle_p\right) \quad \text{VE/PE}$$

Re_p	$\frac{\langle u''^2_{p,2}\rangle_p}{2k_p}$	PS	VD	DE	PE	VE
0.25	0.096	0.008	-0.013	0.008	10^{-4}	-10^{-5}
0.5	0.106	0.027	-0.044	0.021	10^{-4}	-10^{-5}
1.0	0.093	0.093	-0.154	0.062	10^{-3}	-10^{-4}



Reynolds-Averaged Equations for CIT IV

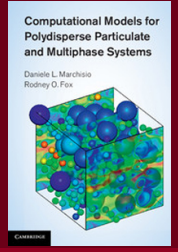
Statistically Homogeneous

$$\frac{1}{2}\frac{\mathrm{d}\langle P_{11}\rangle_p}{\mathrm{d}t} = -\frac{1}{\tau_p}\langle P_{11}\rangle_p \qquad \text{DE}$$

$$-\left\langle \Theta_p \frac{\partial u''_{p,1}}{\partial x_1}\right\rangle_p + \left\langle \sigma_{p,1i}\frac{\partial u''_{p,1}}{\partial x_i}\right\rangle_p \qquad \text{PS/VD}$$

$$+\frac{6}{\sqrt{\pi}d_p}\langle \alpha_p \Theta_p^{1/2}\left(\Delta^*_{11} - P_{11}\right)\rangle_p \qquad \text{CD}$$

Re_p	$\frac{\langle P_{11}\rangle_p}{3\langle\Theta_p\rangle_p}$	PS	VD	DE	CD
0.25	0.621	0.022	0.043	-0.082	-0.030
0.5	0.528	0.075	0.141	-0.192	-0.096
1.0	0.507	0.275	0.539	-0.480	-0.402



Reynolds-Averaged Equations for CIT V

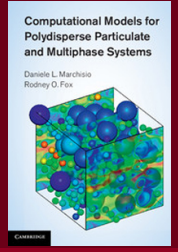
Statistically Homogeneous

$$\frac{1}{2}\frac{\mathrm{d}\langle P_{22}\rangle_p}{\mathrm{d}t} = -\frac{1}{\tau_p}\langle P_{22}\rangle_p \qquad \text{DE}$$

$$-\left\langle \Theta_p \frac{\partial u''_{p,2}}{\partial x_2}\right\rangle_p + \left\langle \sigma_{p,2i}\frac{\partial u''_{p,2}}{\partial x_i}\right\rangle_p \qquad \text{PS/VD}$$

$$+\frac{6}{\sqrt{\pi}d_p}\langle \alpha_p \Theta_p^{1/2}\left(\Delta^*_{22} - P_{22}\right)\rangle_p \qquad \text{CD}$$

Re_p	$\frac{\langle P_{22}\rangle_p}{3\langle\Theta_p\rangle_p}$	PS	VD	DE	CD
0.25	0.198	-0.008	0.013	-0.025	0.011
0.5	0.236	-0.027	0.044	-0.086	0.030
1.0	0.246	-0.093	0.154	-0.233	0.119



Reynolds-Averaged Equations for CIT VI

Statistically Homogeneous

- **Fluid phase:** phase-average velocity $\langle u_{f,1} \rangle_f = 0$

$$\frac{1}{2}\frac{\mathrm{d}\langle u_{f,1}'''^2 \rangle_p}{\mathrm{d}t} = \frac{\varphi}{\tau_p}\left(\langle u_{f,1}''' u_{p,1}'' \rangle_p - \langle u_{f,1}'''^2 \rangle_p\right) + \frac{\varphi}{\tau_p}\langle u_{f,1}''' \rangle_p \langle u_{p,1} \rangle_p \qquad \text{DE/DP}$$

$$+ \frac{1}{\rho_f}\left(\left\langle p_f \frac{\partial u_{f,1}'''}{\partial x_1}\right\rangle - \left\langle \sigma_{f,1i}\frac{\partial u_{f,1}'''}{\partial x_i}\right\rangle\right) \qquad \text{PS/VD}$$

$$- \frac{\varphi}{\rho_p}\left(\left\langle u_{f,1}''' \frac{\partial \sigma_{f,1i}'}{\partial x_i}\right\rangle_p + \left\langle u_{f,1}''' \frac{\partial p_f'}{\partial x_1}\right\rangle_p\right) \qquad \text{VE/PE}$$

