Quantitative comparison of field update algorithms for polymer SCFT and FTS

Daniel L. Vigil,[†] Kris T. Delaney,[‡] and Glenn H. Fredrickson[*,†,‡]

†*Department of Chemical Engineering, University of California, Santa Barbara, California 93106, United States*

‡*Materials Research Laboratory, University of California, Santa Barbara, California 93106, United States*

E-mail: ghf@mrl.ucsb.edu

Abstract

A wide range of field update algorithms for polymer self-consistent field theory (SCFT) and field theoretic simulations (FTS) are analyzed. We provide the first direct comparison between Anderson mixing (AM) and fictitious relaxational dynamics for SCFT and find nearly equivalent performance when both schemes are properly tuned. We also show that predictor-corrector algorithms are the most efficient among fictitious dynamics approaches despite increased costs per step. For FTS adaptive time stepping is found to dramatically improve algorithm stability for inhomogeneous systems and enable simulation at much lower chain length and density than was previously achievable.

Introduction

Field theory has been a prominent tool in polymer physics for the last fifty years. Pioneering work by Edwards, de Gennes, Leibler and many others used analytical approximations to

understand critical phenomena,[1] phase separation,[2] and more.[3,4] In the last thirty years numerical treatment of field theories has become possible, which has allowed the relaxation of approximations such as strong or weak segregation used in earlier work. Numerical solutions of self-consistent field theory (SCFT), which is a mean-field approximation to the full field theory, are now routine and have enabled simulation of full phase diagrams for broad classes of block copolymers and polymer blends.[5–8] It is also possible to conduct simulations of the exact field theory without the mean-field approximation in so-called "Field Theoretic Simulations" (FTS).[7,9,10] FTS has enabled study of phenomena not accessible to SCFT, such as fluctuation-corrected phase diagrams in neat and salt-doped diblock melts,[11–13] polyelectrolyte complexation,[14–16] nematic ordering,[17] ternary microemulsions,[18–20] and novel "bricks & mortar" emulsion phases.[21]

There are multiple equivalent formalisms that can be used to construct a field theory for an assembly of polymers, but the most mature and numerically tractable is the so-called Auxiliary Field (AF) framework, which decouples pairwise non-bonded interactions between chain segments via a set of fields. There are then two primary tasks in evaluating an AF theory: 1) computing the single-molecule partition functions with fixed field configurations; and 2) generating new realizations of the fields. For polymer models the first task is non-trivial due to the correlations between different segments on the backbone. Previous authors have written about how to evaluate single-polymer partition functions accurately and efficiently, and we refer readers to those works.[22–26] In this work we focus on the task of generating new iterations of a field. In SCFT the goal is to identify saddle-point field configurations that represent the most-probable state at equilibrium. Mathematically, the saddle-point nature of this configuration can be stated as:

$$\left.\frac{\delta H[w]}{\delta w(\boldsymbol{r})}\right|_{w=w^*} = 0 \tag{1}$$

Here H is the effective Hamiltonian of the field theory, w is an auxiliary field, and w^* is the

saddle-point configuration. The task of generating new fields then reduces to searching for the saddle-point most quickly from a given initial guess. There are multiple approaches to solving the SCFT saddle-point equations (eq 1). The first treats the task as a non-linear root-finding problem and historically used quasi-Newton approaches to find the saddle-point.[27] More recently the task has been posed as a fixed-point problem, which has led to Anderson mixing (AM) and other "Jacobian-free" approaches that require significantly less memory than quasi-Newton type methods.[28–31]

An alternative approach involves a fictitious dynamics with the saddle-point configuration as a fixed point. The dynamical system is then evolved in fictitious time until the fixed point is reached. Mathematically this can be expressed as

$$\frac{\partial w(\boldsymbol{r}, t)}{\partial t} = -\frac{\delta H[w]}{\delta w(\boldsymbol{r})} \tag{2}$$

where H and $w(\boldsymbol{r})$ are generally complex, corresponding to a gradient descent towards saddle points in the complex plane. This approach allows the use of a wide variety of algorithms developed for solving differential equations, a number of which are described in the Methods section. Our first goal in this paper is to compare the various algorithms available for SCFT to determine which algorithms perform best and under what conditions.

There is a significant amount of work in the literature comparing numerical methods for SCFT, but these largely focus on algorithms for computing single-chain partition functions and chain propagators. In particular, many debate the relative merits of spectral and pseudospectral approximations.[25,26,29,30,32] With few exceptions,[33] the polymer SCFT community has largely settled on using pseudospectral methods, which we adopt in this work. There are few direct comparisons of different field-update methods, and those that do only consider a small subset of algorithms.[28,34] In this work we consider seven different algorithms for conducting SCFT field updates and provide the first direct comparison of the AM algorithm with fictitious dynamics algorithms. Additionally, we demonstrate how to calibrate each

algorithm to obtain optimal performance and provide some heuristics for choosing numerical parameters.

For a fully fluctuating FTS, the goal of generating new field iterations is quite different. Instead of searching for saddle-point field configurations, the goal is to generate a sequence of decorrelated field configurations that allow for importance sampling of average field operators that describe physical properties of the system. Examples of important operators are the chemical potential and pressure. Because field theories are usually complex-valued, the averaging process can suffer from a "sign-problem" where a phase present in the complex statistical weight that is extensive with system size produces wild oscillations and leads to difficult averaging when attempting to use conventional methods such as Monte Carlo sampling.

One approach to overcome the sign-problem is a partial-saddle-point approximation that renders the Hamiltonian purely real, enabling traditional Monte Carlo and real Langevin simulations.[18,35–38] This approach has been successful for studying fluctuations in AB-type polymer systems, including microemulsions of diblock-homopolymer ternary blends and shifts in the order-disorder transition.[13,18–20] It is not clear how to extend the approach to multicomponent and multispecies systems, however.

A more general approach to overcome the sign-problem is the complex Langevin (CL) method, which has been used in single-component systems, binary, ternary and quarternary mixtures.[17,39] Because of its broader applicability, we focus solely on the CL method in this work. For CL simulations, the sampling scheme follows the dynamical equation

$$\frac{\partial w(\boldsymbol{r},t)}{\partial t} = -\frac{\delta H[w]}{\delta w(\boldsymbol{r})} + \eta(\boldsymbol{r},t) \tag{3}$$

where $\eta(\boldsymbol{r},t)$ is a real-valued Brownian random force. This equation is nearly identical to eq 2, only differing by the addition of Brownian noise. However, eq 3 is not a conventional "real" Langevin dynamics since $H[w]$ is complex, leading to field trajectories $w(\boldsymbol{r},t)$ that

are not restricted to real values. Nonetheless, the similarities between eqns 2 and 3 imply that SCFT algorithms relying on fictitious-time relaxation can be readily adapted to CL. The complex Langevin approach also permits use of various algorithms from the stochastic differential equation literature. Our second goal in this paper is to compare algorithms for CL simulations and determine which perform best in terms of stability and efficiency.

There are multiple previous works that compared different algorithms for CL simulations.[26,39–41] Unfortunately, these works were limited to either a subset of algorithms,[26,40] models with no microphase self-assembly,[41] or small parameter ranges.[39] In this work we consider eight different algorithms for CL simulations, including adaptive time steppers that have not been used previously in polymer systems. We show that these adaptive time steppers significantly improve stability in inhomogeneous systems, especially at strong segregation and low polymer densities. These advances allow CL simulations at conditions that were intractable with previous algorithms. Finally, we explain the mechanism by which algorithms become unstable in CL and how adaptive time stepping avoids failure.

Theory

We use an AB diblock copolymer melt in the canonical ensemble as a test system for this study. We treat the diblocks as continuous Gaussian chains with segment mass distributed over a Gaussian packet[42] to regularize the field theory, and include both A-B segmental interactions and a Helfand compressibility penalty in the model. The model equations for

this system are[11]

$$Z(n,V,T) = Z_0 \int \mathcal{D}w_+ \int \mathcal{D}w_- \, \exp(-H[w_+,w_-]) \tag{4}$$

$$\begin{aligned} H[w_+,w_-] = {} & \frac{C}{\chi N}\int d\boldsymbol{r}\,(w_-(\boldsymbol{r}))^2 + \frac{C}{\chi N + 2\zeta N}\int d\boldsymbol{r}\,(w_+(\boldsymbol{r}))^2 \\ & - \frac{2iC\zeta N}{\chi N + 2\zeta N}\int d\boldsymbol{r}\, w_+(\boldsymbol{r}) - C\bar{V}\ln(Q[w_+,w_-]) \end{aligned} \tag{5}$$

$$Q[w_+,w_-] = \frac{1}{\bar{V}}\int d\boldsymbol{r}\, q(\boldsymbol{r},1) \tag{6}$$

$$q(\boldsymbol{r},0) = 1 \tag{7}$$

$$\partial_s q(\boldsymbol{r},s) = \nabla^2 q(\boldsymbol{r},s) - w(\boldsymbol{r},s)q(\boldsymbol{r},s) \tag{8}$$

$$w(\boldsymbol{r},s) = \begin{cases} \Gamma * (iw_+(\boldsymbol{r}) - w_-(\boldsymbol{r})) & s \in (0,f] \\ \Gamma * (iw_+(\boldsymbol{r}) + w_-(\boldsymbol{r})) & s \in (f,1] \end{cases} \tag{9}$$

$$\Gamma * w(\boldsymbol{r}) = \frac{1}{(2\pi a^2)^{3/2}}\int d\boldsymbol{r}' \exp\left(-\frac{1}{2a^2}|\boldsymbol{r}-\boldsymbol{r}'|^2\right) w(\boldsymbol{r}') \tag{10}$$

