Strategies for tackling the computational cost of modeling reacting fluids and related problems

Kyle Niemeyer

School of Mechanical, Industrial, and Manufacturing Engineering

Oregon State University

✉ kyle.niemeyer@oregonstate.edu

@kyleniemeyer

niemeyer-research-group.github.io

27 January 2023

Today's talk

- Introduce applications of high-fidelity simulations of reacting fluid flows
- Discuss the challenges of incorporating detailed chemical kinetics models, and briefly summarize methods for reducing the cost
- Describe our recent work on using adaptive preconditioning to accelerate implicit integration of stiff chemical kinetics, and other recent updates to Cantera
- Illustrate how we have extended methods from combustion to other domains such as modeling of ocean biogeochemistry and neutron transport
- Discuss some alternate career paths for graduate students

Acknowledgments: Students

Anthony Walker

Diba Behnoudfar

Jackson Morgan

Malik Jordan

Taylor Coddington

Jordan Peters

Ali Martz

Acknowledgments: Collaborators

David Blunck

Todd Palmer

Camille Palmer

Peter Hamlington

Raymond Speth

Nikki Lovenduski

Vi Rapp

Richard West

Steven DeCaluwe

Franklin Goldsmith

Acknowledgments: Funders

NSF

SERDP
DOD ▪ EPA ▪ DOE

ALFRED P. SLOAN
FOUNDATION

DEPARTMENT OF ENERGY
UNITED STATES OF AMERICA

NASA

NETL

NNSA
National Nuclear Security Administration

Sandia
National
Laboratories

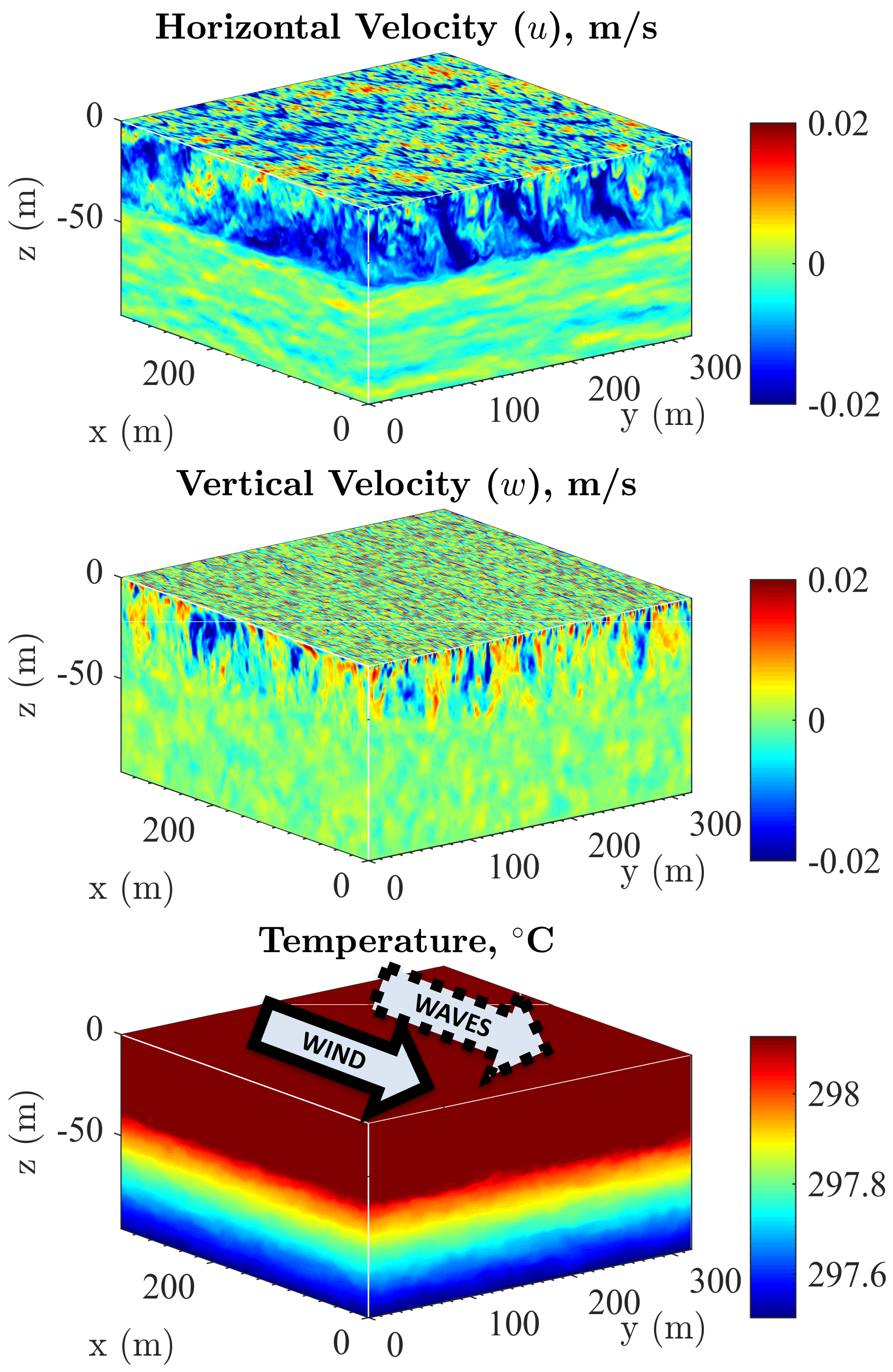
Chevron

Why reacting fluid flows?

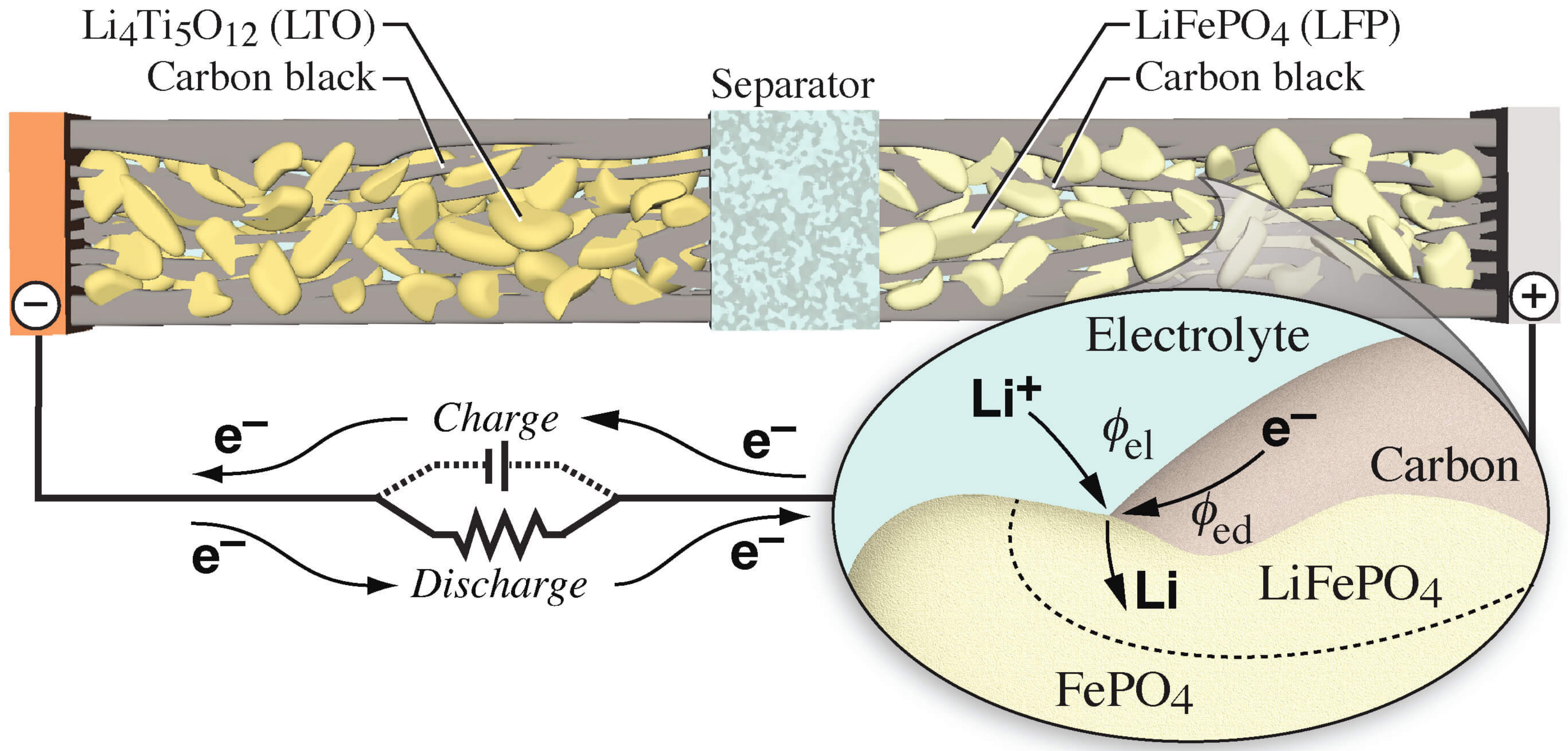
Reacting flow applications

Combustion and fire

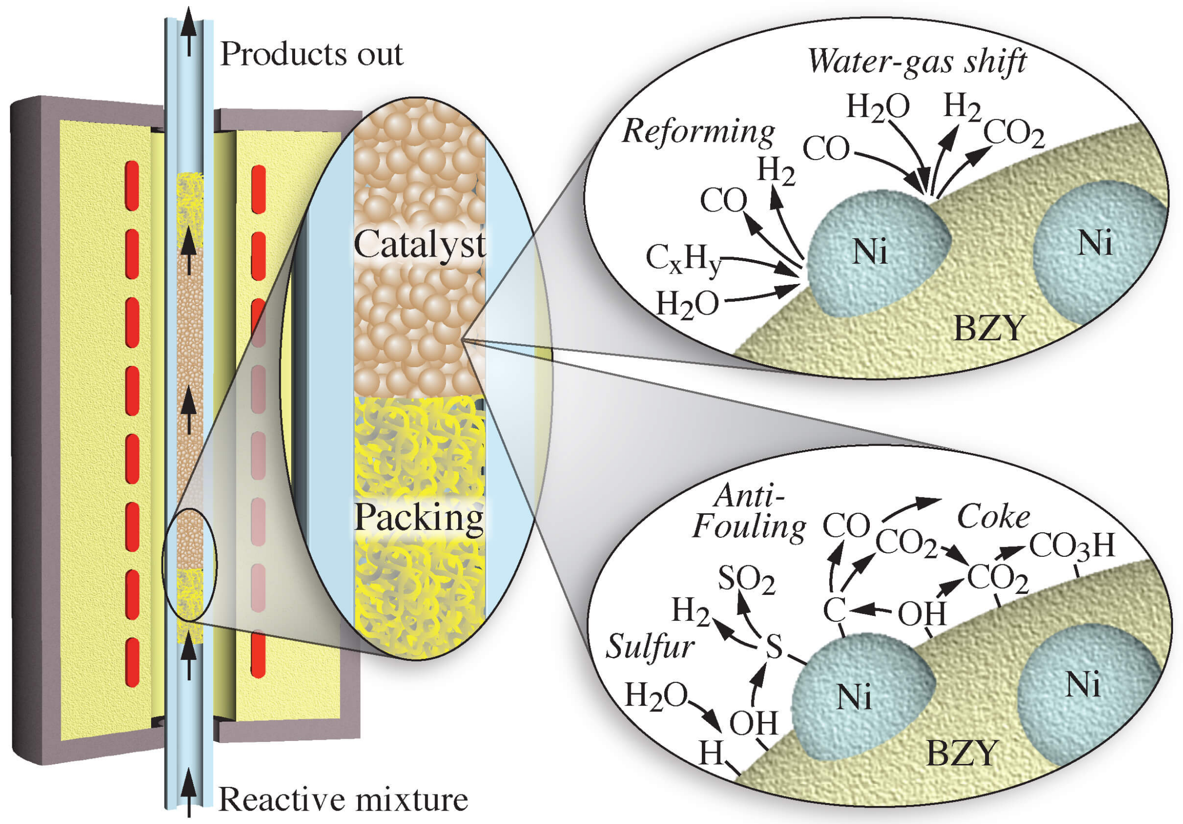
Atmospheric chemistry



Geophysical flows



Electrochemistry



Heterogeneous catalysis

The source of the problem:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \ \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f}$$

$$\frac{\partial \rho T}{\partial t} + \nabla \cdot (\rho \mathbf{u} T) = \nabla \cdot (\rho \alpha \nabla T) + \rho \dot{\omega}_T - \frac{1}{c_p} \sum_i^{N_{\mathrm{species}}} c_{p,i} \mathbf{j}_i \cdot \nabla T + \frac{\rho \alpha}{c_p} \nabla c_p \cdot \nabla T$$

$$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y_i) = -\nabla \cdot \mathbf{j}_i + \rho \dot{\omega}_i$$

The source of the problem:

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho \mathbf{u}) = 0$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla\cdot(\rho\ \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla\cdot \boldsymbol{\tau} + \mathbf{f}$$

$$\frac{\partial \rho T}{\partial t} + \nabla\cdot(\rho \mathbf{u} T) = \nabla\cdot(\rho\alpha\nabla T) + \boxed{\rho\dot{\omega}_T} - \frac{1}{c_p}\sum_i^{N_{\mathrm{species}}} c_{p,i}\mathbf{j}_i\cdot\nabla T + \frac{\rho\alpha}{c_p}\nabla c_p\cdot\nabla T$$

$$\frac{\partial \rho Y_i}{\partial t} + \nabla\cdot(\rho \mathbf{u} Y_i) = -\nabla\cdot\mathbf{j}_i + \boxed{\rho\dot{\omega}_i}$$

$$\dot{\omega}_T = -c_p^{-1}\sum_i^{N_{\mathrm{species}}} h_i(T)\dot{\omega}_i \qquad \dot{\omega}_i = \frac{\rho}{W_i}\frac{dY_i}{dt}$$

