

High-accurate and efficient numerical algorithms for the self-consistent field theory of liquid-crystalline polymers

Liwei Tan^{a,b,*}, Zhijuan He^{a,*}, Xin Wang^{a,*}, Kai Jiang^{a,**}

^aHuman Key Laboratory for Computation and Simulation in Science and Engineering, Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education, School of Mathematics and Computational Science, Xiangtan University, Xiangtan, Hunan, 411105, China

^bSchool of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, P.R. China.

Abstract

In this paper, we develop and investigate numerical methods for the self-consistent field theory (SCFT) of liquid crystalline polymers. Both the Flory-Huggins interaction potential and the Maier-Saupe orientational interaction are considered, enabling simultaneous exploration of microphase separation and liquid crystalline order in these systems. The main challenge in numerically solving this complex system lies in solving 6-dimensional (3D spatial + 2D orientation + 1D time) partial differential equations and performing nonlinear iterations to optimize the fields. We present ten numerical algorithms for solving high-dimensional PDEs and give an evaluation criterion of choosing the most appropriate PDE solver both from theoretical and numerical performance. Results demonstrate that coupling the fourth-order Runge-Kutta scheme with Fourier pseudo-spectral methods and spherical harmonic transformations is superior compared to other algorithms. We develop two nonlinear iteration schemes, alternating direction iteration method and Anderson mixing method, and design an adaptive technique to enhance the stability of the Anderson mixing method. Moreover, the cascadic multi-level (CML) technique further accelerates the SCFT computation when applied to these iteration methods. Meanwhile, an approach to optimize the computational domain is developed. To demonstrate the power of our approaches, we investigate the self-assembled behaviors of flexible-semiflexible diblock copolymers in 4, 5, 6-dimensional simulations. Numerical results illustrate the efficiency of our proposed method.

Keywords: Self-consistent field theory; Semiflexible polymer chain; Maier-Saupe orientational interaction; High-dimensional partial differential equation; Nonlinear iteration; Optimizing computational domain.

1. Introduction

The liquid crystalline polymeric systems have attracted much attention due to their potential industrial applications dependent on customized microstructures [1]. Theoretical studies allow us better understand and predict the phase behavior of liquid crystalline polymers [2, 3]. Among various theories, the self-consistent field theory (SCFT) has been proven to be a powerful tool in the study of phase behaviors of polymeric systems [4, 5]. Many SCFT studies mainly focus on flexible polymers modeled by continuous Gaussian chain and achieve great success [6, 7]. However, the rigidity of liquid crystal polymer chains heavily affects the phase behaviors of polymeric systems [8–14]. Therefore, a thorough investigation of numerical algorithms for SCFT of liquid crystalline polymeric systems is essential.

Liquid crystalline polymer systems typically include semiflexible subchains, which could be modeled by the wormlike chain model. The wormlike chain model [15] can effectively describe the orientation and density distribution of semiflexible subchains, which results in the emergence of 6-dimensional (6D) partial differential equations (PDEs). Meanwhile, the commonly used long-range interaction in liquid crystals, such as the Maier-Saupe interaction, is introduced into the semiflexible chains to promote parallel alignment between rigid blocks. However, such orientational interactions may lead to a singularity phenomenon, causing a slowdown in the SCFT iteration process. Jiang et al. [10, 16] established a theoretical basis and numerical scheme for studying the semiflexible diblock copolymers. They employed a third-order backward differentiation formula in the time direction. However, they did not consider the Maier-Saupe interaction, resulting in the absence of the liquid crystal order. Considering the Maier-Saupe interaction, Liang et al. [17] used Fourier pseudo-spectral method and spherical harmonic

*These authors contributed equally to this work.

**Corresponding author. Email: kaijiang@xtu.edu.cn

expansion to discretization spatial and directional variables in flexi-semiflexible diblock copolymers respectively, and used second-order operator splitting method to discretization time variables. Meanwhile, they proposed two effective iterative schemes to solve the SCFT equation. However, these proposed methods in their work are still insufficient to address the computational challenges arising from the high dimensionality and singularity present in current calculations.