Here χ is the Flory interaction parameter, ζ is the Helfand compressibility parameter, N is the polymer contour length, f is the volume fraction of species A, $C = nR_g^3/V$ is the dimensionless chain density, with n the number of polymers and V the volume of the box, and Z_0 is a reference partition function containing ideal-gas contributions and normalizing denominators. $Q[w_+,w_-]$ represents the partition function for a single polymer chain interacting with the fields w_+ and w_- and is computed from the propagator $q(\boldsymbol{r},s)$, which represents the field-based random walk statistics of the polymer starting from a free end. For all calculations we set $f = 0.34$ unless otherwise specified. In this model, the monomer density has been smeared with a Gaussian kernel, Γ, with a range a. All lengths have been non-dimensionalized in units of the unperturbed polymer radius of gyration $R_g = b(N/6)^{1/2}$ and the contour variable $s \in [0,1]$ has been scaled by $1/N$. All spatial integrals are over the scaled volume $\bar{V} = V/R_g^3$.

For SCFT calculations we assume that the melt is incompressible, $\zeta N \to \infty$, and that the

density is unsmeared $a = 0$. The incompressible, unsmeared version of the model displays pathological ultraviolet divergences that make the model undefined when conducting fully fluctuating CL simulations.[41,43] These pathologies can be removed by using finite values for both ζN and a in CL simulations. Unless otherwise specified, we set $\zeta N = 100$ and $a = 0.2\, R_g$ for CL calculations. All calculations are conducted in a fixed-size cubic box with periodic boundary conditions.

For SCFT calculations, the free energy is equal to the effective Hamiltonian, H, and must be purely real valued. This implies that $w_+(\boldsymbol{r})$ must be purely *imaginary* at the saddle-point, despite the fact that the functional integral is over real-valued functions. The argument of the functional integral is analytic, which allows the path of integration to be deformed off the real axis to include the purely imaginary saddle-point.[7] For fluctuating CL simulations this is accomplished automatically via the complexification of the fields. Nevertheless, for SCFT calculations it is convenient to absorb a factor of i into $w_+(\boldsymbol{r})$ and constrain the search path to purely imaginary fields to render the effective Hamiltonian purely real throughout the search space. Such a change of variables constitutes a so-called "Wick rotation".

Numerical methods

Pseudospectral numerical methods for computing the propagator, $q(\boldsymbol{r}, s)$, have been explored elsewhere.[22–26] Based on these results, for SCFT calculations we use the RQM4 algorithm with a contour step of $\Delta s = 0.01$.[24] For CL calculations we use the RK2 algorithm with $\Delta s = 0.01$.[22,23] The overall evaluation of the force $G(\boldsymbol{r}; [w]) = -\delta H / \delta w(\boldsymbol{r})$ requires evaluating both the forward and reverse propagators, and evaluating such objects has a computational cost that scales like $\mathcal{O}(N_s M \ln(M))$, where $N_s = 1/\Delta s$ is the number of contour samples along the polymer backbone, and M is the number of sample points in space. A single field has M elements and typically requires $\mathcal{O}(M)$ operations to update, so for each field configuration generated during the simulation, evaluating the force is significantly more

expensive than updating the fields and usually represents the majority of computation time in SCFT and FTS. Minimizing the number of field updates and force evaluations is thus critical to reducing computation time. All SCFT calculations were conducted in a box of size $L_x = L_y = L_z = 9\,R_g$ with $M = 64^3$ sample points, while CL calculations used the same size box with $M = 48^3$, unless otherwise specified.

For field-update schemes, we first briefly review the Anderson mixing algorithm. Anderson mixing is typically used for fixed-point-iteration type problems of the form $w(\boldsymbol{r}) = E(w(\boldsymbol{r}))$, where E is a nonlinear function. The deviation of a particular value of $w(\boldsymbol{r})$ is defined by $d(\boldsymbol{r}) = E(w(\boldsymbol{r})) - w(\boldsymbol{r})$. If one has already iterated through a number, n_h, of fields $w_i(\boldsymbol{r}), w_{i-1}(\boldsymbol{r}), \ldots, w_{i-n_h}(\boldsymbol{r})$ then one can compute an optimized guess for the next iteration, $w_{i+1}(\boldsymbol{r})$, by finding coefficients $\alpha_i, \ldots, \alpha_{i-n_h}$ that minimize $\int d\boldsymbol{r} \left(\sum_{j=i-n_h}^{i} \alpha_j d_j(\boldsymbol{r})\right)^2$ subject to the constraint $\sum_{j=i-n_h}^{i} \alpha_j = 1$. Determining these coefficients is a linear optimization problem that requires at least $\mathcal{O}(n_h^2 M)$ operations. Details on efficient implementation of the AM algorithm and initialization strategies can be found elsewhere.[28,29]

The fictitious dynamics algorithms attempt to solve the partial differential equation

$$\partial_t w(\boldsymbol{r}, t) = -\frac{\delta H}{\delta w(\boldsymbol{r})} + \eta(\boldsymbol{r}, t) = G(\boldsymbol{r}; [w]) + \eta(\boldsymbol{r}, t) \tag{11}$$

where for SCFT $\eta = 0$, and for CL η is the Brownian force. From here on we refer to the deterministic term $G(\boldsymbol{r}; [w]) = -\delta H / \delta w(\boldsymbol{r})$ as the "force" on a field, w. The simplest approximation to solve eq 11 is the Euler-Maruyama (EM1) approximation

$$w^{j+1}(\boldsymbol{r}) - w^j(\boldsymbol{r}) = \Delta t\, G(\boldsymbol{r}; [w^j]) + R^j(\boldsymbol{r}) \tag{12}$$

Here the superscript j represents a discrete-time index and R^j is a random variable with zero mean and variance $\langle R^j(\boldsymbol{r}) R^k(\boldsymbol{r}')\rangle = 2\Delta t\, \delta_{jk} \delta(\boldsymbol{r} - \boldsymbol{r}')$. We continue to use this definition for $R^j(\boldsymbol{r})$ when describing other algorithms, unless otherwise specified. Any noise distribution with these first two moments can be used, however we exclusively use normally distributed

noise in this work. Although simple to implement, the EM1 algorithm has poor stability and accuracy compared to other algorithms.

Another class of algorithms splits the force into a linear and a non-linear contribution

$$-\frac{\delta H}{\delta w(\boldsymbol{r})} = G(\boldsymbol{r};[w]) = -c * w(\boldsymbol{r}) + F(\boldsymbol{r};[w]) \tag{13}$$

where $c * w$ is a convolution that represents the linear contribution to $\delta H/\delta w(\boldsymbol{r})$ and F represents all non-linear contributions. For polymer models it is typically most convenient to express the kernel function c in Fourier space where it is diagonal and positive definite[34] and the linearized force convolution $c * w$ can be computed via simple multiplication. The linearized force $c * w$ typically is derived from a linear response analysis about the translationally invariant disordered phase of the system and thus constitutes an approximation to the true linear force that is only accurate for weak perturbations about the disordered state. The explicit kernel functions for the diblock model considered in this work are given in the Supporting Information.

After splitting the force into linear and non-linear parts, semi-implicit algorithms can be devised in order to stabilize the algorithm. One such algorithm is a first order semi-implicit scheme (SI1), which is defined by

$$w^{j+1}(\boldsymbol{r}) - w^{j}(\boldsymbol{r}) = \Delta t \left(-c * w^{j+1}(\boldsymbol{r}) + F(\boldsymbol{r};[w^{j}])\right) + R^{j}(\boldsymbol{r}) \tag{14}$$

Another similar algorithm that uses linearized force information is the first-order exponential time differencing ETD1 method,[44,45] which is derived by using c as an integrating factor over the time interval $t \to t + \Delta t$.

$$\hat{w}^{j+1}(\boldsymbol{k}) - \hat{w}^{j}(\boldsymbol{k}) = \frac{1 - e^{-\hat{c}(k)\Delta t}}{\hat{c}(k)} \hat{G}(\boldsymbol{k};[w^{j}]) + \left(\frac{1 - e^{-2\hat{c}(k)\Delta t}}{2\hat{c}(k)\Delta t}\right)^{1/2} \hat{R}^{j}(\boldsymbol{k}) \tag{15}$$

here the equation has been transformed from real space, $\boldsymbol{r}$, to Fourier space, $\boldsymbol{k}$, with hats over

symbols indicating Fourier transforms $\hat{w}(\boldsymbol{k}) = \mathcal{F}_{\boldsymbol{r}\to\boldsymbol{k}}(w(\boldsymbol{r}))$. Note that the linear response kernel $\hat{c}(k)$ is a function of the magnitude of the Fourier mode $k = |\boldsymbol{k}|$.