Hydrogen oxidation

$$2\,H_2 + O_2 \longrightarrow 2\,H_2O$$

Hydrogen oxidation

$H_2 + M \longleftrightarrow 2H + M$

$O_2 + M \longleftrightarrow 2O + M$

$H + O_2 \longleftrightarrow O + OH$

$H_2 + O \longleftrightarrow H + OH$

$H_2 + OH \longleftrightarrow H_2O + H$

$2OH \longleftrightarrow H_2O + O$

$O + H + M \longleftrightarrow OH + M$

$H_2O_2 + OH \longleftrightarrow HO_2 + H_2O$

$2H_2O \longleftrightarrow H + OH + H_2O$

$H + O_2 \longleftrightarrow HO_2$

$H_2 + O_2 \longleftrightarrow HO_2 + H$

$H + HO_2 \longleftrightarrow 2OH$

$HO_2 + O \longleftrightarrow O_2 + OH$

$HO_2 + OH \longleftrightarrow H_2O + O_2$

$2HO_2 \longleftrightarrow H_2O_2 + O_2$

$H_2O_2 \longleftrightarrow 2OH$

$H_2O_2 + H \longleftrightarrow H_2O + OH$

$H_2O_2 + H \longleftrightarrow HO_2 + H_2$

$H_2O_2 + O \longleftrightarrow OH + HO_2$

$H + OH + M \longleftrightarrow H_2O + M$

Chemistry = ODEs

(usually)

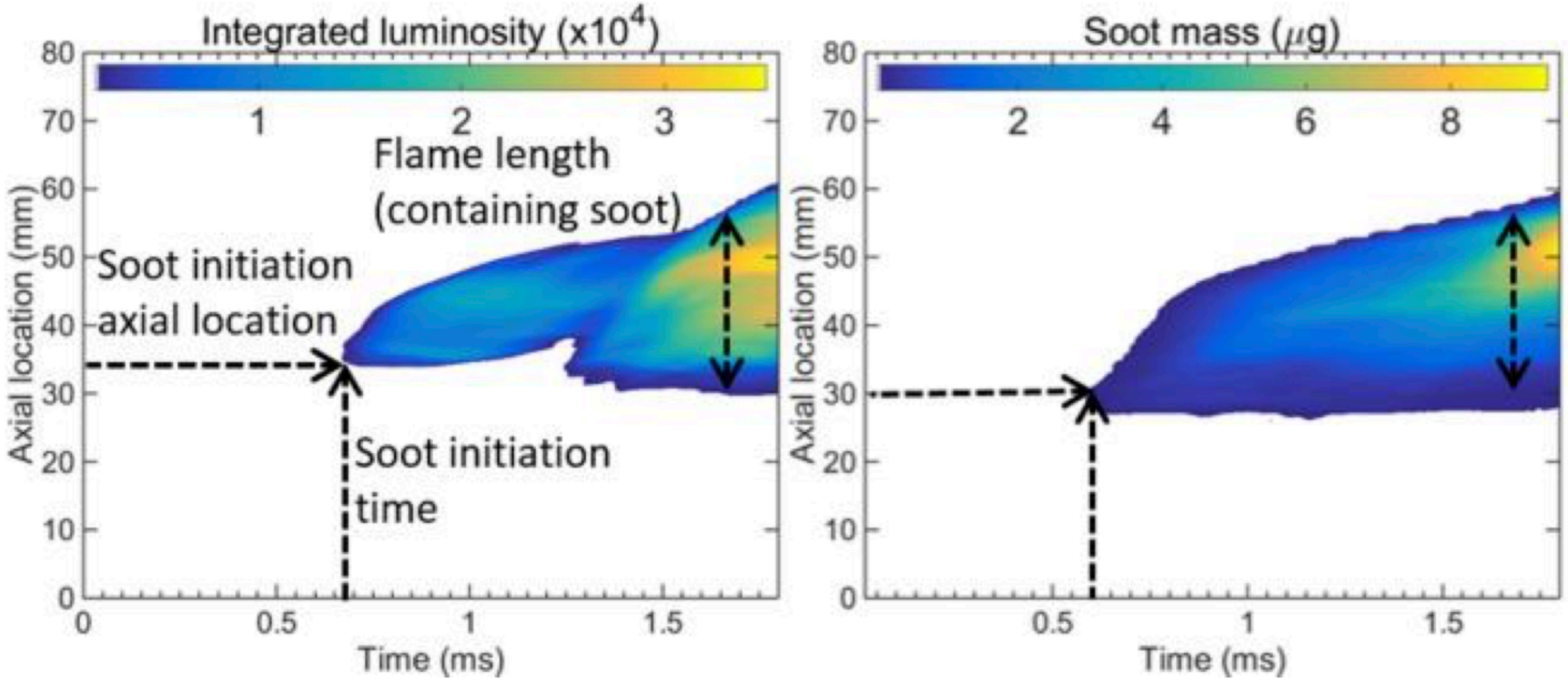
$$\begin{pmatrix} \frac{dY_1}{dt} \\ \frac{dY_2}{dt} \\ \vdots \\ \frac{dY_k}{dt} \end{pmatrix} \qquad \frac{dY_i}{dt} = \frac{W_i}{\rho}\omega_i$$

Large number of independent ODEs to integrate, often using implicit algorithms

Can be even more, and more complicated, for turbulent combustion!

OK, we need detailed chemistry—what's the issue?

Large-eddy simulation of diesel spray with 54-species *n*-dodecane model:



48,000 CPU core-hours for 2 ms after start of injection

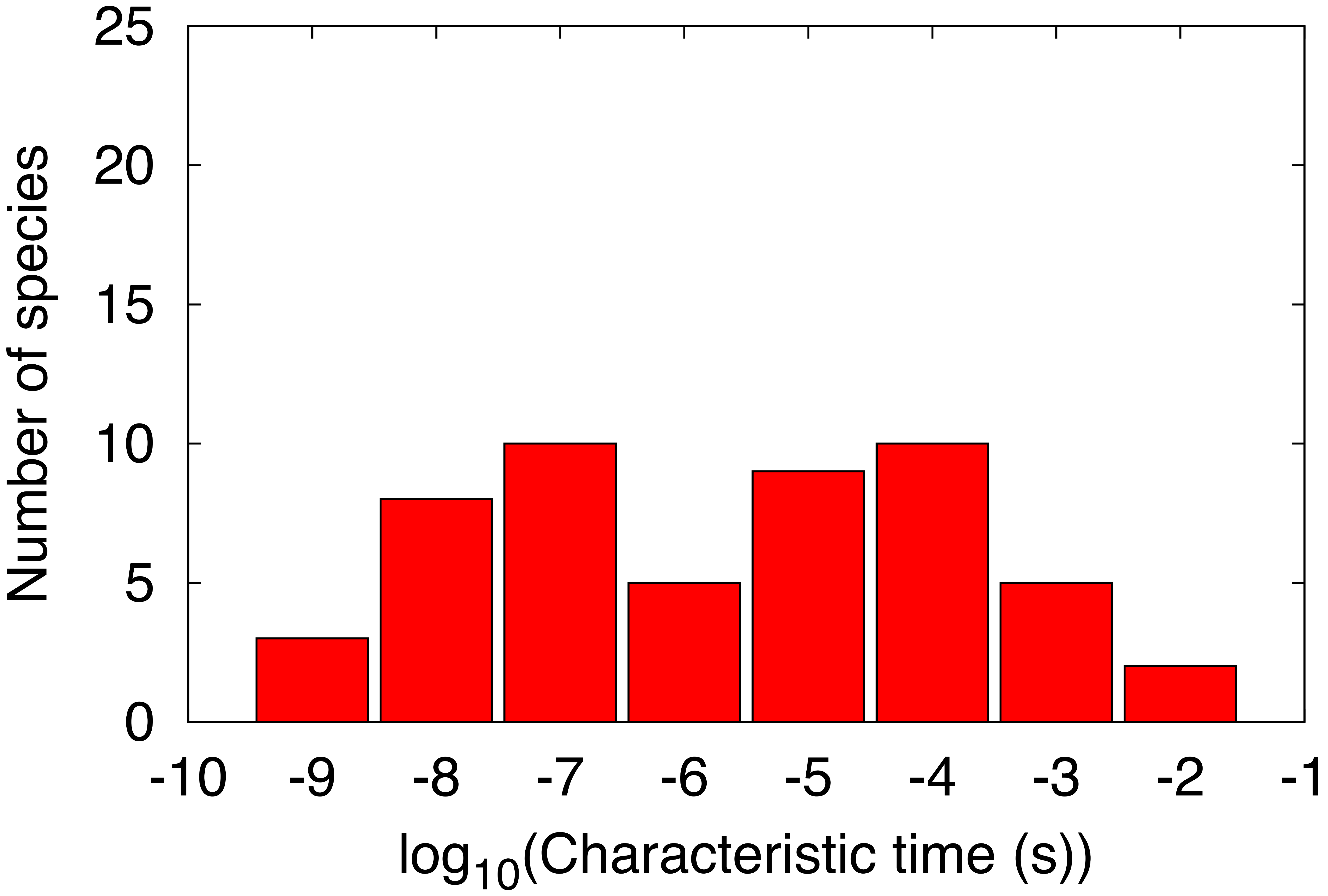
[1] A. A. Moiz et al. *Combust. Flame* 173 (2016): 123–131. doi:10.1016/j.combustflame.2016.08.005

What drives costs?

Stiffness

Size

Kinetic models exhibit high stiffness



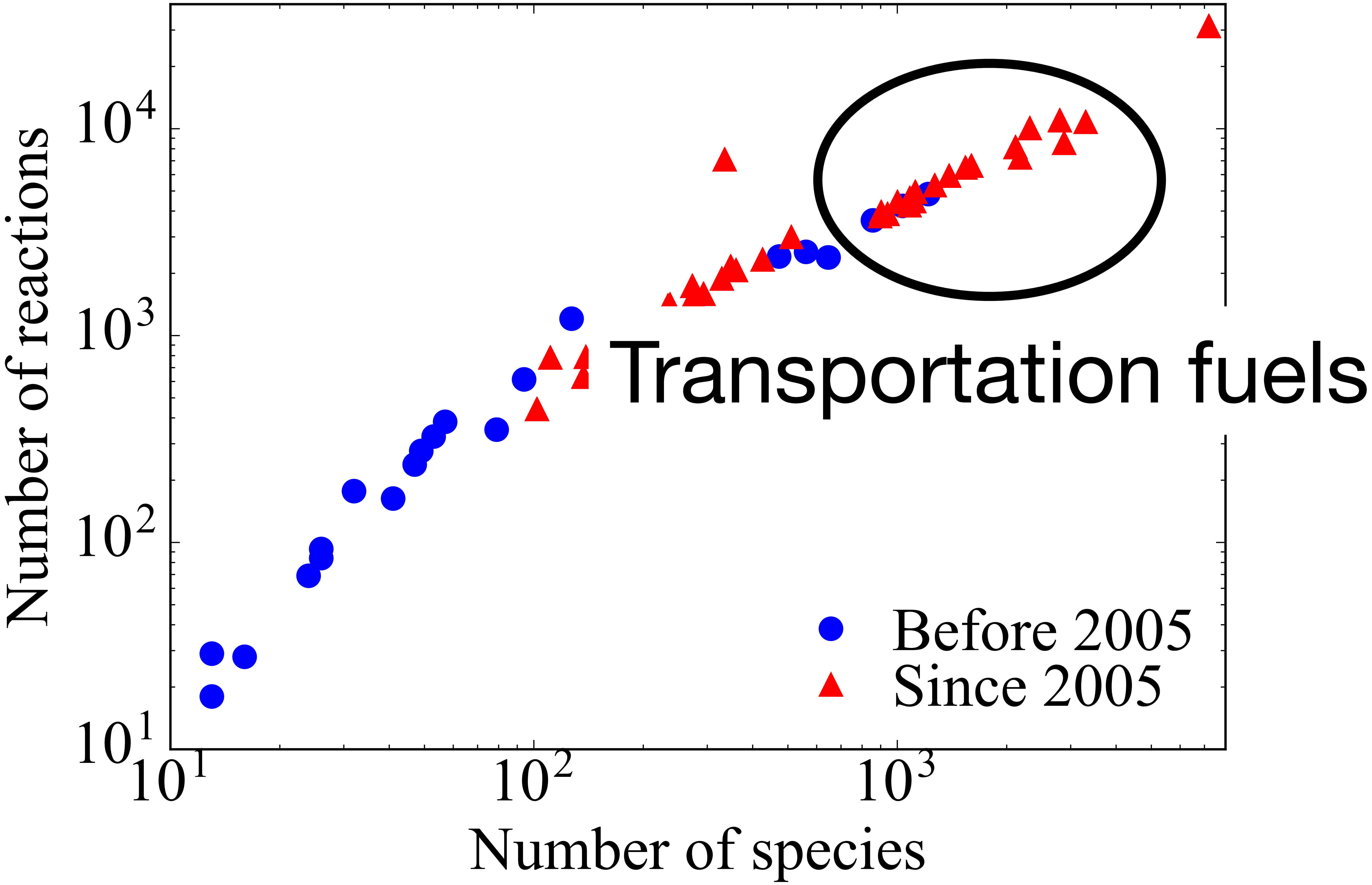
Characteristic creation times of methane oxidation

K. E. Niemeyer, N. J. Curtis, & C. J. Sung. Fall 2015 Meeting of the West. States Sect. Combust. Inst. Provo, UT, USA, Oct. 2015. doi:10.6084/m9.figshare.2075515.v1

Stiffness

- Wide range of species and reaction time scales
- Rapidly depleting radical species, fast reversible reactions
- Traditionally requires implicit integration algorithms (with some exceptions: DNS, high-speed flows)