In this work, we establish the SCFT model with Maier-Saupe interaction, using a linear diblock flexi-semiflexible molecules as an example, and design a series of algorithms to address the computational challenges arising from high dimensionality and singularity in the model. When solving PDEs, discretizing high-dimensional PDEs result in unaffordable computational burden. High-order, high-precision numerical algorithms are an appropriate method to alleviate this issue. For spatial functions, under periodic boundary conditions, we employ Fourier pseudo-spectral methods with exponential convergence [3, 10, 16, 18]. For orientation functions, we use spherical harmonic transformations, which also exhibit exponential convergence [3, 4, 10, 11, 16]. Various time discretization schemes can be applied, such as Runge-Kutta type schemes, [10, 16], operator-splitting scheme [3, 18]. Meanwhile, spectral deferred correction methods [19] can be employed to enhance the accuracy of discretization. Considering the convection-diffusion characteristics of high-dimensional PDEs, combining finite difference methods with upwind schemes is also an option [5]. To alleviate the slow convergence of SCFT iterations caused by Maier-Saupe interactions, a stable and rapidly converging iterative method is required. The design of nonlinear iterative methods for updating field depends on the mathematical properties of SCFT. Gradient-based iterative methods can be proposed to update field functions when the descent and ascent directions of the free energy can be identified, such as explicit Euler schemes [20, 21], semi-implicit scheme [20–22], and hybrid conjugate gradient method [17, 21]. Anderson mixing methods can be employed to accelerate iterations if SCFT is regarded as a fixed-point problem. However, there is currently no definitive standard for evaluating the efficiency of these iterative methods in SCFT iterations. In this work, we develop adaptive Anderson mixing methods and cascading multi-level methods to accelerate SCFT iterations. Moreover, we analyze the effectiveness, accuracy, and computational complexity of the proposed algorithms in this paper.

Our contribution in this paper include: a) providing an evaluation criteria to choose PDE solver from approximation accuracy, computational complexity and numerical performance by exploring, analyzing and implementing ten numerical methods for solving high-dimensional PDEs, b) presenting several effective iterative schemes, developing an adaptive Anderson mixing method and incorporating the cascading multi-level method to accelerate the existing iterative methods, c) developing an efficient approach to optimize computational domain automatically.

The present paper is organized as follows. In Sec. 2, we present a brief description of the SCFT model for the flexible-semiflexible diblock copolymers. In Sec. 3, we introduce ten time discretization schemes designed to solve high-dimensional semiflexible propagators. Meanwhile, we present three iteration approaches for updating fields, and incorporate a cascading multi-level method to accelerate the SCFT iteration. Furthermore, we give an approach to automatically optimize the computational box of ordered structures during the iteration process. In Sec. 4, we present numerical results to demonstrate the effectiveness of all proposed algorithm. In Sec. 5, we provide a comprehensive summary of the study.

2. Self-consistent field theory



Figure 1: Schematic of the flexible-semiflexible chain containing a flexible block A (red) and a semiflexible block B (blue).

We consider an incompressible melt consisting of n flexible-semiflexible diblock copolymers in a volume of V . Each flexible-semiflexible diblock copolymer, with a degree of polymerization N , consists of two chemically distinct monomers A and B , as schematically shown in Fig. 1. The number of monomers of the α block ($i \in \{A, B\}$) is denoted by $N_\alpha = f_\alpha N$, where f_α is the volume fraction. It is noted that $f_A + f_B = 1$, $N_A + N_B = N$. The statistical segment lengths of monomers are b_α . We employ the Gaussian chain model and the wormlike chain model to describe flexible and semiflexible blocks, respectively [3]. The interaction between the monomers A and

B is measured by a Flory-Huggins parameter χ , whereas the orientation interaction between semiflexible blocks is depicted by the Maier-Saupe parameter η . Within the SCFT framework [3, 11], the Hamilton H of flexible-semiflexible diblock copolymer is

$$\frac{H}{nk_B T} = \frac{1}{V} \int_V \left\{ \frac{1}{\chi N} \mu_-^2(\mathbf{r}) - \mu_+(\mathbf{r}) + \frac{1}{2\eta N} \mathbf{M}(\mathbf{r}) : \mathbf{M}(\mathbf{r}) \right\} d\mathbf{r} - \log Q, \quad (1)$$

where k_B is the Boltzmann constant, T is the temperature, Q is the single chain partition function, $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ is the tensor orientational field. The pressure-like field μ_+ corresponds to the local incompressibility. The exchange chemical field μ_- is conjugate to the local density difference between the polymer densities. The mean fields $\omega_A(\mathbf{r})$ and $\omega_B(\mathbf{r})$ are related to $\mu_+(\mathbf{r})$, $\mu_-(\mathbf{r})$,

$$\omega_A(\mathbf{r}) = \mu_+(\mathbf{r}) - \mu_-(\mathbf{r}), \quad \omega_B(\mathbf{r}) = \mu_+(\mathbf{r}) + \mu_-(\mathbf{r}). \quad (2)$$