All of the fictitious dynamics algorithms discussed so far have first-order accuracy with respect to time step. In the stochastic case with $R \neq 0$, this is first-order accuracy in the weak sense. To achieve higher-order accuracy we employ predictor-corrector algorithms. These algorithms perform an initial "predictor" time step, then use this predicted field to more accurately evaluate the force over the time step interval for a subsequent improved corrector step. These algorithms must evaluate the force two times per iteration, but have second-order accuracy (in the weak sense) in fictitious time. The simplest of these is the Euler-Maruyama predictor corrector (EMPEC2) method

$$\bar{w}(\boldsymbol{r}) - w^j(\boldsymbol{r}) = \Delta t\, G(\boldsymbol{r}; [w^j]) + R^j(\boldsymbol{r}) \tag{16}$$

$$w^{j+1}(\boldsymbol{r}) - w^j(\boldsymbol{r}) = \frac{\Delta t}{2}(G(\boldsymbol{r}; [w^j]) + G(\boldsymbol{r}; [\bar{w}])) + R^j(\boldsymbol{r}) \tag{17}$$

It is important to use the same noise realization $R^j(\boldsymbol{r})$ in both the predictor and corrector steps in order to fully cancel the leading-order weak error of the algorithm. The noise statistics here are the same as for the EM1 method.

There are also variants of this algorithm that use linearized force information, such as an algorithm due to Petersen and Öttinger (PO2).[40,46] We also introduce here an ETD type algorithm that is similar to the predictor corrector algorithms, but is instead based on a Runge-Kutta approach.[44] This ETDRK2 algorithm is defined by

$$\hat{\bar{w}}(\boldsymbol{k}) - \hat{w}^j(\boldsymbol{k}) = \frac{1 - e^{-\hat{c}(k)\Delta t}}{\hat{c}(k)} \hat{G}(\boldsymbol{k}; [w^j]) + \left(\frac{1 - e^{-2\hat{c}(k)\Delta t}}{2\hat{c}(k)\Delta t}\right)^{1/2} \hat{R}^j(\boldsymbol{k}) \tag{18}$$

$$\hat{w}^{j+1}(\boldsymbol{k}) - \hat{\bar{w}}(\boldsymbol{k}) = \frac{\hat{c}(k)\Delta t + e^{-\hat{c}(k)\Delta t} - 1}{(\hat{c}(k))^2 \Delta t} \left(\hat{F}(\boldsymbol{k}; [\bar{w}]) - \hat{F}(\boldsymbol{k}; [w^j])\right) \tag{19}$$

The predictor step is an ETD1 step as described above, while the corrector step is based on a trapezoidal Runge-Kutta approximation. Note that this algorithm is slightly different

than a predictor corrector ETD method described elsewhere.[45]

The final type of algorithm we consider is an adaptive time-stepping (ADT) method for CL simulations. This approach was introduced by Aarts and coworkers in the context of CL sampling of quantum chromodynamics models.[47] The first adaptive time-stepping algorithm that we consider, EM1ADT, uses a simple Euler-Maruyama update scheme with a time step that is updated between iterations according to

$$\Delta t_j = \frac{K}{\max(|G(\boldsymbol{r};[w^j])|)}\overline{\Delta t} \tag{20}$$

$$w^{j+1}(\boldsymbol{r}) = w^j(\boldsymbol{r}) + \Delta t_j\, G(\boldsymbol{r};[w^j]) + R^j(\boldsymbol{r}) \tag{21}$$

where $\overline{\Delta t}$ is the nominal time step and K is an adjustable parameter. Typically K is set to be close to the average modulus of the force so that if a large force value is encountered, the time step is reduced to allow for more accurate time integration. In contrast, if the forces are small, then Δt is increased to allow for sampling more states. In all ADT calculations in this work, the parameter K was computed by averaging the modulus of the force over the first 1000 iterations, unless otherwise specified. No adaptive time-stepping is performed over these initial calibration steps and the iteration proceeds with the nominal time step. One can determine if the ADT method is well calibrated by plotting the time step to ensure it fluctuates around the nominal value $\overline{\Delta t}$. Operator values must also be weighted by the adaptive time step when computing averages and other statistics. The adaptive time-stepping approach can easily be generalized to any other algorithm. A second ADT scheme that we consider here is the EMPEC2ADT method, which layers adaptive time-stepping on top of

EMPEC2 updates:

$$\Delta t_j = \frac{K}{\max(|G(\boldsymbol{r};[w^j])|)}\overline{\Delta t} \tag{22}$$

$$\bar{w}(\boldsymbol{r}) = w^j(\boldsymbol{r}) + \Delta t_j\, G(\boldsymbol{r};[w^j]) + R^j(\boldsymbol{r}) \tag{23}$$

$$w^{j+1}(\boldsymbol{r}) = w^j(\boldsymbol{r}) + \frac{\Delta t_j}{2}(G(\boldsymbol{r};[\bar{w}]) + G(\boldsymbol{r};[w^j])) + R^j(\boldsymbol{r}) \tag{24}$$

It is important for both the EM1ADT and EMPEC2ADT schemes that the variance of the noise is adjusted with the time step: $\langle R^j(\boldsymbol{r})R^k(\boldsymbol{r}')\rangle = 2\Delta t_j\,\delta_{jk}\delta(\boldsymbol{r}-\boldsymbol{r}')$.

As a final note, there are other small changes to each algorithm that can be made. For both SCFT and CL fictitious dynamics simulations, we include a positive, constant mobility λ as a coefficient to the force for each field and in the noise variance. These mobilities affect the relative speed at which each field is updated. Additionally, when performing SCFT calculations with first-order fictitious dynamics algorithms (EM1, SI1, ETD1), the fields are not updated simultaneously but rather follow a staggered scheme where only one field is updated per iteration and the stiffer w_+ pressure mode is updated first.[34] For the AB diblock model considered here, the staggered updates with the EM1 algorithm can be described mathematically as,

$$w_+^{j+1}(\boldsymbol{r}) = w_+^j(\boldsymbol{r}) + \lambda_+\Delta t\, G_+(\boldsymbol{r};[w_+^j, w_-^j]) \tag{25}$$

$$w_-^{j+1}(\boldsymbol{r}) = w_-^j(\boldsymbol{r}) + \lambda_-\Delta t\, G_-(\boldsymbol{r};[w_+^{j+1}, w_-^j]) \tag{26}$$

For CL calculations, all fields are always updated simultaneously, i.e.

$$w_+^{j+1}(\boldsymbol{r}) = w_+^j(\boldsymbol{r}) + \lambda_+\Delta t\, G_+(\boldsymbol{r},[w_+^j, w_-^j]) + R_+^j(\boldsymbol{r}) \tag{27}$$

$$w_-^{j+1}(\boldsymbol{r}) = w_-^j(\boldsymbol{r}) + \lambda_-\Delta t\, G_-(\boldsymbol{r},[w_+^j, w_-^j]) + R_-^j(\boldsymbol{r}) \tag{28}$$

Including a mobility $\lambda_\pm$ changes the variance of the noise to $\langle R_\pm^j(\boldsymbol{r})R_\pm^k(\boldsymbol{r}')\rangle = 2\lambda_\pm\Delta t\,\delta_{jk}\delta(\boldsymbol{r}-$

$\boldsymbol{r}'$).

Software

All calculations involving Anderson mixing were computed using the publicly available PSCF code developed at the University of Minnesota.[31] Both the FORTRAN90 and C++ versions of the code were tested, with quantitative agreement between the two. All data presented in this work used the C++ version of the code. The FORTRAN90 version of the code was pulled on February 2, 2020 (SHA1 67d107f3a6) and the C++ version was pulled on July 1, 2020 (SHA1 3ed5f6caac). All calculations involving fictitious dynamics or CL sampling were performed using a UCSB-developed custom C++ code.

All SCFT calculations were computed using a single thread on an Intel Xeon E5-2630 CPU. All CL simulations were executed on NVIDIA V100 GPUs.[48]

Results

Self-Consistent Field Theory

We first evaluate the various algorithms in the context of SCFT. For each algorithm we examine the stability properties and the ways its performance can be adjusted via numerical parameters.

There is one adjustable parameter for the AM method: the history length, n_h. Although keeping a longer history length typically leads to faster convergence of field updates, it also increases the computational cost of the calculation. This is both due to increased memory requirements of storing the field history as well as the arithmetic operations required to update the fields, which scale like $\mathcal{O}(n_h^2 M)$. Recall that to evaluate the force requires $\mathcal{O}(N_s M \ln(M))$ operations, so if $n_h^2 \sim N_s \ln(M)$ then field updates can have comparable cost to that of evaluating the force. For typical parameter values of $N_s = 100$ and $M = 64^3$ this crossover occurs at $n_h = 35$. To combat this problem it is common practice when

performing field updates to map the fields from the original grid in real space with M samples to a symmetry reduced space with M_r samples, where $M_r \ll M$.[29,30] This leads to a field update cost of $n_h^2 M_r$. After conducting the field update, the fields are then mapped back to the original grid to perform pseudospectral evaluations of the force. In the case of the highly symmetric double gyroid phase, which has the space group $Ia\bar{3}d$, a full grid of size $M = 64^3$ is reduced to $M_r = 2761$, corresponding to a factor of 95 reduction. In addition to reducing the computational cost of a given field update step, using symmetry reduction also decreases the dimension of the space that must be searched for the saddle-point, which typically leads to faster convergence. A disadvantage to using symmetry reduction is that it limits the overall set of structures that are possible to explore. Performing large-cell quenches to discover phases,[49–52] and to study defects, thin films, and other asymmetric structures is not possible using techniques that impose the symmetry elements of a space group on the fields.