Re_p	$\frac{\langle u_{f,1}'''^2 \rangle_p}{2k_f}$	PS	VD	DP	DE	PE	VE
0.25	0.810	-0.465	-0.219	5.747	-5.329	0.012	-10^{-7}
0.5	0.801	-1.380	-0.365	14.21	-12.94	0.030	-10^{-7}
1.0	0.815	-4.643	-0.719	45.89	-41.47	0.116	-10^{-7}

Reynolds-Averaged Equations for CIT VII

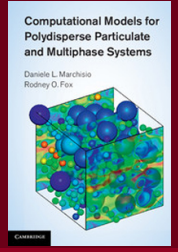
Statistically Homogeneous

$$\frac{1}{2}\frac{\mathrm{d}\langle u_{f,2}'''^2\rangle_p}{\mathrm{d}t} = \frac{\varphi}{\tau_p}\left(\langle u_{f,2}''' u_{p,2}''\rangle_p - \langle u_{f,2}'''^2\rangle_p\right) \qquad \text{DE}$$

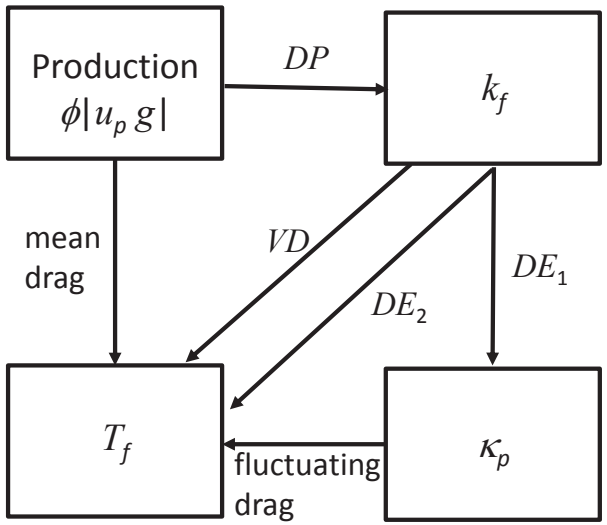
$$+ \frac{1}{\rho_f}\left(\left\langle p_f \frac{\partial u_{f,2}'''}{\partial x_2}\right\rangle - \left\langle \sigma_{f,2i}\frac{\partial u_{f,2}'''}{\partial x_i}\right\rangle\right) \qquad \text{PS/VD}$$

$$- \frac{\varphi}{\rho_p}\left(\left\langle u_{f,2}''' \frac{\partial \sigma_{f,2i}'}{\partial x_i}\right\rangle_p + \left\langle u_{f,2}''' \frac{\partial p_f'}{\partial x_2}\right\rangle_p\right) \qquad \text{VE/PE}$$

Re_p	$\frac{\langle u_{f,1}'''^2\rangle_p}{2k_f}$	PS	VD	DE	PE	VE
0.25	0.095	0.237	-0.012	-0.266	-0.003	10^{-4}
0.5	0.099	0.705	-0.023	-0.749	-0.010	10^{-4}
1.0	0.093	2.384	-0.044	-2.490	-0.037	10^{-3}



Kinetic Energy Balance in Fully Developed CIT

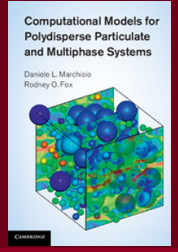


Re_p	$\frac{2k_f}{\mathcal{V}^2}$	$\frac{2\kappa_p}{\mathcal{V}^2}$	$\frac{k_p}{\kappa_p}$
0.25	19.52	13.63	0.90
0.5	11.27	7.78	0.88
1.0	8.04	5.41	0.89

DE_1: drag exchange $\approx 22\%$

DE_2: drag dissipation $\approx 78\%$

In CIT, $\approx$ 90% of particle-phase KE is spatially correlated



Reynolds-Stress Model for Fully Developed CIT I

- Particle phase:

$$\langle u_{p,1}\rangle_p = \langle u'''_{f,1}\rangle_p + \mathcal{V}$$

where drift velocity is
$\langle u'''_{f,1}\rangle_p = C_g\langle u_{p,1}\rangle_p$ with C_g =0.63, 0.57, 0.55 (vs. Re_p)

