Elements: Open-source tools for block copolymer phase behavior

Co-PIs: Kevin Dorfman and David Morse, University of Minnesota (NSF CSSI 2103627)

Abstract

Goals: PSCF is a package for field theoretic classical statistical mechanical simulations of inhomogeneous structures formed by block polymer materials. The completed package will provide:

- Self-consistent field theory (SCFT)
- Stochastic field theoretic simulation (FTS)

History: The current C++/CUDA package evolved from an older Fortran SCFT program of the same name. The name PSCF stands for "Polymer Self-Consistent Field".

Repository: https://github.com/dmorse/pscfpp

Block Polymer Ordered Phases

- Block polymer materials exhibit a wide variety of spatially periodic equilibrium structures, driven by the immiscibility of different blocks.
- Characteristic structural lengths are of order 10 - 100 nm, and are comparable to the polymer nandom-walk coil sizes.
- Emergence of order at mesoscopic length scales allows the use of coarse-grained models in which polymers are idealized as random walks with short range interactions.

Diblock Copolymer Ordered Phases

BCC Spheres Hexagonal Cylinders Gyroid Network Lamellar

Frank-Kasper Sphere Phases (SCFT):

A15 σ phase

Mai and Eisenberg, Chem Soc. Rev. **41**, 5969 (2012)

Arora et al., Macromolecules **49**, 4675 (2012)

Single-Chain Problem

- Field theoretic methods replace inter-particle interactions by interactions of monomers with chemical potential-like fields.
- A field $w_i(\mathbf{r})$ represents an excess free energy cost of inserting a monomer of type i at position $\mathbf{r}$.
- The field configuration may be stationary, in SCFT, or may fluctuate, in stochastic FTS methods.
- Schematic of fields for A and B monomers in an SCFT model of an A/B diblock copolymer lamellar phase :

w_A w_B

- Field theoretic methods require computation of a function $q(\mathbf{r}, s)$, which is a normalized partition function of a biased random walk segment of chain length s (s monomers) with one end constrained to position $\mathbf{r}$.

View Online

$\mathbf{r}$

- The function $q(\mathbf{r}, s)$ satisfies a modified diffusion equation (MDE) in which chain length s plays a role analogous to time:

$$\frac{\partial q(\mathbf{r}, s)}{\partial s} = -\left[-\frac{b^2}{6}\nabla^2 + w(\mathbf{r}) \right] q(\mathbf{r}, s)$$

- Solution of the MDE for blocks of copolymers dominates the computational cost of all field theoretic methods.
- The PSCF MDE solver for periodic systems uses an operator splitting method in which the Laplacian is treated in Fourier space, using frequent FFTs.
- Fast Fourier transforms (FFTs) dominate computation time.

Self-Consistent Field Theory

- SCFT is a classical mechanical density for liquids containing flexible (random-walk) polymers.
- Free energy functional:

$$F[c] = F_{\text{ideal}}[c] + F_{\text{ex}}[c]$$

ideal gas of polymers   monomer-monomer interactions

$$c = [c_A(\mathbf{r}), c_B(\mathbf{r}), \ldots]$$ Mean monomer concentration fields

- $F_{\text{ex}}[c]$ is an explicit, empirically determined, usually local functional, with parameters fit to experiment.
- Free energy minimization yields a self-consistency condition:

$$w_i(\mathbf{r}) \equiv \frac{\delta F_{\text{ex}}[c]}{\delta c_i(\mathbf{r})} = -\frac{\delta F_{\text{ideal}}[c]}{\delta c_i(\mathbf{r})}$$

- Program structure:

  - Inner problem: Solve MDE for specified w field field configuration, then use solution to compute monomer concentrations.

  - Outer loop : Iterate w fields to satisfy the self-consistency (minimization) condition

Field Theoretic Simulation

- Field theoretic simulation (FTS) methods involve repeated solution of the MDE in a stochastically fluctuating field configuration.
- The completed package will implement two FTS methods:

  - Field Theoretic Monte-Carlo (FTMC)

  - Field-Theoretic Complex Langevin (FTCL)

- The functional integral sampled by the FTCL method can be derived by an exact Hubbard-Stratonovich transformation of an equivalent particle-based model.
- FTCL requires sampling of *complex-valued* fields, & is prone to sign problems. FTMC uses real fields.
- FTMC relies on a mathematical approximation to FTCL that reduces computational costs, at modest cost in reduced accuracy.

Software Package Design

- PSCF is written in idiomatic object-oriented C++
- A web manual with full user- and developer-level documentation is created with doxygen by combining source code docs with manual pages
- PSCF was designed to facilitate development and maintenance of several programs that have analogous top-level structure, but are designed for use with different boundary conditions or symmetry, or run on different hardware (CPU vs. GPU)
- C++ templates are used heavily to minimize code duplication among analogous programs that use different data types, e.g., to represent fields.
- The completed package will contain the following programs:

  - Finite difference CPU SCFT code for 1D planar, spherical and cylindrical geometries (e.g. micelles and interfaces).

  - Pseudo-spectral CPU code for SCFT with periodic boundary conditions (ordered phases).

  - Pseudo-spectral GPU code for SCFT & FTMC with periodic ordered phases. (Note: These methods use the same field representations and MDE solvers)

  - Pseudo-spectral GPU code for FTCL sampling with periodic structures (complex fields)

- SCFT tools are mature and in use, while both FTS tools are still in development

User Features

- Arbitrary mixtures of polymers and solvent
- Acyclic branched polymers of arbitrary topology
- Canonical, grand canonical and mixed ensembles
- Ordered phases with 1D, 2D, or 3D periodicity
- Arbitrary user-specified crystal system and space group symmetry in SCFT (e.g., any of 230 3D space groups)
- Automatic optimization of SCFT unit cell parameters
- Efficient continuation of SCFT solutions along lines through parameter space, for phase diagram studies
- Efficient GPU SCFT parallelization (30x speedup)
- Simple 1D program to simulate spherical and wormlike micelles.
- Visualization and data analysis tools included
- Thorough user and source-code documentation