We compare the number of iterations and the time it takes to converge a calculation to a force tolerance of 10^{-8} using the AM algorithm for a unit cell of the double gyroid phase with and without symmetry reduction at $f = 0.34$ and $\chi N = 30$ in Figure 1. All calculations were initialized from a converged SCFT calculation at $\chi N = 20$ and $f = 0.37$ and were run until a prescribed force cutoff of 10^{-8} was achieved. When constructing a phase diagram, it is common to use one point in phase space with a converged structure to initialize a nearby point in phase space to accelerate the saddle-point search. As such, the change in both χN and f between the seed fields and the converged fields is a representative test for the algorithms.

The top panel of Figure 1 illustrates that increasing history length reduces the iterations required to converge, nearly monotonically. This is expected because a longer history length leads to a more efficient search towards the saddle-point. Additionally, the symmetrized calculations ($Ia\bar{3}d$) converge with fewer iterations than the calculations without symmetrization ($P1$). In the bottom panel, the CPU time required to complete the calculation is plotted.

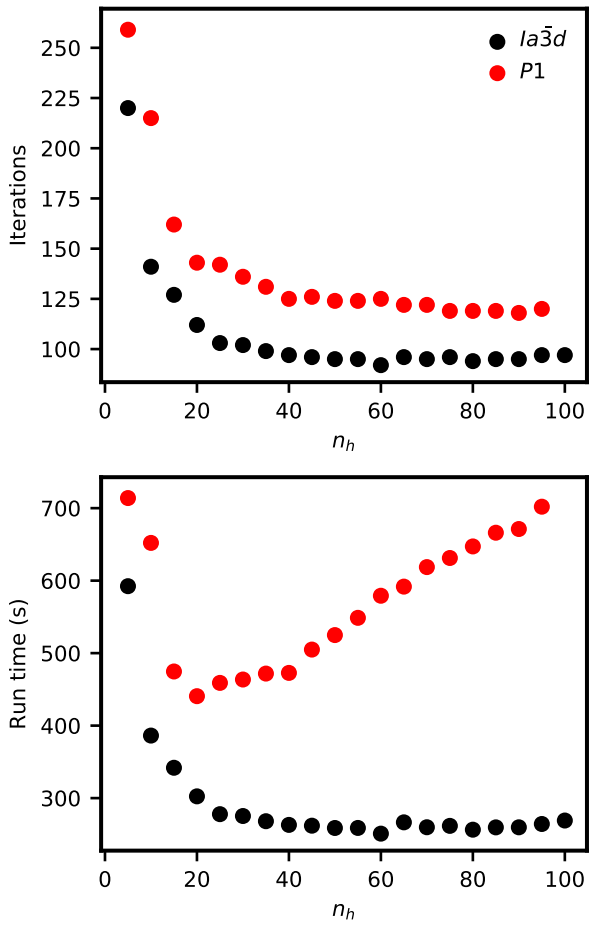

Figure 1: Convergence rate for AM method with and without symmetrization at $\chi N = 30$. $P1$ indicates that no symmetry reduction is used, whereas $Ia\bar{3}d$ uses the symmetry of the double gyroid phase. The spatial discretization with no symmetry reduction was $M = 64^3$. All calculations were performed using a single thread on an Intel Xeon E5-2630 CPU.

For the symmetrized case, the run time nearly perfectly mirrors the iterations in the top panel. This indicates that the increased computational cost of larger n_h is negligible in the total cost of an iteration. For the $P1$ calculations, the conclusion is quite different. The CPU time required decreases as n_h is increased from 5 to 20, but for $n_h \geq 25$ the CPU time increases with increasing n_h despite the fact that the total number of field update iterations is decreasing. This clearly indicates that the cost of a field update is no longer negligible and is affecting the overall run time. The transition occurs close to the prior scaling-based estimate of $n_h = 35$. This procedure was repeated at $\chi N = 40$ and $\chi N = 60$ with qualitatively the same conclusions. The only major difference occurred at $\chi N = 60$, where the $P1$ calculations would not converge for any value of n_h attempted.

For the fictitious dynamics algorithms, the adjustable parameters are the time step Δt and mobilities, λ_+ and λ_-, used to update each field. Each field mobility can be set independently, but only two out of the three time step and mobility parameters are independent because they appear in the combinations $\lambda_+ \Delta t$ and $\lambda_- \Delta t$ in all algorithms. For the SCFT studies we therefore arbitrarily set $\Delta t = 1$ and vary the two mobilities. Because fictitious dynamics algorithms have one more parameter compared to AM, the parameter space for fictitious dynamics is larger and requires more work to optimize the performance. Some heuristics can be constructed to help reduce the burden of this search, however.[39]

In analogy to the AM method, we evaluated the various time steppers over a range of field mobility values. These data are presented in Figure 2. In all cases the calculations were initialized from the same fields used in the AM studies and were run until the l_2 norm of the forces on w_+ and w_- were both less than 10^{-8}.

Unlike the AM method, changing the parameters does not affect the arithmetic cost of a field update step, so we only consider the number of iterations required to reach the force cutoff. Figure 2 shows that the EM1 method is by far the least stable and only converges for $\lambda_+ = \lambda_- = 1$ out of the values attempted. The mobility values attempted here are relatively aggressive and most algorithms will have larger stability windows at smaller mobility values.

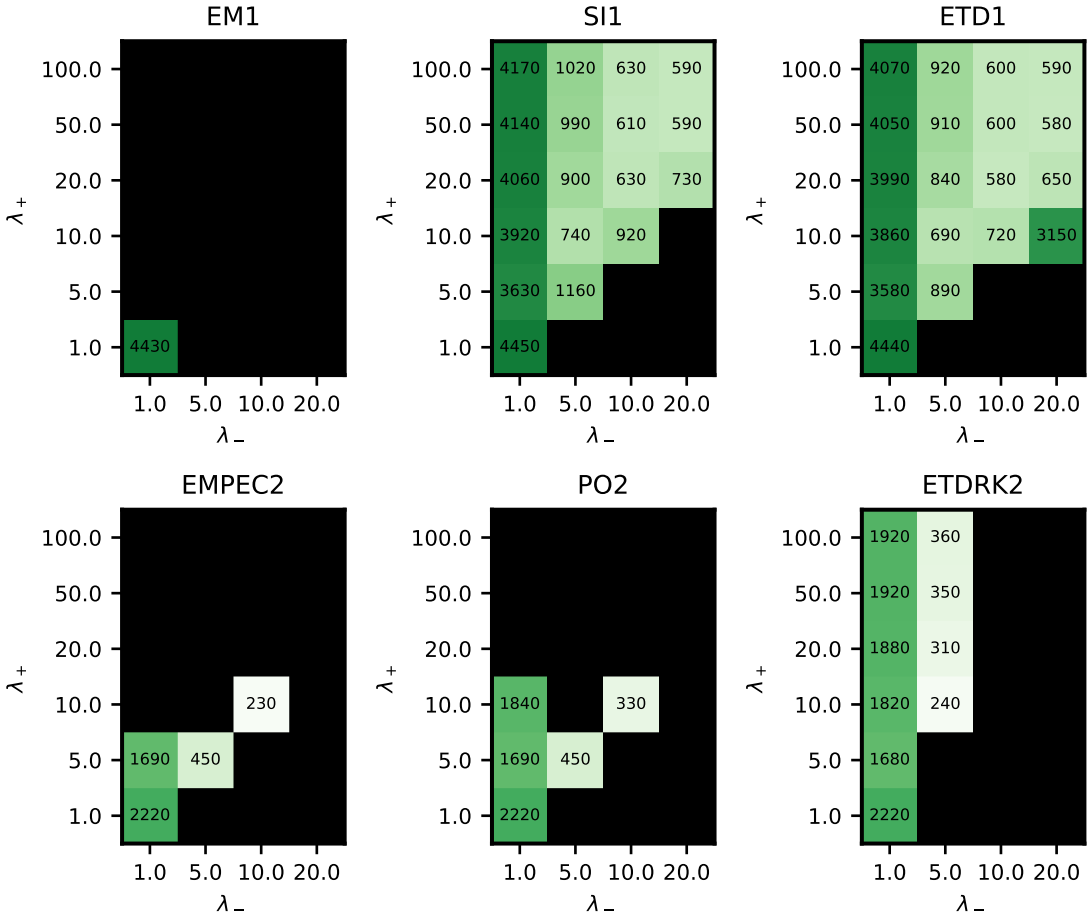

Figure 2: Iterations to convergence for time steppers at $\chi N = 30$. λ_+ is the time step for the w_+ field and λ_- is the time step for the w_- field. The number inside each square indicates iterations to convergence. Black squares indicate that the calculation did not converge.