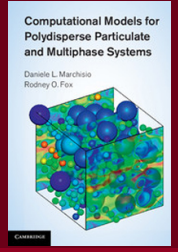
$$0 = \tau_p\mathcal{R}_{p,11} - \tau_p\epsilon_{p,11} + \beta_{fp}\langle u''^2_{p,1}\rangle_p^{1/2}\langle u'''^2_{f,1}\rangle_f^{1/2} - \langle u''^2_{p,1}\rangle_p$$

$$0 = \tau_p\mathcal{R}_{p,22} - \tau_p\epsilon_{p,22} + \beta_{fp}\langle u''^2_{p,2}\rangle_p^{1/2}\langle u'''^2_{f,2}\rangle_f^{1/2} - \langle u''^2_{p,2}\rangle_p$$

where $\beta_{fp} = 0.876$ and pressure-redistribution tensor is modeled as

$$\mathcal{R}_{p,ii} = -C_R\frac{\varepsilon_p}{k_p}\left(\langle u''^2_{p,i}\rangle_p - \frac{2}{3}k_p\right) \text{ with } C_R = 0.179 \text{ (vs. 1.8)}$$

$$\epsilon_{p,ii} = 2\left[f_s\frac{\langle u''^2_{p,i}\rangle_p}{2k_p} + (1-f_s)\frac{1}{3}\right]\varepsilon_p \text{ with } f_s = 0.93$$



Reynolds-Stress Model for Fully Developed CIT II

$$0 = \tau_p \epsilon_{p,11} - \langle P_{11} \rangle_p + C_c \frac{\tau_p}{\tau_c} \left(\Delta_{11} - \langle P_{11} \rangle_p \right)$$

$$0 = \tau_p \epsilon_{p,22} - \langle P_{22} \rangle_p + C_c \frac{\tau_p}{\tau_c} \left(\Delta_{22} - \langle P_2 \rangle_p \right)$$

with $C_c = 2$ and $\Delta_{ii} = \frac{1}{4}(1+e)^2 \langle \Theta_p \rangle_p + \frac{1}{4}(1-e)^2 \langle P_{ii} \rangle_p$

- Fluid phase:

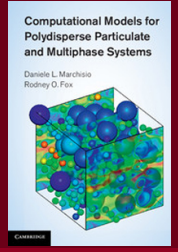
$$0 = \tau_p \mathcal{R}_{f,11} - \tau_p \epsilon_{f,11} + \varphi \beta_{fp} \langle u''^2_{p,1} \rangle_p^{1/2} \langle u'''^2_{f,1} \rangle_f^{1/2} - \varphi \beta_f \langle u'''^2_{f,1} \rangle_f + \mathcal{DP}_{11}$$

$$0 = \tau_p \mathcal{R}_{f,22} - \tau_p \epsilon_{f,22} + \varphi \beta_{fp} \langle u''^2_{p,2} \rangle_p^{1/2} \langle u'''^2_{f,2} \rangle_f^{1/2} - \varphi \beta_f \langle u'''^2_{f,2} \rangle_f$$

where $\beta_f = 1.03$ and pressure-redistribution tensor is modeled as

$$\mathcal{R}_{f,ii} = -C_R \frac{\varepsilon_f}{k_f} \left(\langle u'''^2_{f,i} \rangle_f - \frac{2}{3} k_f \right) - C_{\mathcal{D}} \left(\mathcal{DP}_{ii} - \frac{2}{3} \mathcal{DP} \right) \text{ with } C_{\mathcal{D}} = 0.139 \text{ (vs. 0.6)}$$

$$\epsilon_{f,ii} = 2 \left[f_s \frac{\langle u'''^2_{f,i} \rangle_f}{2k_f} + (1 - f_s) \frac{1}{3} \right] \varepsilon_f, \; \mathcal{DP}_{11} = \frac{2\varphi}{\tau_p} C_g \langle u_{p,1} \rangle_p^2, \; \mathcal{DP}_{22} = 0, \; \mathcal{DP} = \frac{1}{2} \mathcal{DP}_{11}$$



Dissipation Rate Model

Form of dissipation model is critical for overall performance!

$$0 = -C_{2,\epsilon} St_f \varepsilon_f + 2C_3 \varphi \left(\beta_{fp} \varepsilon_f^{1/2} \varepsilon_p^{1/2} - \beta_f \varepsilon_f \right) + C_4 St_p \mathcal{DP}$$

$$0 = -C_{2,\epsilon} St_p \varepsilon_p + 2C_3 \left(\beta_{fp} \varepsilon_f^{1/2} \varepsilon_p^{1/2} - \varepsilon_p \right)$$