Kinetic models can be large



Chemical kinetic model size for hydrocarbon oxidation

K. Niemeyer. Hydrocarbon chemical kinetic model survey. figshare. 2016. doi:10.6084/m9.figshare.3792660.v1

How to reduce the cost of kinetics

Model reduction

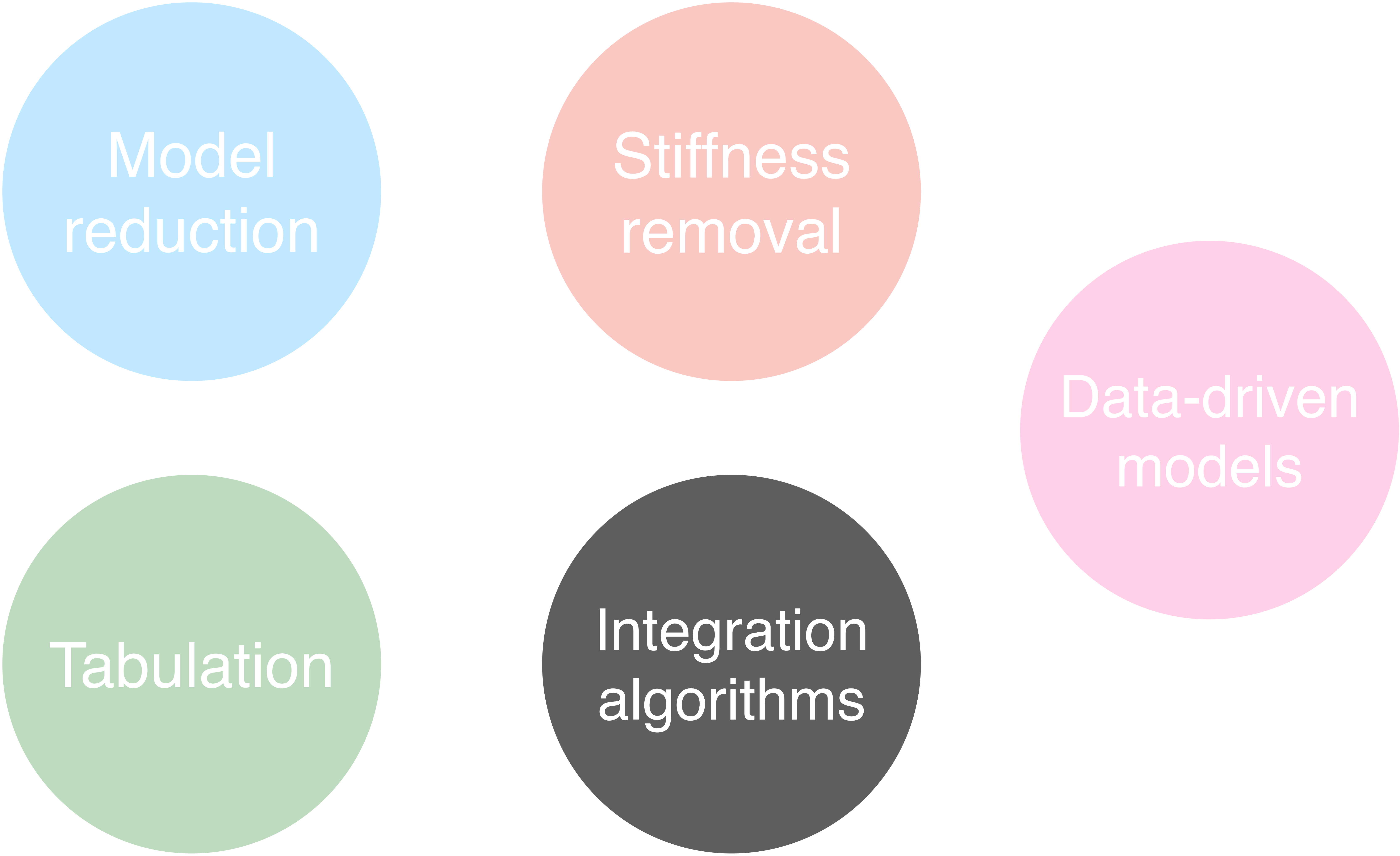
Stiffness removal

Data-driven models

Tabulation

Integration algorithms

How to reduce the cost of kinetics



Model reduction

Stiffness removal

Data-driven models

Tabulation

Integration algorithms

Accelerating implicit integration

- Solving with typical implicit algorithms requires evaluating and factorizing the Jacobian matrix to solve a linear system
- We can (significantly) speed up integration by combining a few steps:
  - Using a semi-analytical Jacobian with a mole-based system
  - Increasing sparsity by removing small, unimportant terms, and using sparse linear algebra operations
  - Preconditioning the iterative solution to the linear system of equations

M.J. McNenly, R.A. Whitesides, and D.L. Flowers. *Proc. Combust. Inst.*, 35(1) (2015) 581-587. https://doi.org/10.1016/j.proci.2014.05.113

A.S. Walker, R.L. Speth, and K.E. Niemeyer. *Proc. Combust. Inst.*, in press (2023). https://doi.org/10.1016/j.proci.2022.07.256

Overall approach: generalized adaptive preconditioning

Implicit algorithm: $H(Y) \equiv Y^{n+1} - Y^{n} + \Delta t \cdot F\left(t^{n+1}, Y^{n+1}\right) = 0$

Newton iteration: $H(Y) \approx H^{k-1} + \Delta t \cdot J\left(H^{k}\right) \cdot \left(H^{k} - H^{k-1}\right)$

Solution to linear system: $Y^{k+1} = Y^{k} - \left(I - J\left(Y^{k}\right)\right)^{-1} H\left(Y^{k}\right)$

Solve using **preconditioned** GMRES

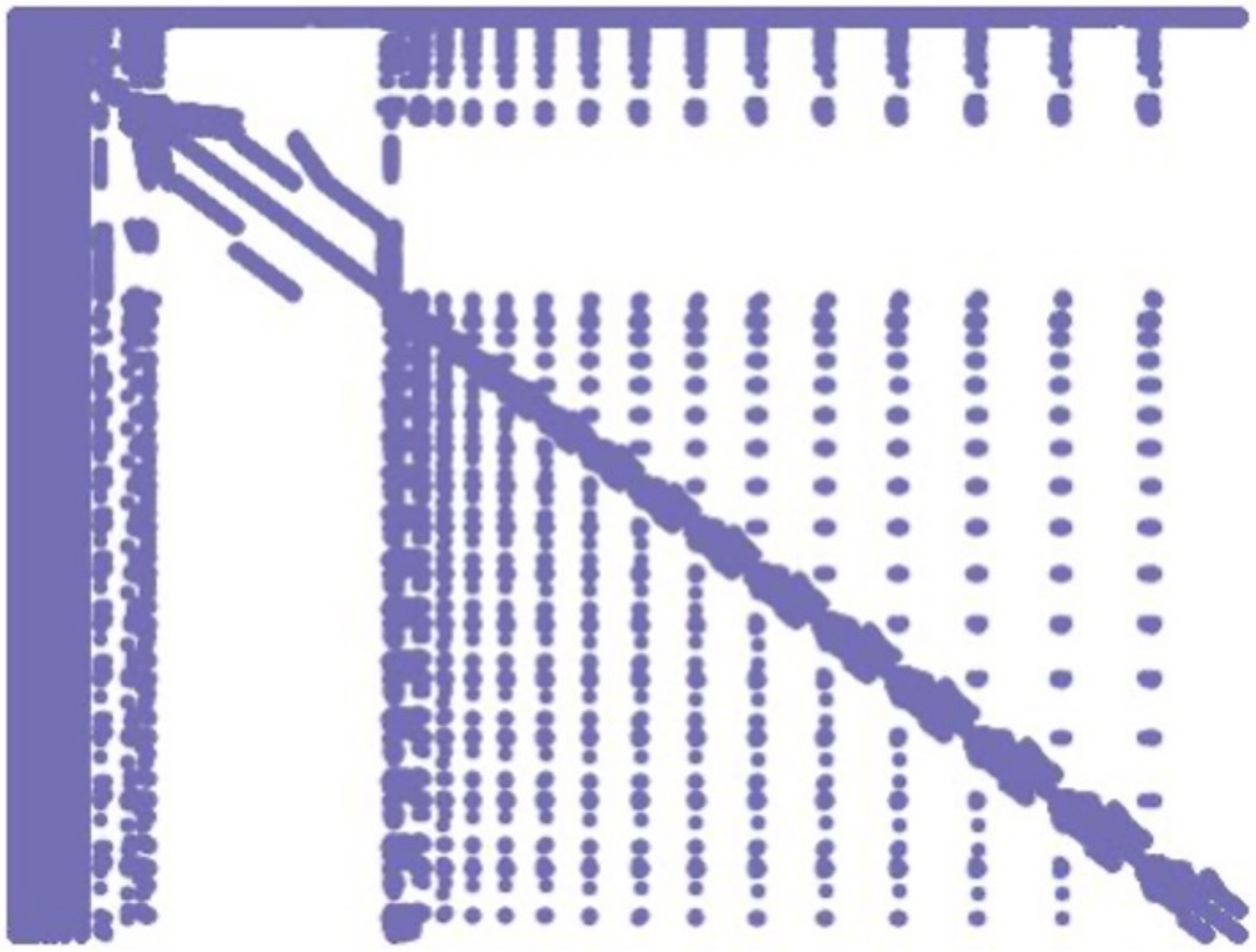
A.S. Walker, R.L. Speth, and K.E. Niemeyer. *Proc. Combust. Inst.*, in press (2023). https://doi.org/10.1016/j.proci.2022.07.256