The semi-implicit, first-order algorithms (SI1 and ETD1) perform much better than EM1 and converge for nearly any $\lambda_+ > \lambda_-$. The fastest convergence from the values considered occurs for $\lambda_+ = 50$ and $\lambda_- = 20$. Although not shown, increasing either mobility above these values starts to destabilize the algorithm and leads to slower convergence. Finally, the second order algorithms show an intermediate level of stability, but converge quite quickly with optimal parameter selection. The EMPEC2 and PO2 algorithms perform best for $\lambda_+ = \lambda_-$, while the ETDRK2 algorithm performs best with $\lambda_+ > \lambda_-$. These three algorithms require approximately half as many iterations as the SI1 and ETD1 algorithms, but each iteration requires twice as many force evaluations, so it is unclear which is faster from these data alone.

To further probe the question of which algorithm converges fastest, we examine the absolute value of the error in the intensive Hamiltonian as a function of the number of force evaluations. Comparing the different algorithms based on iterations to convergence can be misleading because the second order algorithms require twice as many force evaluations per iteration. It is also not reliable to compare run times from different software implementations that were used for different algorithms. Many software design decisions such as single vs. double precision arithmetic, numerical library selection, and hardware availability (CPU vs. GPU) can outweigh the effect of algorithm choice. As such we use number of force evaluations as the computational effort metric because, apart from the previously mentioned edge case of non-symmetrized AM with long history length, evaluating the forces should typically be the dominant computational burden and the number of times that this is required therefore determines the overall run time.

Figure 3 shows the error in the intensive Hamiltonian versus the number of force evaluations conducted for the various algorithms at $\chi N = 30$. All calculations were run until the change in the Hamiltonian between iterations was less than 10^{-10}. The error was then referenced to this final value of the Hamiltonian. For all algorithms we used the parameter values that led to the lowest CPU time to convergence. For most applications an error of

10^{-4} to 10^{-6} in the Hamiltonian is sufficient to accurately compute phase boundaries. At $\chi N = 30$, the EMPEC2 time stepper and both AM algorithms are all nearly equivalent. The ETDRK2 algorithm is slightly slower, and is then followed by PO2, ETD1, and SI1. Finally EM1 is much slower to converge than any other algorithm. Increasing χN to 40 or 60 leads to some slight variations in the relative performance of the algorithms shown in Figure 4 and Figure 5. For very high accuracy (tighter than 10^{-7}) and strong segregation strengths, the AM algorithm with symmetrization shows faster convergence than any of the time steppers (Figure 5). The time steppers demonstrate a long tail of slow convergence at very high accuracy for strong segregation. AM without symmetrization is not stable at $\chi N = 60$, re-emphasizing the importance of symmetrization to the efficacy of the AM algorithm.

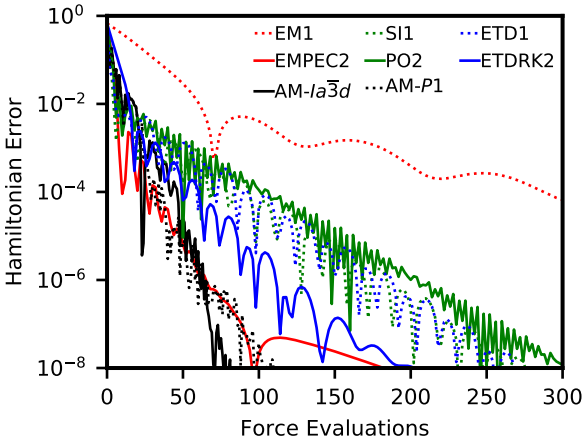

Figure 3: Error in the per-chain intensive Hamiltonian in units of k_BT after a given number of force evaluations. All calculations were conducted at $\chi N = 30$.

For all conditions tested, the EMPEC2 algorithm was fastest of all the time steppers, despite being one of the most simple. All of the semi-implicit algorithms rely on a linear response derived in the disordered phase. This linear response information is not a good match for the true linear force in the ordered double gyroid phase being tested here, which may limit or eliminate the benefit from the implicit part. The EMPEC2 algorithm on the other hand obtains all information about force variation over a time interval numerically from the predictor-corrector scheme and does not rely on approximate linear response information.

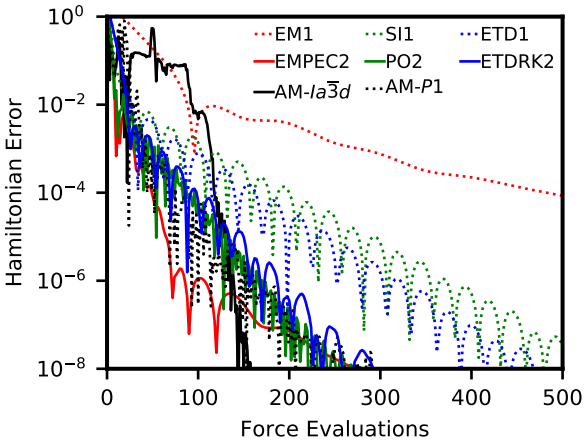

Figure 4: Same as Figure 3 but at $\chi N = 40$.

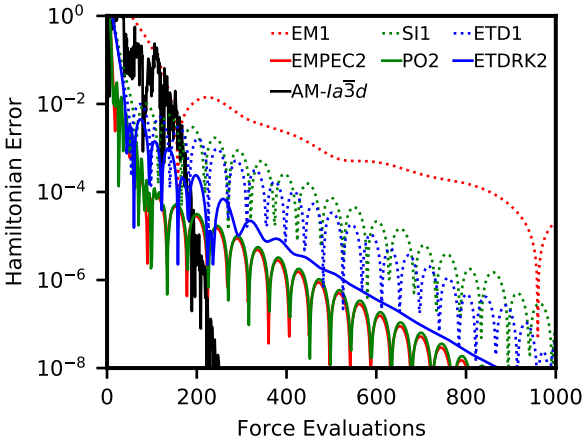

Figure 5: Same as Figure 3 but at $\chi N = 60$.

Additionally, the second-order schemes nearly universally converge faster than the first-order algorithms, despite having smaller stability windows. This may be attributed to the lower time accuracy of the first-order schemes. Although in SCFT we are concerned only with the effort to find the saddle point and not with accurately reproducing a dynamical trajectory, the time-step errors in the first-order methods can make the trajectory divert from the fastest path to convergence. This can be seen in the oscillatory change in the Hamiltonian in Figures 3–5.

Field Theoretic Simulations

We now consider full FTS that are not limited by the mean field approximation of SCFT. As previously stated, the most efficient way to conduct such simulations without approximation is with the complex Langevin method. Unfortunately, there is no easy way to convert the AM algorithm into one that can correctly sample fluctuations in an FTS, so we do not consider it for the remainder of this work. The time steppers on the other hand can trivially be extended to include fluctuations by including an additional noise term in the update. We begin by evaluating the performance of the different algorithms in the *disordered* phase at $f = 0.34$, $\chi N = 10$ and $C = 20$ in a cubic box of size $V = 9^3\, R_g^3$ with $M = 48^3$. The melt is also compressible with $\zeta N = 100$ and the polymer density has been smeared with a range of $a = 0.2\, R_g$. The mobilities are fixed at $\lambda_+ = 2$ and $\lambda_- = 1$. In FTS, the effective Hamiltonian is not a physically relevant operator, so we instead consider the excess chemical potential relative to the ideal gas of copolymer chains, the average of which is plotted in Figure 6 for a range of time step sizes. The excess chemical potential operator is defined as $\tilde{\mu}_{\text{ex}} = -\ln Q[w_+, w_-]$ where $\tilde{\mu}_{\text{ex}}$ is in units of the thermal energy $k_B T$. The physical observable, μ_{ex}, is then computed according to

$$\mu_{\text{ex}} = \langle \tilde{\mu}_{\text{ex}} \rangle = \frac{\int \mathcal{D}w_+ \int \mathcal{D}w_- \, \tilde{\mu}_{\text{ex}} \exp(-H[w_+, w_-])}{\int \mathcal{D}w_+ \int \mathcal{D}w_- \exp(-H[w_+, w_-])} \tag{29}$$

where $\langle \ldots \rangle$ indicates a thermodynamic ensemble average. Under the ergodic principle with CL importance sampling, the ensemble average can be replaced with a CL time average.

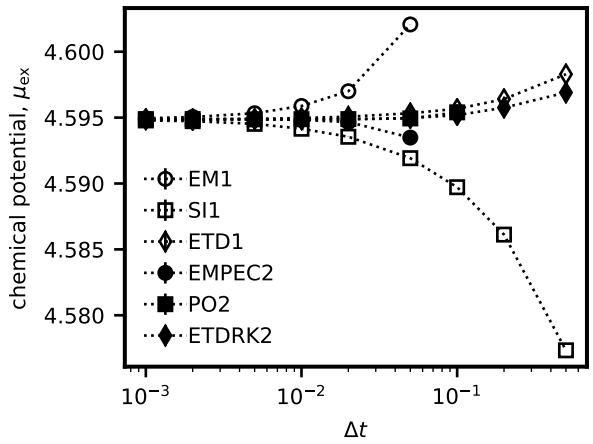

Figure 6: Average value of the chemical potential for the disordered phase at $\chi N = 10$, $f = 0.34$, and $C = 20$.