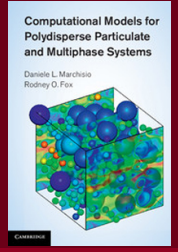
with $C_{2,\epsilon} = C_4 = 1.92$ and $C_3 = 0.736$

Stokes numbers are defined by $St_f = \frac{\tau_p \varepsilon_f}{k_f}$ and $St_p = \frac{\tau_p \varepsilon_p}{k_p}$

In fully developed CIT:

$$St_f = 0.25 \quad St_p = 0.19 \quad \frac{k_f}{k_p} \approx 1.6$$

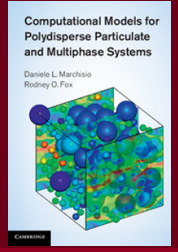
Compared to sheared turbulence, C_R and $C_{\mathcal{D}}$ are significantly smaller



Concluding Remarks on CIT I

Cluster-Induced Turbulence

- CIT arises due to mean velocity difference
- Velocity difference arises due to body force or inlet conditions
- With constant velocity difference, CIT is homogeneous and stationary
- TKE is produced by fluid velocity fluctuations seen by the particles
- CIT is anisotropic in direction of mean velocity, but otherwise diagonal



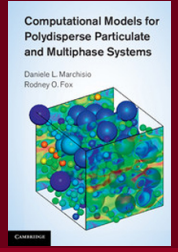
Concluding Remarks on CIT II

Mesoscale Simulation of CIT

- Lagrangian particle tracking best captures flow physics
- Wide separation of scales between filter and cluster sizes exists
- Local Eulerian fields can be extracted with α_p-dependent filter
- Converged statistics require very large domains $\gg \mathcal{L}$!

Turbulence Model for CIT

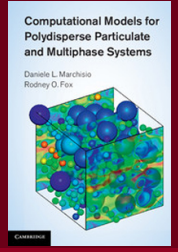
- CIT depends on a relatively small number of correlations
- Drift velocity changes mean drag and produces fluid TKE through C_g
- Reynolds-stress model needed to describe anisotropy
- Principal modeling challenges stem from ε_f and ε_p, but choices limited



Concluding Remarks on CIT III

Some Open Questions

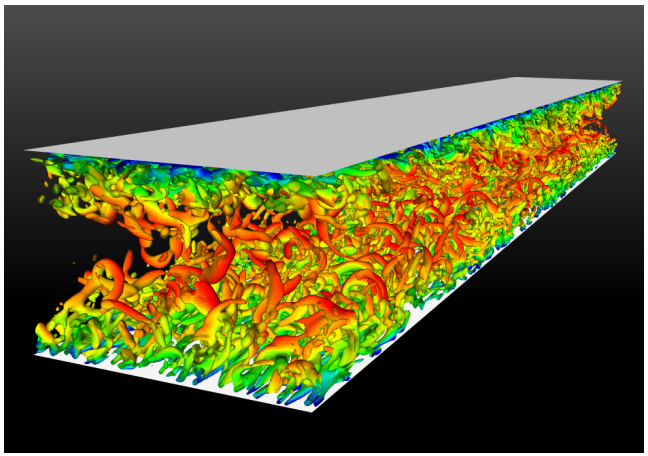
- What exactly determines the cluster size distribution?
- Are CIT statistics (one-point, two-point, ...) self similar?
- How do CIT statistics depend on dimensionless parameters (Re_p, φ, ...)?
- How do statistics change when homogeneous mean shear is added?



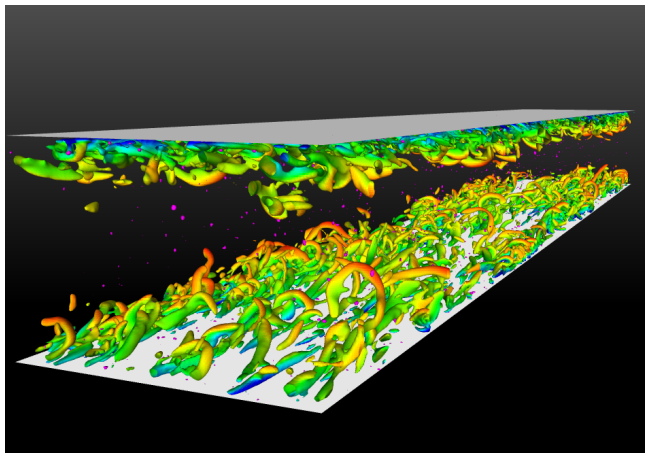
Particle-Laden Channel Flow

Effect of mass loading

0.2



1.2



20