Adaptive preconditioning

- Mass-fraction state vector

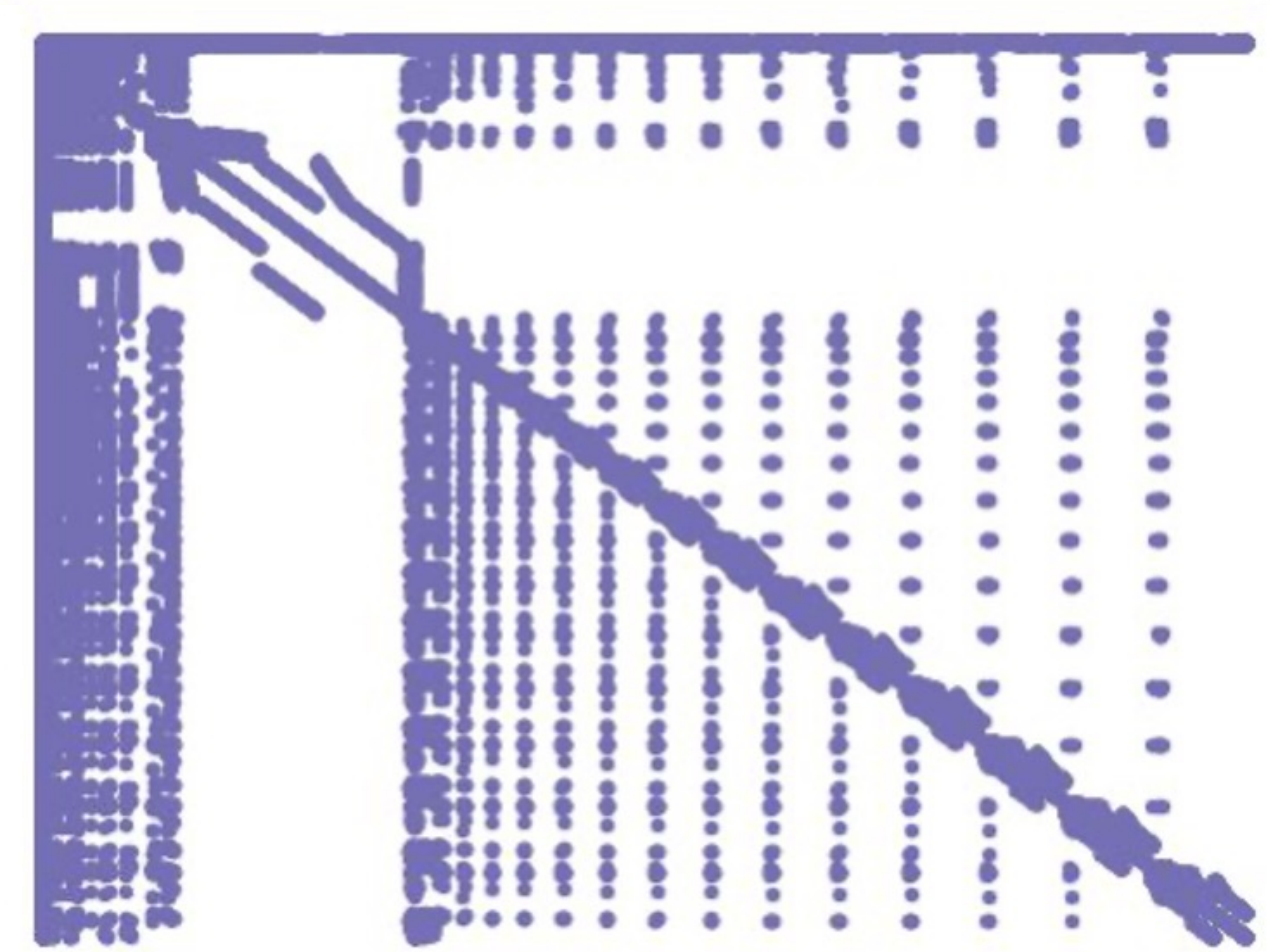
Adaptive preconditioning

- Mass-fraction state vector
- Mole state vector



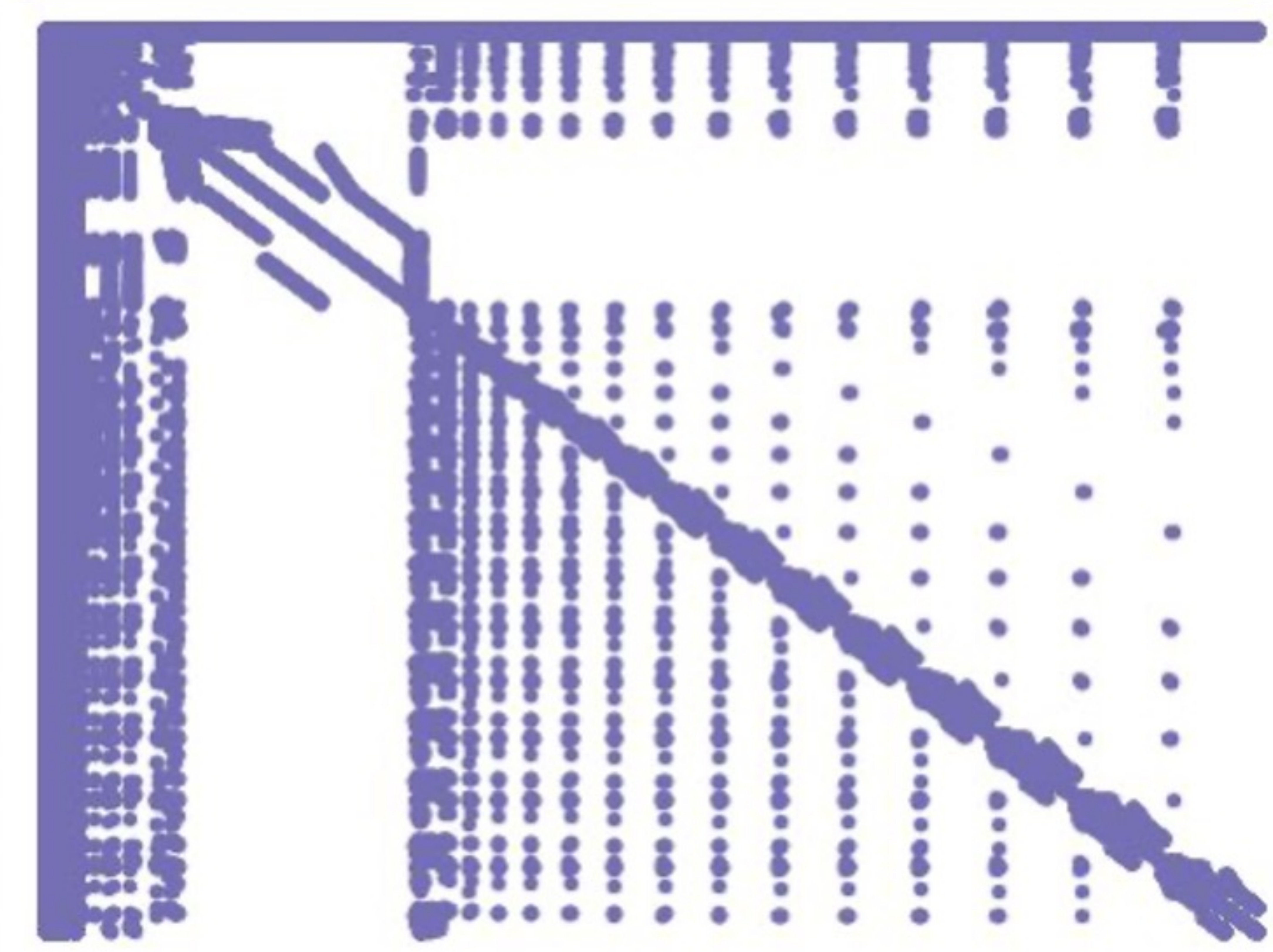
Adaptive preconditioning

- Mass-fraction state vector
- Mole state vector
- Remove third-body effects



Adaptive preconditioning

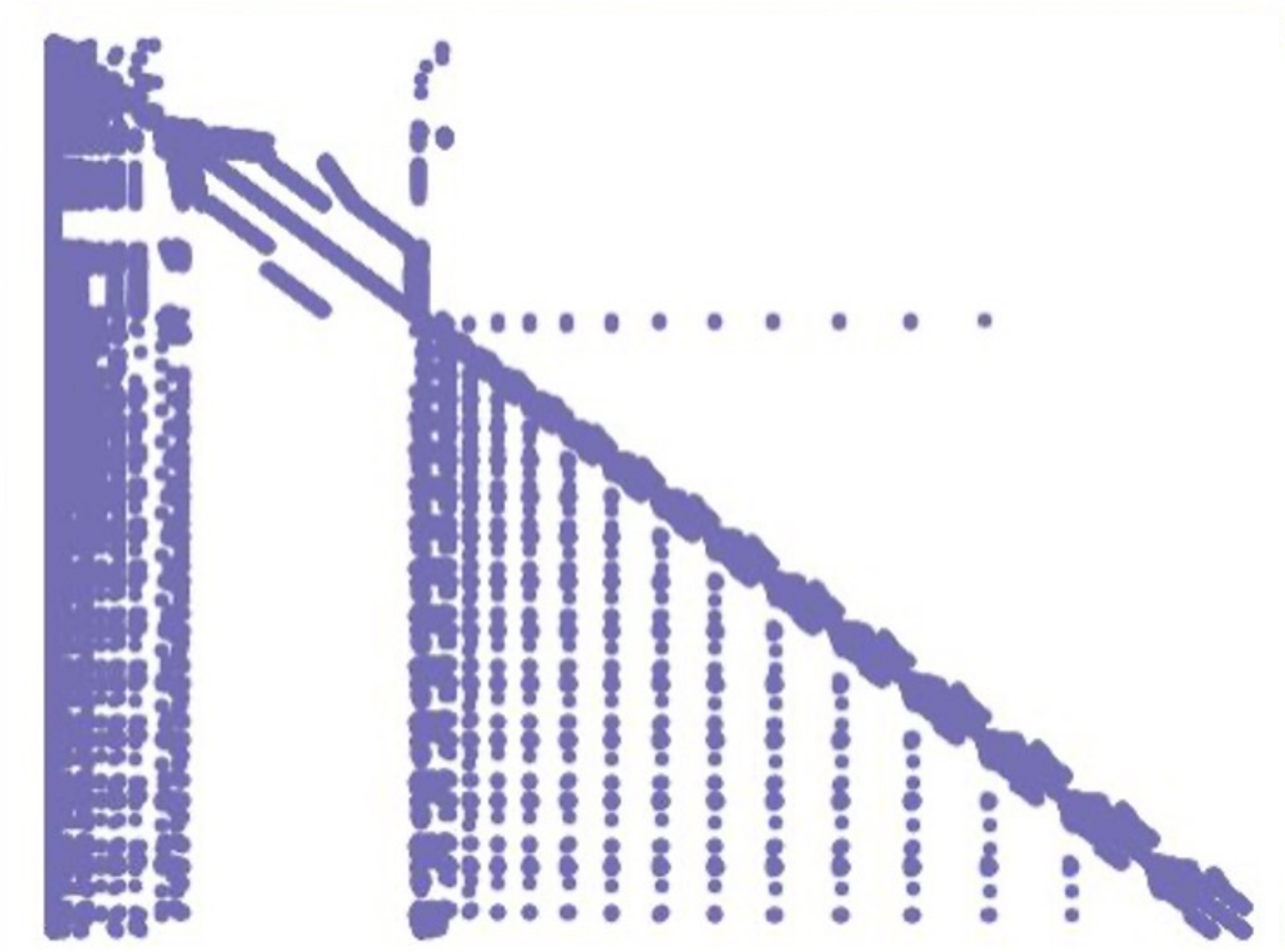
- Mass-fraction state vector
- Mole state vector
- Remove third-body effects
- Remove fall-off effects



Adaptive preconditioning

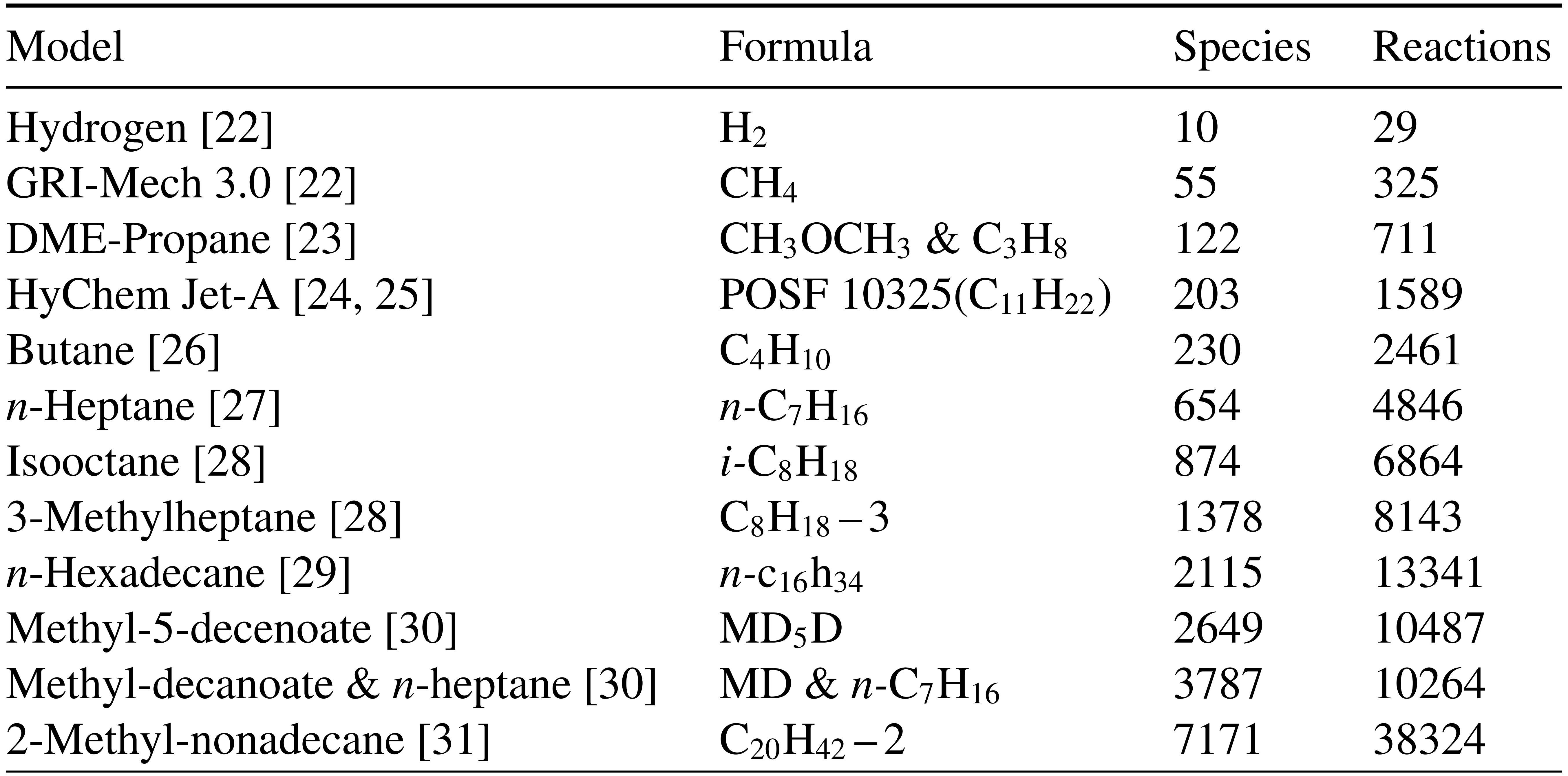
- Mass-fraction state vector
- Mole state vector
- Remove third-body effects
- Remove fall-off effects
- Apply threshold

Not applied to system!



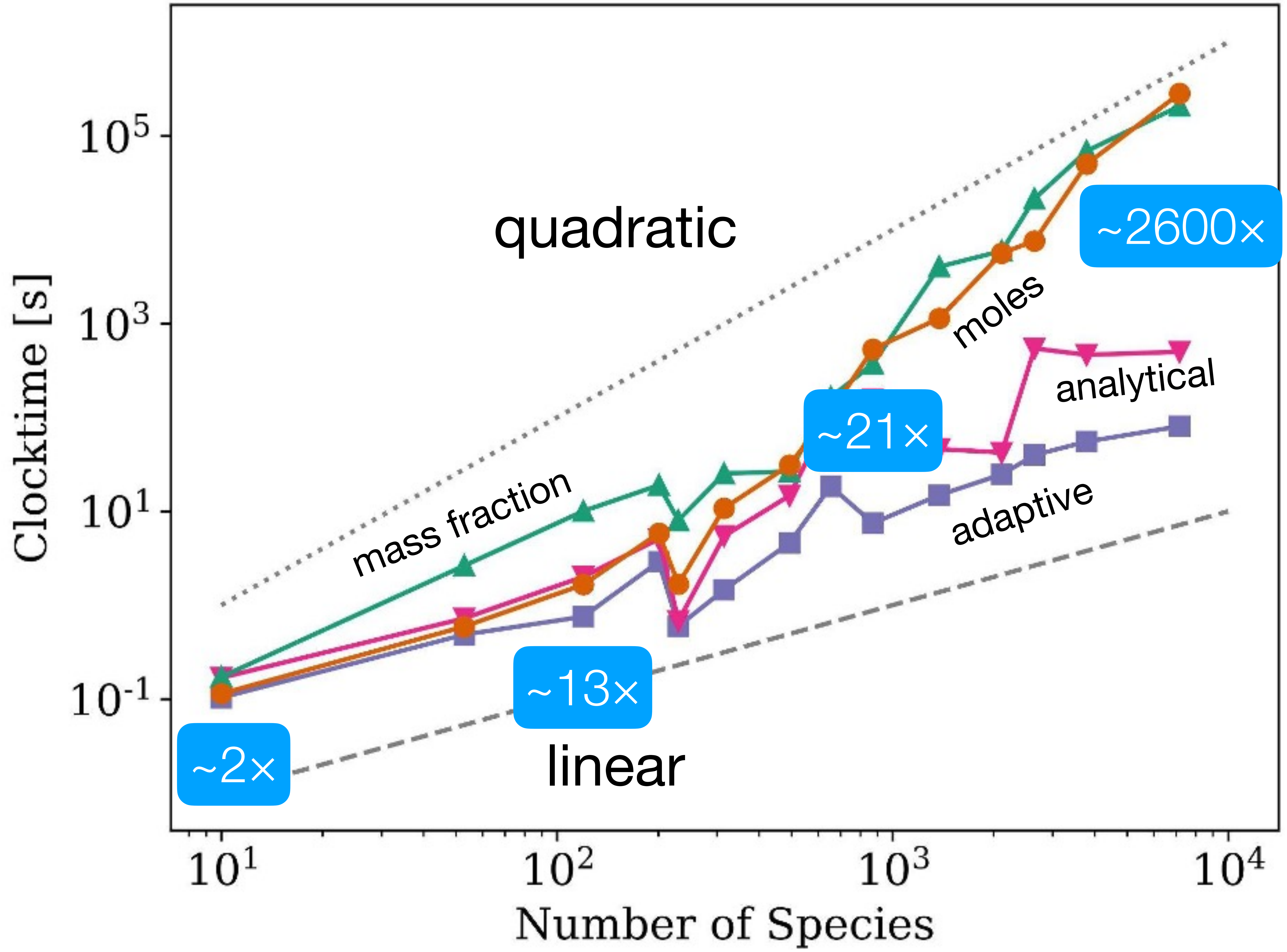
Testing

- Considered 14 kinetic models
- Considered fully analytical Jacobian, no threshold, and threshold from 10^{-18} to 10^{-1}
- Applied to both constant-volume and constant-pressure homogeneous ignition