An ideal algorithm will be able to produce accurate values at large time steps in order to reduce simulation times. At small Δt all algorithms show good agreement, but as Δt is increased various algorithms start to show time-integration errors. For CL simulations it is important to accurately reproduce the *trajectory* of the fictitious dynamics in order to properly importance sample the system. The first-order methods (EM1, SI1, and ETD1) have the worst accuracy scaling with Δt and show the strongest divergence from the true value of μ_{ex} as Δt increases. In particular the EM1 and SI1 algorithms perform worst, while the ETD1 algorithm is nearly as accurate as the second order methods, which weakly diverge from the true value of μ_{ex} with increasing Δt over the range of time steps tested. In the Supporting Information we replot the data in Figure 6 as an error relative to a high accuracy result and show that the expected first order scaling is obtained for the EM1, SI1 and ETD1 methods. Although perhaps less limiting than accuracy requirements for FTS, another important aspect for an algorithm is its stability. The most stable algorithm will be able to run at large Δt, which enables sampling more CL time for a given amount of CPU time. This leads to tighter confidence intervals for estimates (via reduced statistical

sampling error) for a given amount of resources.

In the disordered phase there is a clear maximum time step for each algorithm, above which the simulation diverges in fewer than 100 iterations; these conditions are found where the lines of Figure 6 terminate for large Δt. The most stable algorithms, i.e. the ones with largest maximum Δt, are the semi-implicit algorithms SI1, ETD1, and ETDRK2 which have $\Delta t_{\max} = 0.5$. The PO2 algorithm uses semi-implicit information in the corrector step but relies on a fully explicit Euler step in the predictor, which confers it with worse stability ($\Delta t_{\max} = 0.1$) compared to the other semi-implicit methods. Finally, the EM1 and EM-PEC2 algorithms show the worst stability of the tested algorithms ($\Delta t_{\max} = 0.05$), which is unsurprising as they make no attempt to use semi-implicit information. Notably in this disordered system, the linear response functions used to construct the semi-implicit algorithms are accurate for weak fluctuations about the homogeneous state, which likely leads to the much better performance of these algorithms compared to SCFT of the double gyroid phase. When considering both stability and accuracy, the ETD1 and ETDRK2 algorithms perform best and produce comparable levels of error. The ETD1 algorithm requires half as many force evaluations as the ETDRK2 algorithm, however, making it the most cost efficient algorithm. The high accuracy and stability found for the ETD1 algorithm are in agreement with similar studies of the Edwards homopolymer solution model[41] and three and four-species block polymer melts.[39]

Repeating the same CL calculations at $\chi N = 30$ for the double gyroid phase yields qualitatively similarly results, though the trends are not quite as clear. These data are presented in Figure 7. At small Δt all algorithms converge to the same value. As Δt is increased, the algorithms with first-order accuracy start to show significant errors while second-order accurate algorithms remain close to the true value. Unlike in the disordered phase, there is no clear maximum time step for each algorithm, however. Instead, as Δt is increased, the algorithms become statistically more likely to follow divergent trajectories. This can be quantified via a mean time to divergence, defined as the harmonic mean of the

CL simulation time before a calculation diverges. Mathematically,

$$\bar{\tau}_{div} = \left(\frac{1}{n_s} \sum_{j=1}^{n_s} \frac{1}{\tau_{div,j}} \right)^{-1} \tag{30}$$

where $\tau_{div,j}$ is the divergence time for an individual trajectory and j indexes statistically independent simulations of which there are n_s. A simulation is terminated as divergent when any individual field value is IEEE 754-defined +INF, -INF or NAN. The mean divergence times were computed from ten independent trajectories and are plotted in Figure 8.

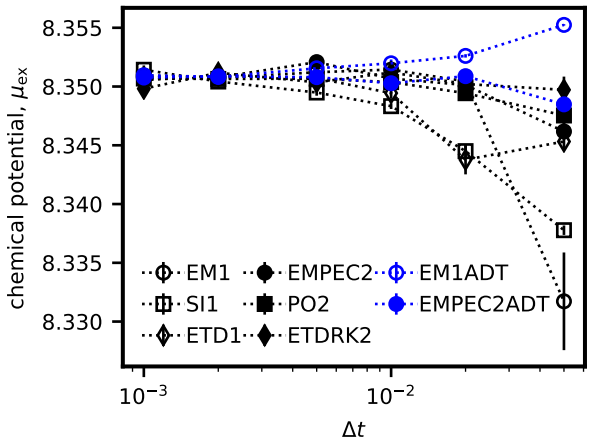

Figure 7: Average value of the chemical potential for the double gyroid phase at $\chi N = 30$, $f = 0.34$, and $C = 20$.

All calculations were run for a maximum of two million iterations. The solid black line in Figure 8 indicates the maximum CL time that could be achieved with two million iterations at a given time step. For $\Delta t \leq 0.005$ all algorithms have a divergence time that lies on top of the $\tau_{\max}$ curve, indicating that no calculations diverged for the entire CL time window. At larger Δt various algorithms have $\tau_{\text{div}} < \tau_{\max}$ indicating that divergence became limiting rather than the specified iteration cap. For $\Delta t = 0.05$ all algorithms except EM1ADT have $\tau_{div} \ll \tau_{max}$. In contrast to the simulations for the disordered phase, simulations of a microphase separated system show no overall performance benefit from algorithms using semi-implicit information over the Euler-based methods, which is similar to the behavior

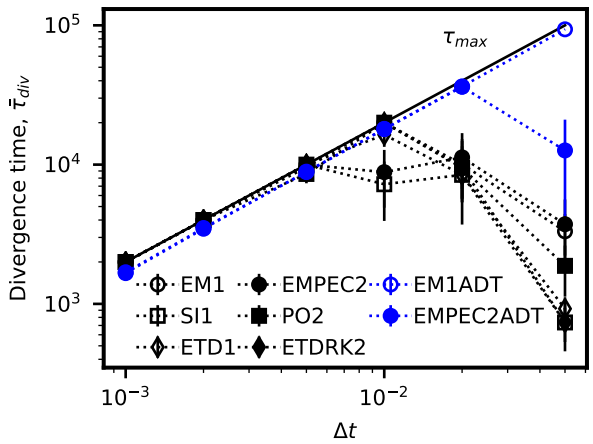

Figure 8: Harmonic mean divergence time for the double gyroid phase at $\chi N = 30$, $f = 0.34$, and $C = 20$.

for SCFT of the double gyroid phase. Again this is presumably due to the fact that the linear response functions used in the semi-implicit algorithms are not appropriate for the double gyroid phase. The algorithms that perform best are the ones that use adaptive time stepping. The EM1ADT algorithm never diverged over the range of time steps used in this study, while the EMPEC2ADT algorithm only diverged for $\Delta t = 0.05$.

In the supporting information we explore in further detail what actually causes a CL simulation to diverge. We conclude that even though the saddle-point is a local fixed point of the CL dynamics, it is possible to fluctuate to a nearby trajectory that is analytically divergent (non-bound). These divergent trajectories are not attractive, but cause the force to grow exponentially, which can lead to poor time-integration accuracy and the emergence of extremely large field values. It is also possible for trajectories to leave the basin of attraction for the desired microphase and enter an attractive basin for another microphase (i.e. free-energy barrier crossing) or an unphysical state. This topic is explored further in the supporting information. If a very small time step is used on the periphery of the basin near a divergent trajectory, then the algorithm can be significantly stabilized. The adaptive time stepping algorithms do exactly this: when the force is large the time step is small. The EMPEC2ADT algorithm is slightly less stable than the EM1ADT algorithm because

EMPEC2ADT chooses the time step based on the current value of the force, but updates the fields based on the current and future forces. When near a divergent trajectory this can lead to reduced suppression of instabilities. Nevertheless the EMPEC2ADT algorithm has much better accuracy compared to EM1ADT (see Figure 7), and may be a better choice if stability is not limiting. As noted previously, it is possible to construct adaptive time-stepped versions of the other algorithms, but the fixed-time-step semi-implicit algorithms do not perform better than the Euler-based algorithms in the double gyroid phase, so we do not expect the ADT versions to perform better either.

As a final study, we examine how the adaptive time stepping algorithms can be used to run simulations in regions of parameter space that were previously inaccessible. We consider the same diblock copolymer system, but now at $\chi N = 80$ and $\zeta N = 200$ and in a cubic box size $V = 12\, R_g^3$ with $M = 64^3$. The time step is fixed at $\Delta t = 0.01$ and we examine algorithm performance as the chain number density, C, is varied. As C decreases the relative strength of fluctuations increases, leading to increased probability for divergence. The divergence time is plotted in Figure 9. All calculations were initialized from the SCFT saddle-point fields of the same double gyroid phase. This initialization choice causes a warm up period for every trajectory, during which the fluctuations are slowly incorporated as the simulation evolves. The warm up time for each algorithm is presented in Figure 10.

Figure 9 again shows that adaptive time-stepping is much more stable than fixed-time-step approaches. Over all values of C considered, the EM1ADT algorithm never diverged. Although not shown, the algorithm is also stable below $C = 1$, however the double gyroid phase starts to become unstable relative to the disordered phase because of strong fluctuations at small C. In contrast, the fixed-time-step algorithms (EM1 and EMPEC2) have a decreasing divergence time with decreasing C. As fluctuations become stronger at small C, the fixed-time-step algorithms become susceptible to an instantaneous perturbation to the fields which pushes the fields far away from the saddle-point, and can lead to a divergent trajectory. The adaptive time steppers on the other hand can accurately integrate forward

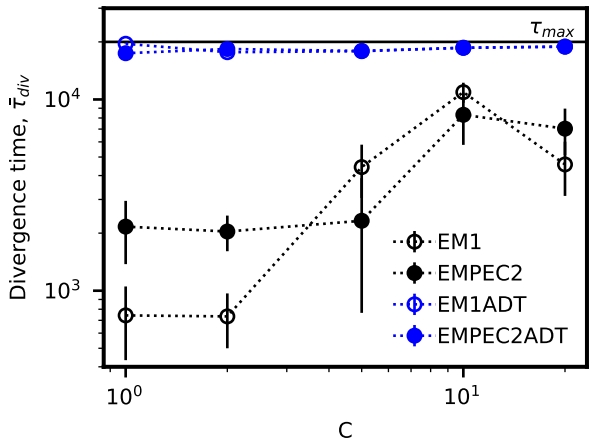

Figure 9: Harmonic mean divergence time for the double gyroid phase at $\chi N = 80$ and $f = 0.34$.

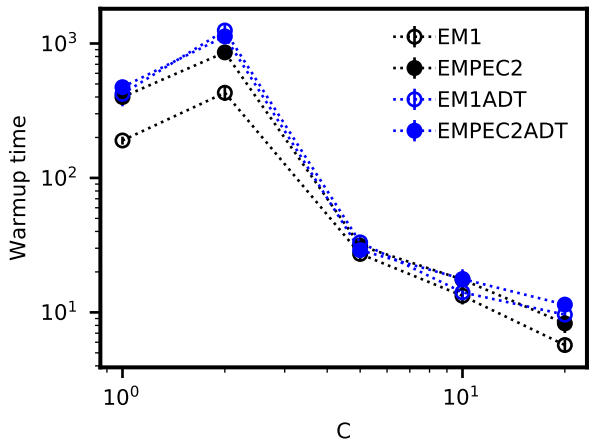

Figure 10: Mean warm up time for the double gyroid phase at $\chi N = 80$ and $f = 0.34$.

in time and prevent escape from the stable basin.

The problem of simulating small C with fixed-time-step algorithms is further accentuated by examining the warm up time in Figure 10. For all algorithms the warm up time increases as C decreases, indicating that fluctuations away from the saddle-point become more important as C is decreased. For $C < 2$ the warm up time for the EM1 and EMPEC2 algorithms is approaching the divergence time for each algorithm. This makes it difficult to collect any data because the simulation is likely to diverge by the time the system has become thermalized. Such stability problems are not present with adaptive time stepping, but the averaging time for the K parameter had to be increased to 50,000 iterations for $C \leq 2$.

Conclusion

We examined a wide array of field update algorithms to perform polymer SCFT calculations and field theoretic simulations (FTS). These algorithms fall into two groups: Anderson-mixing (AM) and fictitious dynamics. We show that AM and fictitious dynamics algorithms perform similarly under physically relevant conditions for SCFT, as long as the numerical parameters have been tuned appropriately. The AM approach is advantaged in that it has fewer numerical parameters compared to the fictitious dynamics algorithms, but the latter methods are more robust when simulating systems with low spatial symmetry. We also find that fictitious dynamics algorithms with second-order time accuracy can outperform first-order methods despite doubled cost per time step.

For complex Langevin field theoretic simulations we observed that the exponential time-differencing type algorithms outperform all others in disordered phases because of the availability of useful linear response information and their sophisticated use of this information. In ordered mesophases, adaptive time stepping (ADT) is found to dramatically stabilizes algorithms and is much more important than individual algorithm choice. The ADT algo-

rithms also allow access to parameter spaces characteristic of strong fluctuations (e.g. low C) that were previously intractable. These insights should aid future field theoretic simulations of a wide variety of polymer models.

Acknowledgement

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1650114 and by the National Science Foundation CMMT Program under Award Nos. DMR-1822215 and DMR-2104255. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Use was made of computational facilities purchased with funds from the National Science Foundation (OAC-1925717) and administered by the Center for Scientific Computing (CSC). The CSC is supported by the California NanoSystems Institute and the Materials Research Science and Engineering Center (MRSEC; NSF DMR 1720256) at UC Santa Barbara.

Supporting Information Available

Forces for diblock copolymer melt, CL algorithm error scaling, visualization of a divergent trajectory, visualization of a non-ergodic trajectory, simplified model for studying CL, and literature summary.

References

(1) Gennes, P.-G. d. *Scaling concepts in polymer physics*; Cornell University Press: Ithaca, N.Y, 1979.

(2) Leibler, L. Theory of Microphase Separation in Block Copolymers. *Macromolecules* **1980**, *13*, 1602–1617, DOI: 10.1021/ma60078a047.

(3) Edwards, S. F. The statistical mechanics of polymers with excluded volume. *Proc. Phys. Soc., London* **1965**, *85*, 613–624, DOI: 10.1088/0370-1328/85/4/301.

(4) Edwards, S. F. The theory of polymer solutions at intermediate concentration. *Proc. Phys. Soc., London* **1966**, *88*, 265–280, DOI: 10.1088/0370-1328/88/2/301.

(5) Schmid, F. Self-consistent-field theories for complex fluids. *J. Phys.: Condens. Matter* **1998**, *10*, 8105–8138, DOI: 10.1088/0953-8984/10/37/002.

(6) Matsen, M. W. The standard Gaussian model for block copolymer melts. *J. Phys.: Condens. Matter* **2001**, *14*, R21–R47, DOI: 10.1088/0953-8984/14/2/201.

(7) Fredrickson, G. H. *The Equilibrium Theory of Inhomogeneous Polymers*; International series of monographs on physics (Oxford, England) ; 134; Clarendon Press: Oxford; New York, 2006.

(8) Shi, A.-C. Self-Consistent Field Theory of Inhomogeneous Polymeric Systems: A Variational Derivation. *Adv. Theory Simul.* **2019**, *2*, 1800188, DOI: https://doi.org/10.1002/adts.201800188.

(9) Ganesan, V.; Fredrickson, G. H. Field-Theoretic Polymer Simulations. *Europhys. Lett.* **2001**, *55*, 814–820, DOI: 10.1209/epl/i2001-00353-8.

(10) Fredrickson, G. H.; Ganesan, V.; Drolet, F. Field-Theoretic Computer Simulation Methods for Polymers and Complex Fluids. *Macromolecules* **2002**, *35*, 16–39, DOI: 10.1021/ma011515t.

(11) Delaney, K. T.; Fredrickson, G. H. Recent Developments in Fully Fluctuating Field-Theoretic Simulations of Polymer Melts and Solutions. *J. Phys. Chem. B* **2016**, *120*, 7615–7634, DOI: 10.1021/acs.jpcb.6b05704, PMID: 27414265.

(12) Grzetic, D. J.; Delaney, K. T.; Fredrickson, G. H. Field-Theoretic Study of Salt-Induced

Order and Disorder in a Polarizable Diblock Copolymer. *ACS Macro Lett.* **2019**, *8*, 962–967, DOI: 10.1021/acsmacrolett.9b00316.

(13) Beardsley, T. M.; Matsen, M. W. Fluctuation correction for the order-disorder transition of diblock copolymer melts. *J. Chem. Phys.* **2021**, *154*, 124902, DOI: 10.1063/5.0046167.

(14) Popov, Y. O.; Lee, J.; Fredrickson, G. H. Field-theoretic simulations of polyelectrolyte complexation. *J. Polym. Sci., Part B: Polym. Phys.* **2007**, *45*, 3223–3230, DOI: 10.1002/polb.21334.

(15) Lee, J.; Popov, Y. O.; Fredrickson, G. H. Complex coacervation: A field theoretic simulation study of polyelectrolyte complexation. *J. Chem. Phys.* **2008**, *128*, 224908, DOI: 10.1063/1.2936834.

(16) Delaney, K. T.; Fredrickson, G. H. Theory of polyelectrolyte complexation–Complex coacervates are self-coacervates. *J. Chem. Phys.* **2017**, *146*, 224902, DOI: 10.1063/1.4985568.

(17) Panagiotou, E.; Delaney, K. T.; Fredrickson, G. H. Theoretical prediction of an isotropic to nematic phase transition in bottlebrush homopolymer melts. *J. Chem. Phys.* **2019**, *151*, 094901, DOI: 10.1063/1.5114698.

(18) Düchs, D.; Ganesan, V.; Fredrickson, G. H.; Schmid, F. Fluctuation Effects in Ternary AB + A + B Polymeric Emulsions. *Macromolecules* **2003**, *36*, 9237–9248, DOI: 10.1021/ma030201y.