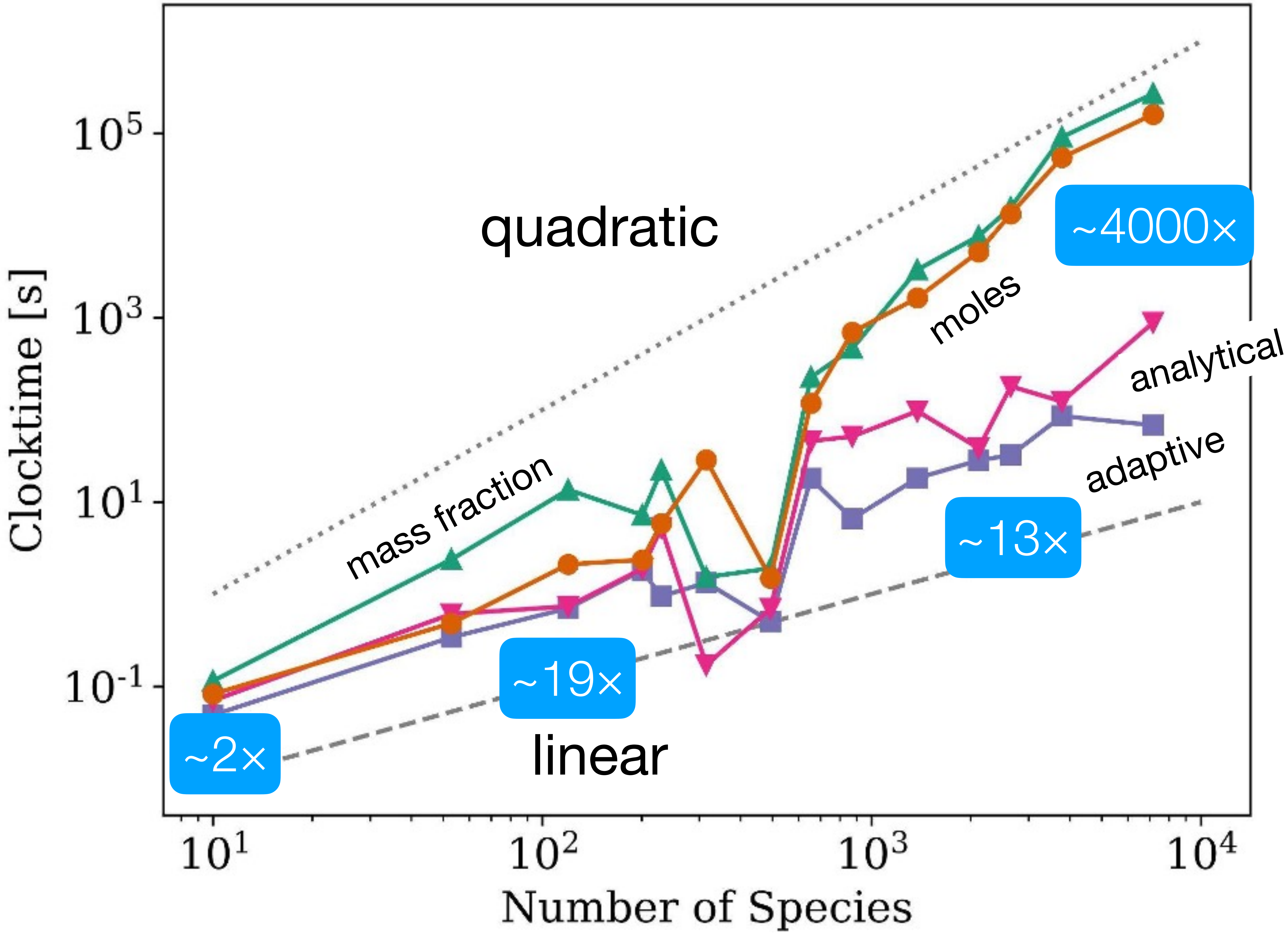


Model	Formula	Species	Reactions
Hydrogen [22]	H_2	10	29
GRI-Mech 3.0 [22]	CH_4	55	325
DME-Propane [23]	CH_3OCH_3 & C_3H_8	122	711
HyChem Jet-A [24, 25]	POSF 10325($C_{11}H_{22}$)	203	1589
Butane [26]	C_4H_{10}	230	2461
n-Heptane [27]	n-C_7H_{16}	654	4846
Isooctane [28]	i-C_8H_{18}	874	6864
3-Methylheptane [28]	$C_8H_{18}-3$	1378	8143
n-Hexadecane [29]	n-$c_{16}h_{34}$	2115	13341
Methyl-5-decenoate [30]	MD_5D	2649	10487
Methyl-decanoate & *n*-heptane [30]	MD & n-C_7H_{16}	3787	10264
2-Methyl-nonadecane [31]	$C_{20}H_{42}-2$	7171	38324

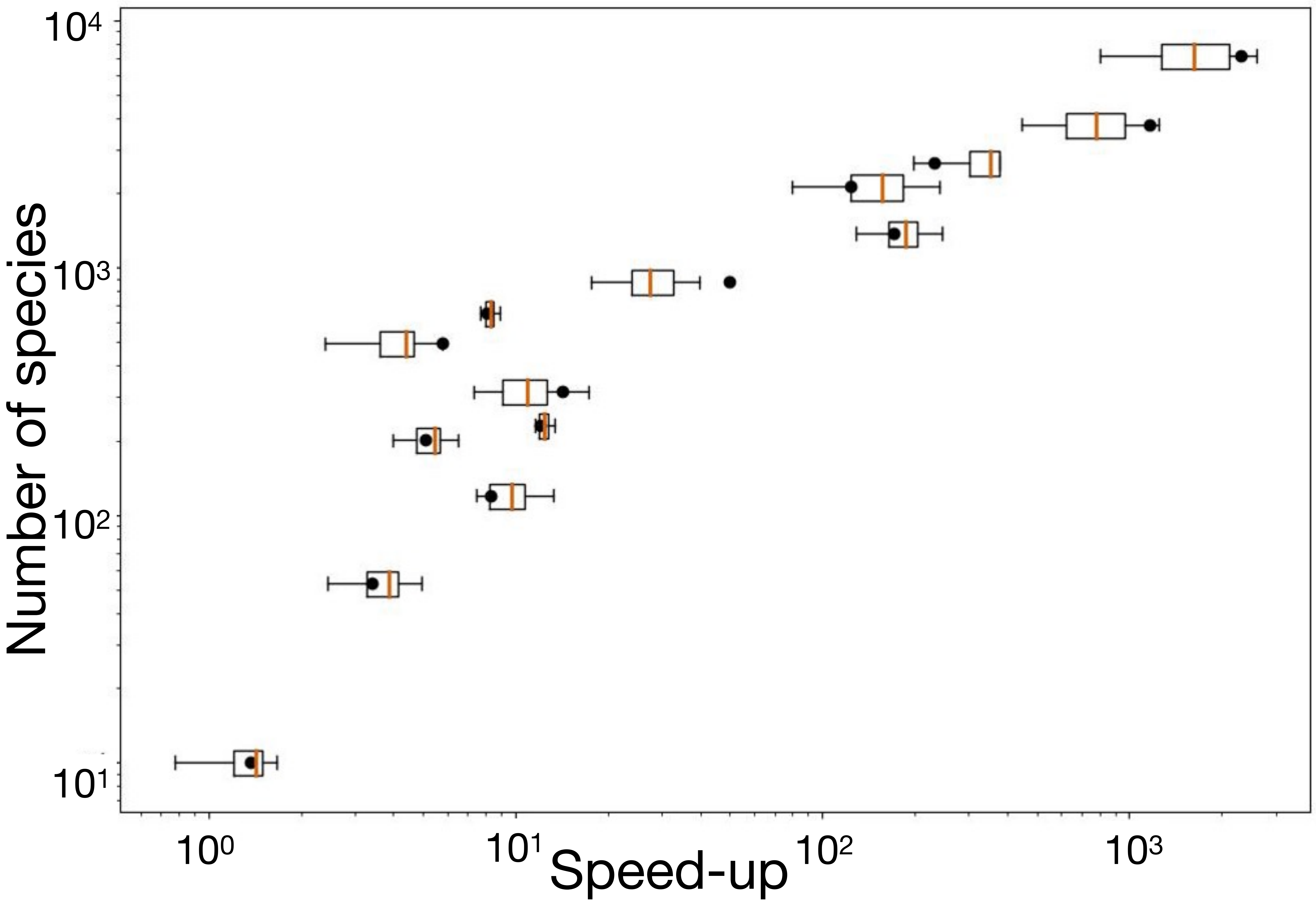
Constant pressure ignition



Constant volume ignition



Impact of threshold



Summary and current work

Cantera

- Adaptive preconditioning with a mole-based state vector and sparse linear algebra can be applied to constant-pressure and constant-volume systems and significantly improves performance
- Applying a threshold does not have a major impact on performance and we obtain substantial speedup without using a threshold
- Performance is improved mainly by reducing the number of nonlinear iterations, as well as replacing dense linear algebra with sparse.
- This is implemented in Cantera—and can be used now
- Currently extending to problems with multiple coupled reactors and surface chemistry (see NCM talk!)

Cantera: open-source community software

Raymond Speth

Richard West

Steven DeCaluwe

Bob Kee

Franklin Goldsmith

NSF

Bryan Weber

Greg Jackson

Xinyu Zhao

NUMFOCUS
OPEN CODE = BETTER SCIENCE

Cantera: open-source community software

Gandhali Kogekar

China Hagström

Anthony Walker

Jongyoon Bae

Chao Xu

Daniel Korff

Sun Su

Summary of recent work

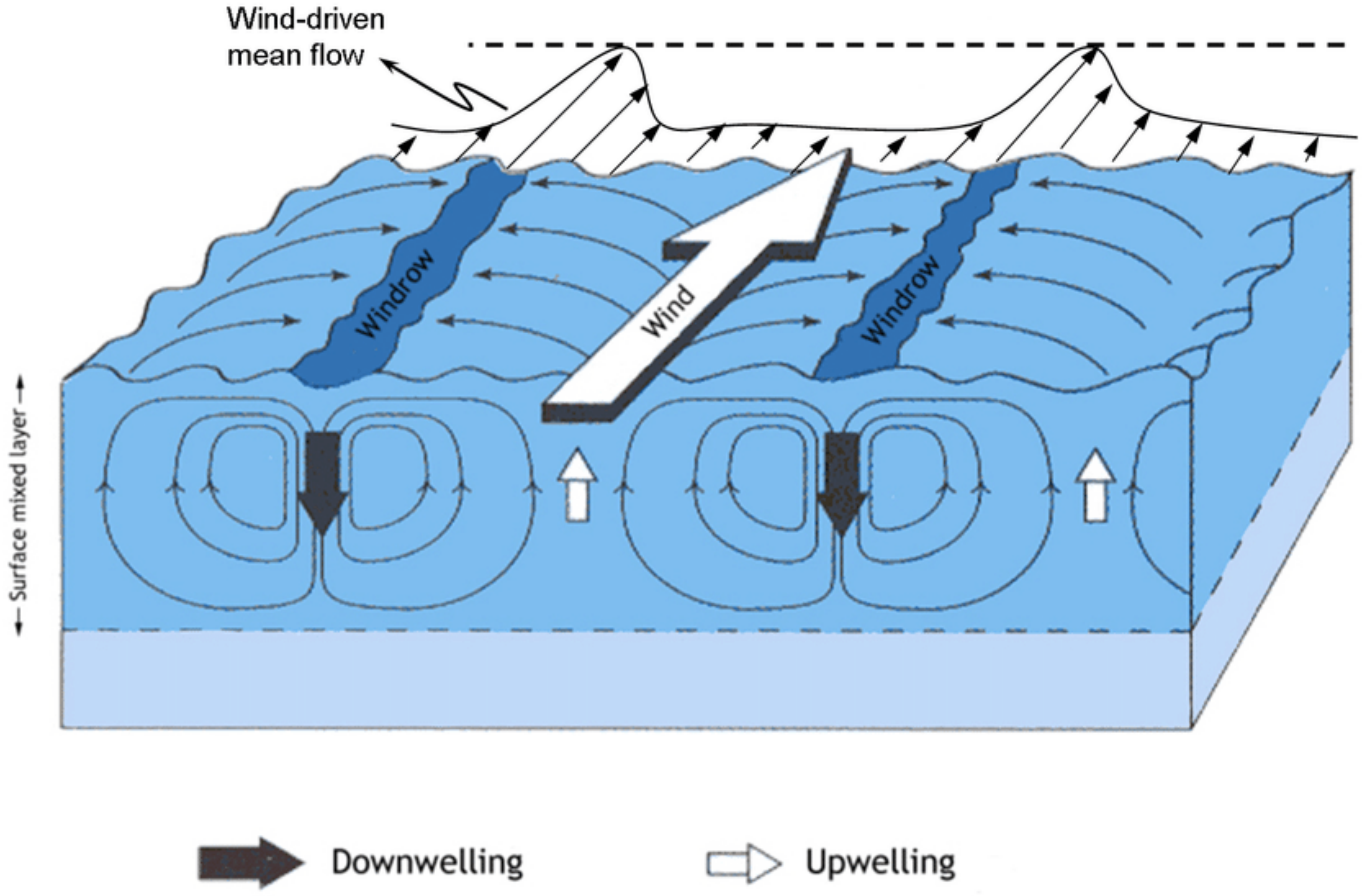
Cantera

- Cantera version 2.6.0 released in May 2022
    - Contributions from 23 developers
    - Over 1400 commits, 162 pull requests, 101 issues closed
- Key new features:
    - Easy installation via pip (as well as conda)
    - YAML-based input format
    - Extensible reactor classes
    - Refactored kinetics classes
    - New thermodynamics, kinetics classes (Peng–Robinson, Blowers–Masel surface kinetics)
    - MATLAB toolbox revamp