(19) Vorselaars, B.; Spencer, R. K. W.; Matsen, M. W. Instability of the Microemulsion Channel in Block Copolymer-Homopolymer Blends. *Phys. Rev. Lett.* **2020**, *125*, 117801, DOI: 10.1103/PhysRevLett.125.117801.

(20) Spencer, R. K. W.; Matsen, M. W. Coexistence of Polymeric Microemulsion with Homopolymer-Rich Phases. *Macromolecules* **2021**, *54*, 1329–1337, DOI: 10.1021/acs.macromol.0c02668.

(21) Liu, Y.-X.; Delaney, K. T.; Fredrickson, G. H. Field-Theoretic Simulations of Fluctuation-Stabilized Aperiodic "Bricks-and-Mortar" Mesophase in Miktoarm Star Block Copolymer/Homopolymer Blends. *Macromolecules* **2017**, *50*, 6263–6272, DOI: 10.1021/acs.macromol.7b01106.

(22) Tzeremes, G.; Rasmussen, K. O.; Lookman, T.; Saxena, A. Efficient Computation of the Structural Phase Behavior of Block Copolymers. *Phys. Rev. E* **2002**, *65*, 041806, DOI: 10.1103/PhysRevE.65.041806.

(23) Rasmussen, K. O.; Kalosakas, G. Improved Numerical Algorithm for Exploring Block Copolymer Mesophases. *J. Polym. Sci., Part B: Polym. Phys.* **2002**, *40*, 1777–1783, DOI: 10.1002/polb.10238.

(24) Ranjan, A.; Qin, J.; Morse, D. C. Linear Response and Stability of Ordered Phases of Block Copolymer Melts. *Macromolecules* **2008**, *41*, 942–954, DOI: 10.1021/ma0714316.

(25) Cochran, E. W.; Garcia-Cervera, C. J.; Fredrickson, G. H. Stability of the Gyroid Phase in Diblock Copolymers at Strong Segregation. *Macromolecules* **2006**, *39*, 2449–2451, DOI: 10.1021/ma0527707.

(26) Audus, D. J.; Delaney, K. T.; Ceniceros, H. D.; Fredrickson, G. H. Comparison of Pseudospectral Algorithms for Field-Theoretic Simulations of Polymers. *Macromolecules* **2013**, *46*, 8383–8391, DOI: 10.1021/ma401804j.

(27) Matsen, M. W.; Thompson, R. B. Equilibrium behavior of symmetric ABA triblock copolymer melts. *J. Chem. Phys.* **1999**, *111*, 7139–7146, DOI: 10.1063/1.480006.

(28) Thompson, R. B.; Rasmussen, K. O.; Lookman, T. Improved convergence in block copolymer self-consistent field theory by Anderson mixing. *J. Chem. Phys.* **2004**, *120*, 31–34, DOI: 10.1063/1.1629673.

(29) Matsen, M. W. Fast and accurate SCFT calculations for periodic block-copolymer morphologies using the spectral method with Anderson mixing. *Eur. Phys. J. E* **2009**, *30*, 361, DOI: 10.1140/epje/i2009-10534-3.

(30) Stasiak, P.; Matsen, M. W. Efficiency of Pseudo-Spectral Algorithms with Anderson Mixing for the SCFT of Periodic Block-Copolymer Phases. *Eur. Phys. J. E* **2011**, *34*, 110, DOI: 10.1140/epje/i2011-11110-0.

(31) Arora, A.; Qin, J.; Morse, D. C.; Delaney, K. T.; Fredrickson, G. H.; Bates, F. S.; Dorfman, K. D. Broadly Accessible Self-Consistent Field Theory for Block Polymer Materials Discovery. *Macromolecules* **2016**, *49*, 4675–4690, DOI: 10.1021/acs.macromol.6b00107.

(32) Jiang, K.; Huang, Y.; Zhang, P. Spectral method for exploring patterns of diblock copolymers. *J. Comput. Phys.* **2010**, *229*, 7796–7805, DOI: doi.org/10.1016/j.jcp.2010.06.038.

(33) Vigil, D. L.; García-Cervera, C. J.; Delaney, K. T.; Fredrickson, G. H. Linear Scaling Self-Consistent Field Theory with Spectral Contour Accuracy. *ACS Macro Lett.* **2019**, *8*, 1402–1406, DOI: 10.1021/acsmacrolett.9b00632.

(34) Ceniceros, H.; Fredrickson, G. Numerical Solution of Polymer Self-Consistent Field Theory. *Multiscale Model. Simul.* **2004**, *2*, 452–474, DOI: 10.1137/030601338.

(35) Alexander-Katz, A.; Fredrickson, G. H. Diblock Copolymer Thin Films: A Field-Theoretic Simulation Study. *Macromolecules* **2007**, *40*, 4075–4087, DOI: 10.1021/ma070005h.

(36) Stasiak, P.; Matsen, M. W. Monte Carlo Field-Theoretic Simulations for Melts of Symmetric Diblock Copolymer. *Macromolecules* **2013**, *46*, 8037–8045, DOI: 10.1021/ma401687j.

(37) Beardsley, T. M.; Spencer, R. K. W.; Matsen, M. W. Computationally Efficient Field-Theoretic Simulations for Block Copolymer Melts. *Macromolecules* **2019**, *52*, 8840–8848, DOI: 10.1021/acs.macromol.9b01904.

(38) Matsen, M. W. Field theoretic approach for block polymer melts: SCFT and FTS. *J. Chem. Phys.* **2020**, *152*, 110901, DOI: 10.1063/1.5145098.

(39) Düchs, D.; Delaney, K. T.; Fredrickson, G. H. A multi-species exchange model for fully fluctuating polymer field theory simulations. *J. Chem. Phys.* **2014**, *141*, 174103, DOI: 10.1063/1.4900574.

(40) Lennon, E. M.; Mohler, G. O.; Ceniceros, H. D.; García-Cervera, C. J.; Fredrickson, G. H. Numerical Solutions of the Complex Langevin Equations in Polymer Field Theory. *Multiscale Model. Simul.* **2008**, *6*, 1347–1370, DOI: 10.1137/070689401.

(41) Villet, M. C.; Fredrickson, G. H. Efficient field-theoretic simulation of polymer solutions. *J. Chem. Phys.* **2014**, *141*, 224115, DOI: 10.1063/1.4902886.

(42) Wang, Z.-G. Fluctuation in electrolyte solutions: The self energy. *Phys. Rev. E* **2010**, *81*, 021501, DOI: 10.1103/PhysRevE.81.021501.

(43) Alexander-Katz, A.; Moreira, A. G.; Sides, S. W.; Fredrickson, G. H. Field-theoretic simulations of polymer solutions: Finite-size and discretization effects. *J. Chem. Phys.* **2005**, *122*, 014904, DOI: 10.1063/1.1827211.

(44) Cox, S.; Matthews, P. Exponential Time Differencing for Stiff Systems. *J. Comput. Phys.* **2002**, *176*, 430 – 455, DOI: 10.1006/jcph.2002.6995.

(45) Villet, M. C.; Fredrickson, G. H. Numerical coarse-graining of fluid field theories. *J. Chem. Phys.* **2010**, *132*, 034109, DOI: 10.1063/1.3289723.

(46) Petersen, W. P. A General Implicit Splitting for Stabilizing Numerical Simulations of Itô Stochastic Differential Equations. *SIAM J. Numer. Anal.* **1998**, *35*, 1439–1451, DOI: 10.1137/0036142996303973.

(47) Aarts, G.; James, F. A.; Seiler, E.; Stamatescu, I.-O. Adaptive stepsize and instabilities in complex Langevin dynamics. *Phys. Lett. B* **2010**, *687*, 154–159, DOI: 10.1016/j.physletb.2010.03.012.

(48) Delaney, K. T.; Fredrickson, G. H. Polymer field-theory simulations on graphics processing units. *Comput. Phys. Commun.* **2013**, *184*, 2102–2110, DOI: 10.1016/j.cpc.2013.04.002.

(49) Drolet, F. m. c.; Fredrickson, G. H. Combinatorial Screening of Complex Block Copolymer Assembly with Self-Consistent Field Theory. *Phys. Rev. Lett.* **1999**, *83*, 4317–4320, DOI: 10.1103/PhysRevLett.83.4317.

(50) Guo, Z.; Zhang, G.; Qiu, F.; Zhang, H.; Yang, Y.; Shi, A.-C. Discovering Ordered Phases of Block Copolymers: New Results from a Generic Fourier-Space Approach. *Phys. Rev. Lett.* **2008**, *101*, 028301, DOI: 10.1103/PhysRevLett.101.028301.

(51) Xu, W.; Jiang, K.; Zhang, P.; Shi, A.-C. A Strategy to Explore Stable and Metastable Ordered Phases of Block Copolymers. *J. Phys. Chem. B* **2013**, *117*, 5296–5305, DOI: 10.1021/jp309862b, PMID: 23551204.

(52) Hsu, Y.-C.; Huang, C.-I.; Li, W.; Qiu, F.; Shi, A.-C. Micellization of linear A-b-(B-alt-C)n multiblock terpolymers in A-selective solvents. *Polymer* **2013**, *54*, 431–439, DOI: 10.1016/j.polymer.2012.11.005.

Graphical TOC Entry