Extending to other fields: ocean biogeochemistry & neutron transport

Ocean turbulence–chemistry

Coupled with overlapping time scales



Langmuir turbulence

$$\begin{aligned}
CO_2 + H_2O &\rightleftharpoons HCO_3{}^- + H^+ \\
CO_2 + OH^- &\rightleftharpoons HCO_3{}^- \\
CO_3{}^{2-} + H^+ &\rightleftharpoons HCO_3{}^- \\
HCO_3{}^- + OH^- &\rightleftharpoons CO_3{}^{2-} + H_2O \\
H_2O &\rightleftharpoons H^+ + OH^- \\
B(OH)_3 + OH^- &\rightleftharpoons B(OH)_4{}^- \\
CO_3{}^{2-} + B(OH)_3 + H_2O &\rightleftharpoons B(OH)_4{}^- + HCO_3{}^-
\end{aligned}$$

Carbonate chemistry

Cigdem Akan https://digitalcommons.usf.edu/etd/3944/

Adapting methods from combustion

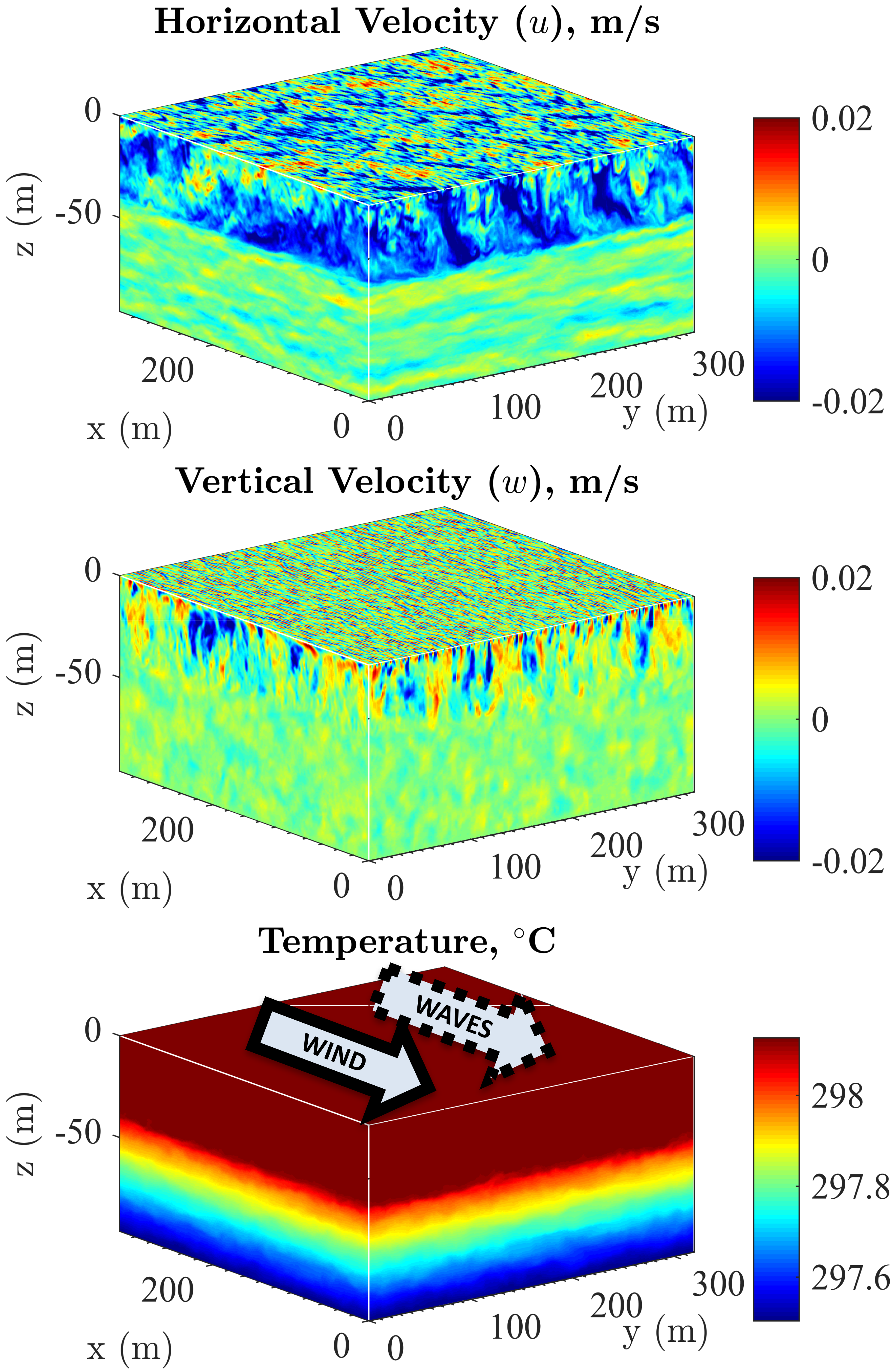
- Carbonate kinetic system is stiff, and requires prohibitively small time step sizes with explicit time-integration scheme

- Applied computational singular perturbation, identifying H^+ and OH^- as potential quasi-steady state (QSS) species

- Made QSS approximation for H^+, reducing stiffness

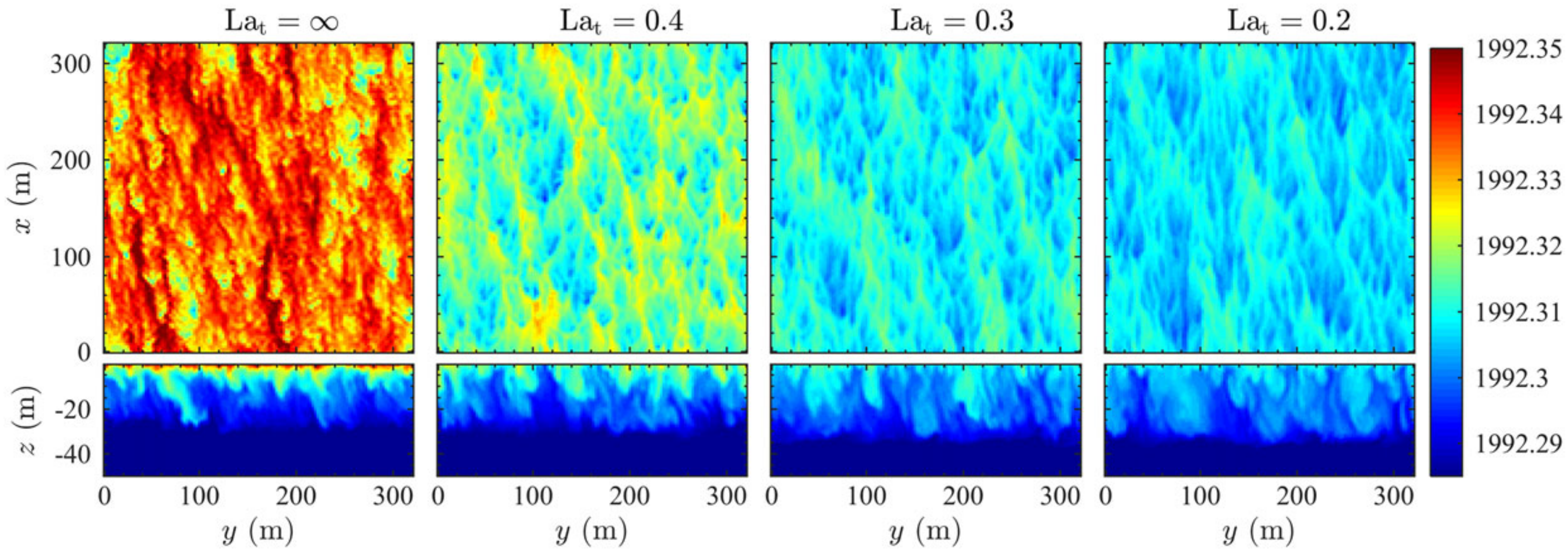
- Also Implemented third-order Runge–Kutta–Chebyshev time integration scheme

- Large-eddy simulations using NCAR LES of upper open ocean



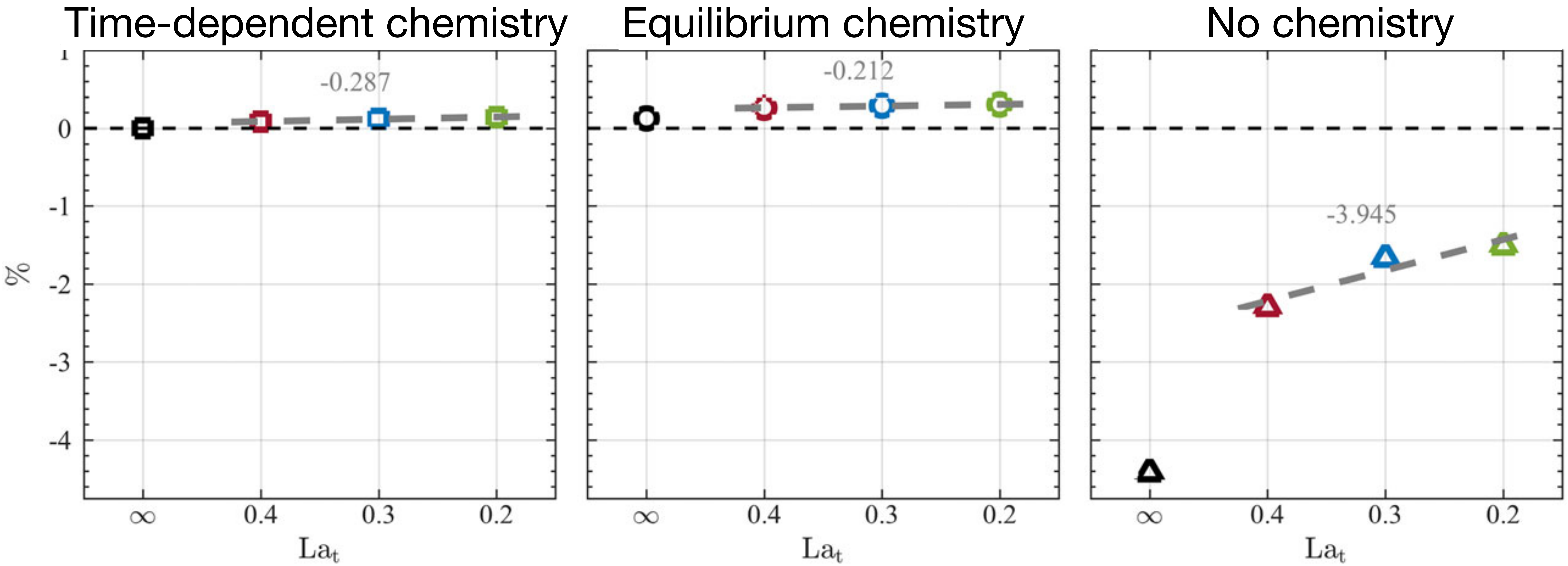
KM Smith, PE Hamlington, KE Niemeyer, B Fox-Kemper, and NS Lovenduski. 2018. *Journal of Advances in Modeling Earth Systems*, 10:3030–3048. https://doi.org/10.1029/2018MS001486

Effect of Langmuir turbulence on dissolved carbon



Increasing Langmuir turbulence

Combined effect of turbulence and chemical model fidelity



Time-dependent chemistry

Equilibrium chemistry

No chemistry

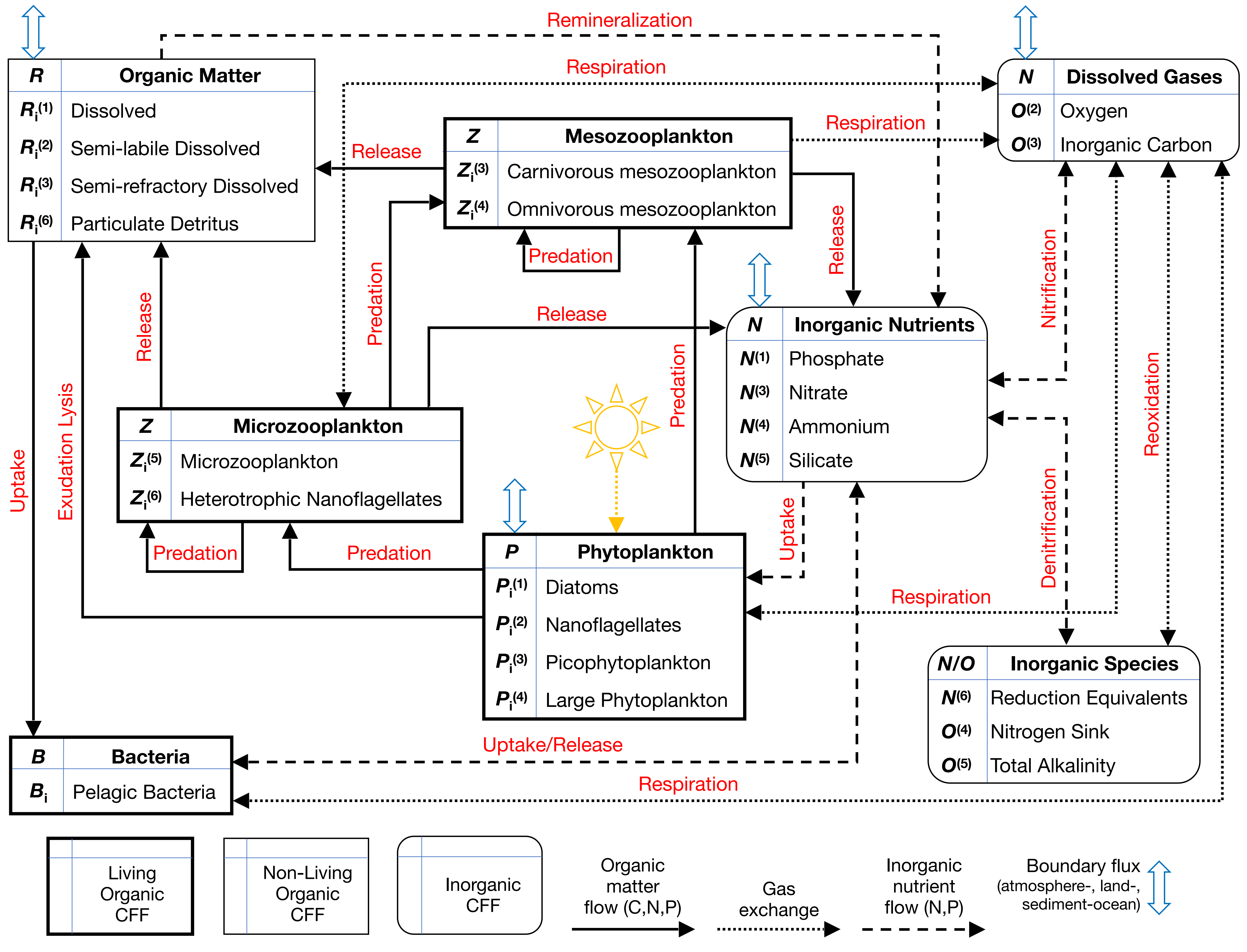
Percent change in volume-integrated total dissolved carbon compared to non-Langmuir, equilibrium chemistry

Biogeochemistry

- Want to similarly examine interaction between Langmuir turbulence and biogeochemical tracers—crucial to understanding role of ocean in global carbon cycle

- Focusing on 56-component Biogeochemical Flux Model (BFM), can capture complex ecosystem dynamics

- Too large to use in turbulence-resolving LES, so we are now extending the model reduction methods from combustion (DRGEP)



R Organic Matter
$R_i^{(1)}$ Dissolved
$R_i^{(2)}$ Semi-labile Dissolved
$R_i^{(3)}$ Semi-refractory Dissolved
$R_i^{(6)}$ Particulate Detritus

Z Mesozooplankton
$Z_i^{(3)}$ Carnivorous mesozooplankton
$Z_i^{(4)}$ Omnivorous mesozooplankton

N Dissolved Gases
$O^{(2)}$ Oxygen
$O^{(3)}$ Inorganic Carbon

N Inorganic Nutrients
$N^{(1)}$ Phosphate
$N^{(3)}$ Nitrate
$N^{(4)}$ Ammonium
$N^{(5)}$ Silicate

Z Microzooplankton
$Z_i^{(5)}$ Microzooplankton
$Z_i^{(6)}$ Heterotrophic Nanoflagellates

P Phytoplankton
$P_i^{(1)}$ Diatoms
$P_i^{(2)}$ Nanoflagellates
$P_i^{(3)}$ Picophytoplankton
$P_i^{(4)}$ Large Phytoplankton

N/O Inorganic Species
$N^{(6)}$ Reduction Equivalents
$O^{(4)}$ Nitrogen Sink
$O^{(5)}$ Total Alkalinity

B Bacteria
B_i Pelagic Bacteria

Living Organic CFF
Non-Living Organic CFF
Inorganic CFF
Organic matter flow (C,N,P)
Gas exchange
Inorganic nutrient flow (N,P)
Boundary flux (atmosphere-, land-, sediment-ocean)

Summary: ocean biogeochemistry

- We are successfully applying methods for model reduction from combustion to ocean biogeochemistry

- So far, this has let us show how finite-rate carbonate chemistry needs to be considered for accurate calculations of carbon uptake in the ocean, due to the interactions between turbulence and chemistry

- We are extending this to more-complex scenarios that capture ocean ecosystem dynamics in the upper ocean

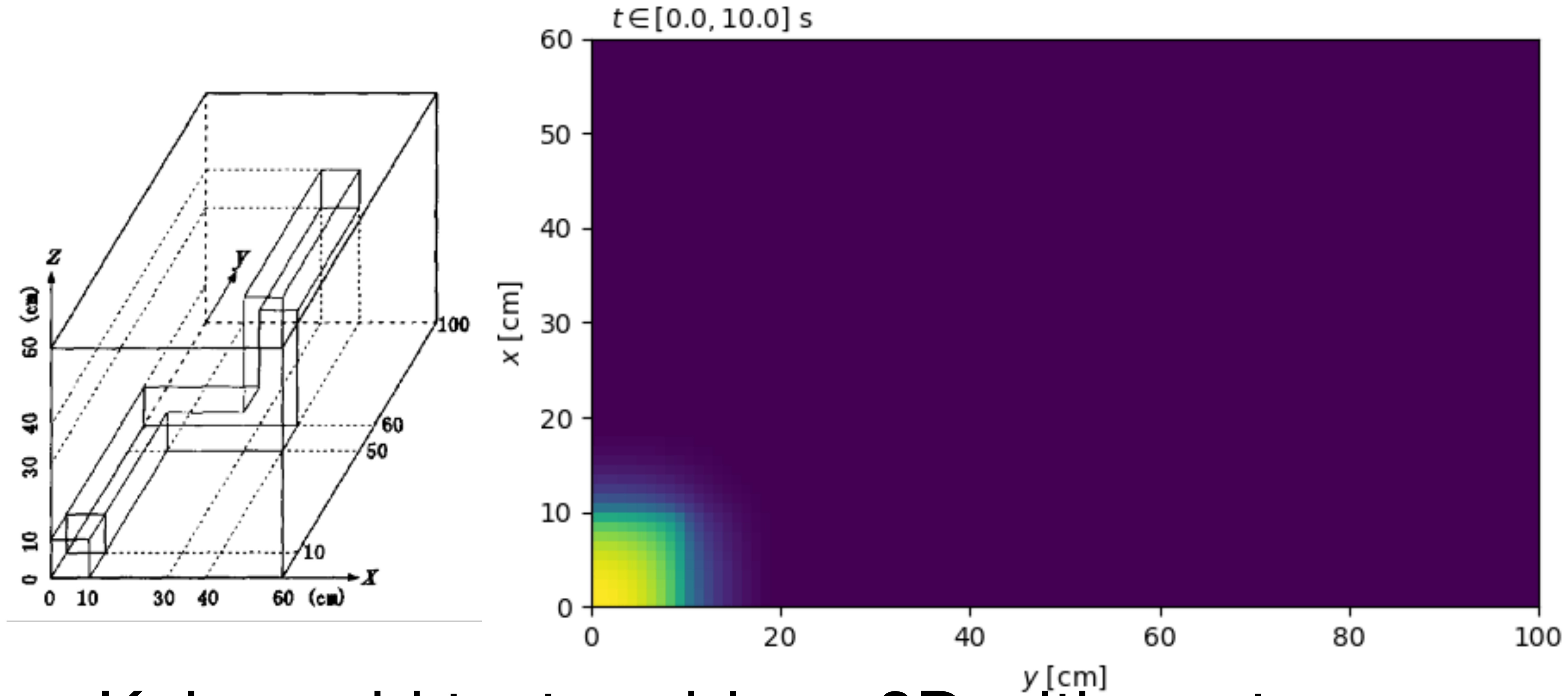
High-performance neutron transport code

Monte Carlo / Dynamic Code (MCDC)

MC DC

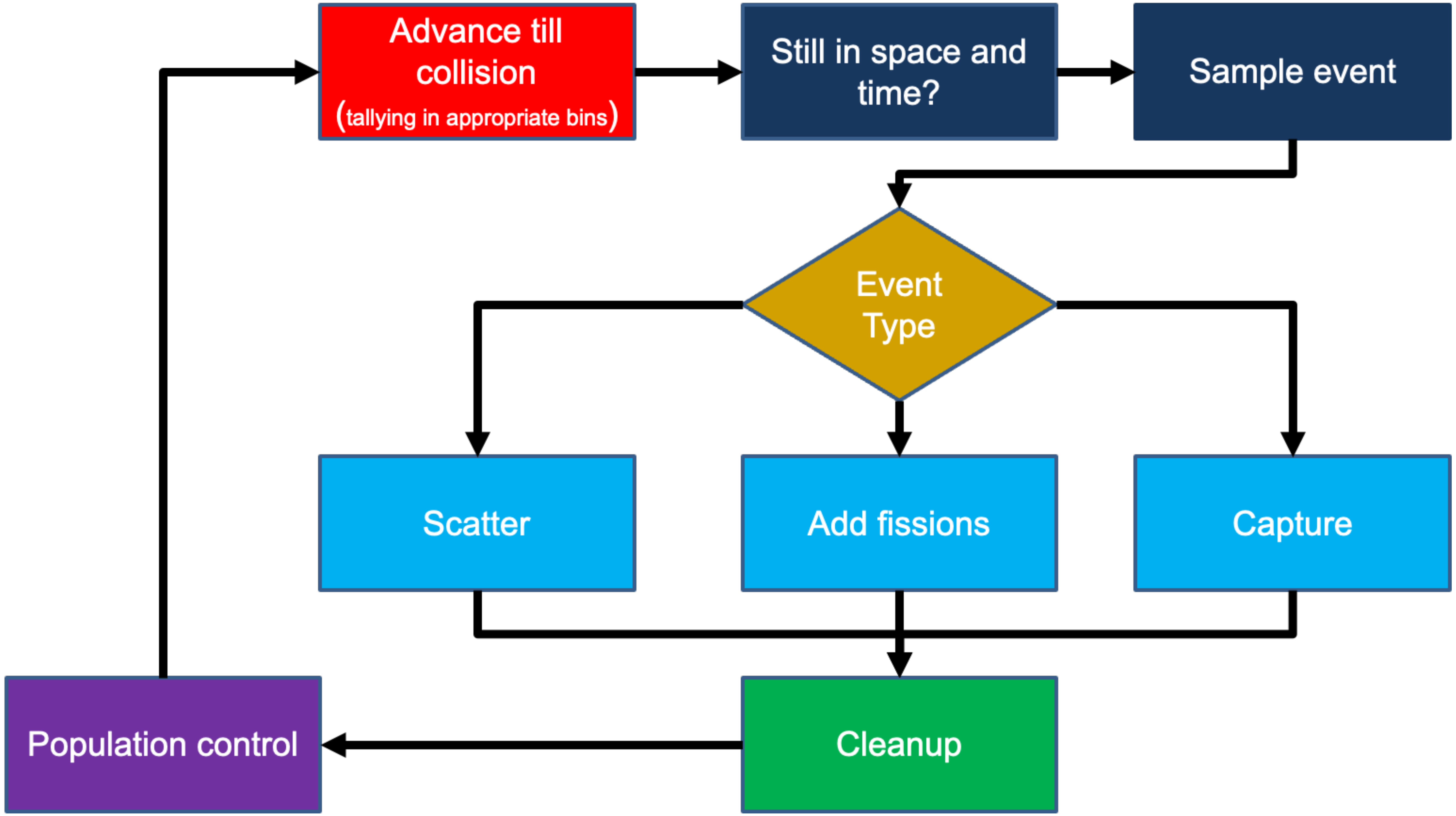
Monte Carlo Dynamic Code

- Design and analysis of nuclear reactors requires modeling the transport of neutrons, to describe where and how they trigger fission
- Governed by complex intergro-partial differential Boltzmann-type equation with seven independent variables
- Monte Carlo methods commonly used to solve particle transport, but have high computational cost for necessary sampling size
- We are developing a Python-based Monte Carlo code for method and algorithm research, but need portable performance at **large** scales



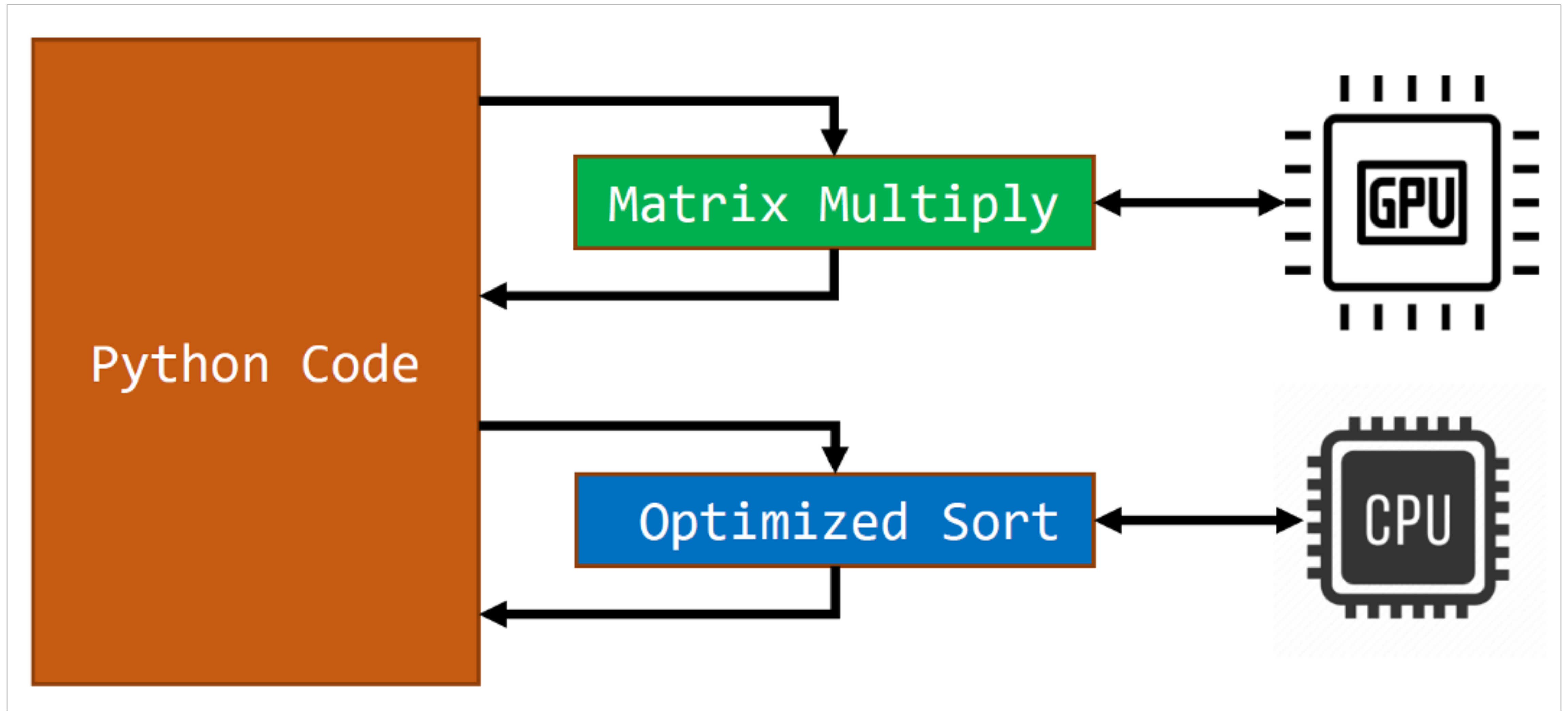
Kobayashi test problem, 3D with neutron source in absorber with “dog-leg” void

Monte Carlo neutron transport



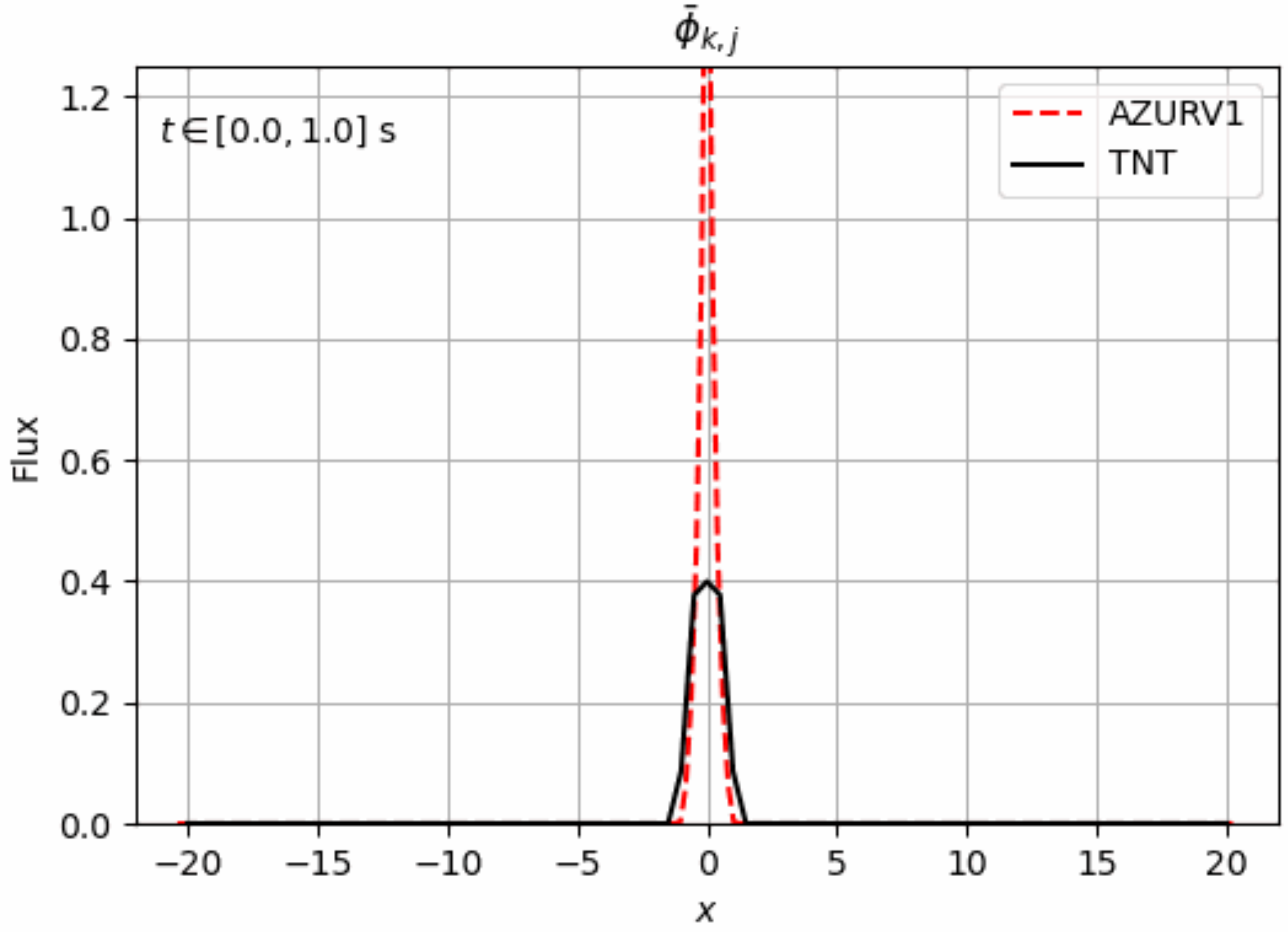
MC/DC parallelization strategy

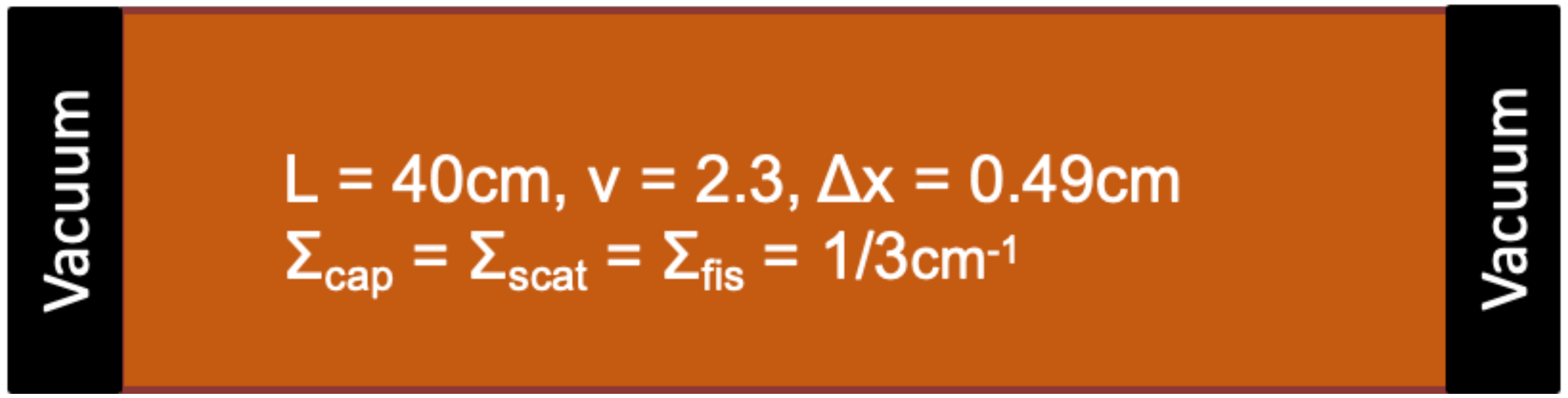
- Inspired by work in computational fluids and combustion, developing a Python-based code that uses other framework for generating parallel compute kernels
- These allow “easy” writing of new code in Python, relying on code-generation libraries to handle creating performant kernels
- Considering Numba, PyKokkos, and Mako templating engine
- These use just-in-time (JIT) compilation at run time to create machine-specific code



MC/DC Toy Neutronics Testbed

- Created a pared-down version that contains core compute functions of MD/DC
- Test problem: sub-critical slab with initial population of 10^8 particles
- Verified with analytic solution
- Simulate particle transport until death





L = 40cm, v = 2.3, Δx = 0.49cm
$\Sigma_{cap} = \Sigma_{scat} = \Sigma_{fis} = 1/3\text{cm}^{-1}$

Performance on CPU and GPU

Single CPU results:

Method of Implementation	Compile Time [s]	Run Time [s]
Pure Python	–	52970
Numba (Native threading)	5.28	232.3
Numba (PyOMP)	5.66	382.3
PyKokkos (OpenMP)	37.50	158.4

Single GPU results:

Method of Implementation	Compile Time [s]	Run Time [s]
Numba	6.25	179.36
PyKokkos	39.72	385.24
PyCUDA	2.45	160.53

Takeaways and ongoing work

- We can abstract the work of generating performant, portable code away and write in Python
- Numba is our method of choice moving forward, due to performance and ease of use
- Also researching usability with researchers familiar with neutron Monte Carlo methods but not writing high-performance massively parallel software
- Implementing this approach into the full MC/DC software

CEMENT

https://cement-psaap.github.io/

For the graduate students in the room…

“Alternate” career paths

What comes after grad school… other than academic research?

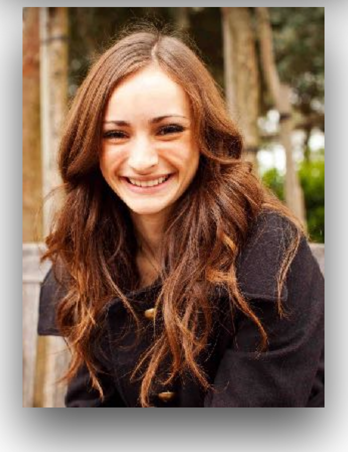
Where are they now?

Jayani Jayasuriya, PhD
Instructor @ Oregon State University

AJ Fillo, PhD
Additive Development Responsible Engineer @ Relativity Space

Paige Lorson, MS
Engineer @ EOG Resources

Morgan Mayer, MS
Intern @ Infinium

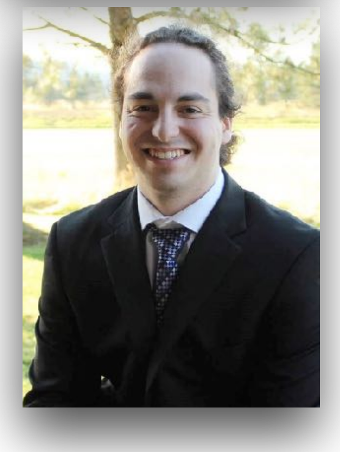

Emily Klee, MS
Aerospace Engineer @ NASA

Khang Tran, MS
Lithography Process Engineer @ Intel

Andrew Alferman, MS
Engr. Supervisor @ ASC Engineered Solns.

Matt Zaiger, MS
Metrology Engineer @ Rapid Machining Solns.

Luz Pacheco, MS
Mechanical Engineer @ Mercury Systems

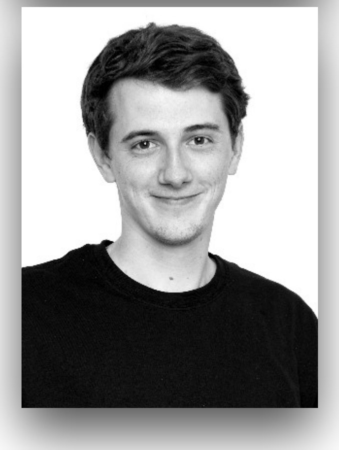

Phillip Mestas, MS
Software Engineer @ Google

Shane Daly, PhD
Senior Computational Scientist @ Enel X

Dan Magee, MS
HPC Engineer @ LANL

Tejas Mulky, MS
Thermal Engineer @ Cisco

Miguel Soler, MS
Product Engineer @ Sierra Olympic


Chris Minar, MS
Software Engineer @ Argo AI

Brittany Blankma-Stark
Aerospace Engineer @ Sierra Space

Cailin Moore
Project Engineer @ Vim Pacific

Maria Politi
PhD Student @ Univ. Washington

Parker Clayton
Sr. Software Engineer @ Tesla

What else is out there?

- If working on NSF-funded project: Non-Academic Research Internships for Graduate Students (INTERN) supplemental funding
- ORISE Internships, Fellowships, Postdocs: laboratory internships for current students; research and policy placements for recent graduates (BS, MS, PhD)
- AAAS Science & Technology Policy Fellowship: one/two-year placement in Executive, Legislative, or Judicial offices working on policy (PhD)
- Professional society Congressional Fellowships (ASME, etc.): one-year placement in Congressional offices working on policy (MS or PhD)
- ARPA-E Fellows: two-year position working in program creation, agency strategy, and outreach (PhD)

With an MS or PhD in Mechanical Engineering, you have options!

Focus on growing your skill set.

?

Thank you!
Questions?

kyle.niemeyer@oregonstate.edu

@kyleniemeyer

`niemeyer-research-group.github.io`