The single chain partition function Q can be calculated as

$$\begin{aligned} Q &= \frac{1}{V} \int_V \int_S q_A(\mathbf{r}, f) q_B^\dagger(\mathbf{r}, \mathbf{u}, 1 - f) d\mathbf{u} d\mathbf{r}, \\ &= \frac{1}{V} \int_V \int_S q_B(\mathbf{r}, \mathbf{u}, s) q_B^\dagger(\mathbf{r}, \mathbf{u}, 1 - s) d\mathbf{u} d\mathbf{r}, \quad \forall s \in (f, 1], \\ &= \frac{1}{V} \int_V q_A(\mathbf{r}, s) q_A^\dagger(\mathbf{r}, 1 - s) d\mathbf{r}, \quad \forall s \in [0, f], \end{aligned} \quad (3)$$

where \mathbf{u} represents the local orientation of the semiflexible segment defined on a unit spherical surface \mathcal{S} . Flexible forward propagator $q_A(\mathbf{r}, s)$ describes the probability of finding the s -th ($0 \leq s \leq f$) segment at spatial position \mathbf{r} under the mean field $\omega_A(\mathbf{r})$. $q_A(\mathbf{r}, s)$ satisfies modified diffusion equation

$$\frac{\partial}{\partial s} q_A(\mathbf{r}, s) = \nabla_r^2 q_A(\mathbf{r}, s) - \omega_A(\mathbf{r}) q_A(\mathbf{r}, s), \quad s \in [0, f], \quad (4)$$

with initial condition $q_A(\mathbf{r}, 0) = 1$.

Semiflexible forward propagator $q_B(\mathbf{r}, \mathbf{u}, s)$ describes the probability of finding the s -th segment at spatial position \mathbf{r} with orientation \mathbf{u} ranging from $s = 1$ to $s = 1 - f$ under the mean field $\omega_B(\mathbf{r})$. $q_B(\mathbf{r}, \mathbf{u}, s)$ satisfies the ‘‘convection diffusion’’ equation

$$\begin{aligned} \frac{\partial}{\partial s} q_B(\mathbf{r}, \mathbf{u}, s) &= -\beta \mathbf{u} \cdot \nabla_r q_B(\mathbf{r}, \mathbf{u}, s) - \left(\omega_B(\mathbf{r}) - \mathbf{M}(\mathbf{r}) : \left[\mathbf{u}\mathbf{u} - \frac{1}{3} \mathbf{I} \right] \right) q_B(\mathbf{r}, \mathbf{u}, s) \\ &+ \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B(\mathbf{r}, \mathbf{u}, s), \quad s \in (f, 1], \end{aligned} \quad (5)$$

with initial condition $q_B(\mathbf{r}, \mathbf{u}, f) = q_A(\mathbf{r}, f)/4\pi$. $\beta = \sqrt{6}/v$, $v = b_A/(b_B \sqrt{N})$ measures the asymmetry of two monomers, λ is the stiffness parameter. Similarly, the backward semiflexible propagator $q_B^\dagger(\mathbf{r}, \mathbf{u}, s)$ satisfies

$$\begin{aligned} \frac{\partial}{\partial s} q_B^\dagger(\mathbf{r}, \mathbf{u}, s) &= \beta \mathbf{u} \cdot \nabla_r q_B^\dagger(\mathbf{r}, \mathbf{u}, s) - \left(\omega_B(\mathbf{r}) - \mathbf{M}(\mathbf{r}) : \left[\mathbf{u}\mathbf{u} - \frac{1}{3} \mathbf{I} \right] \right) q_B^\dagger(\mathbf{r}, \mathbf{u}, s) \\ &+ \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B^\dagger(\mathbf{r}, \mathbf{u}, s), \quad s \in [0, 1 - f], \end{aligned} \quad (6)$$

with initial condition $q_B^\dagger(\mathbf{r}, \mathbf{u}, 0) = 1/4\pi$. Backward flexible propagator $q_A^\dagger(\mathbf{r}, s)$ satisfies

$$\frac{\partial}{\partial s} q_A^\dagger(\mathbf{r}, s) = \nabla_r^2 q_A^\dagger(\mathbf{r}, s) - \omega_A(\mathbf{r}) q_A^\dagger(\mathbf{r}, s), \quad s \in (1 - f, 1], \quad (7)$$

with initial condition $q_A^\dagger(\mathbf{r}, 1 - f) = \int_S q_B^\dagger(\mathbf{r}, \mathbf{u}, 1 - f) d\mathbf{u}$.

The SCFT equations obtained from the first-order variational derivative of the free energy with respect to field functions are

$$\begin{aligned}
\phi_A(\mathbf{r}) &= \frac{1}{Q} \int_0^f q_A(\mathbf{r}, s) q_A^\dagger(\mathbf{r}, 1-s) ds, \\
\phi_B(\mathbf{r}) &= \frac{4\pi}{Q} \int_f^1 \int_S q_B(\mathbf{r}, \mathbf{u}, s) q_B^\dagger(\mathbf{r}, \mathbf{u}, 1-s) d\mathbf{u} ds, \\
\mathbf{S}(\mathbf{r}) &= \frac{4\pi}{Q} \int_f^1 \int_S q_B(\mathbf{r}, \mathbf{u}, s) \left(\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I} \right) q_B^\dagger(\mathbf{r}, \mathbf{u}, 1-s) d\mathbf{u} ds, \\
\phi_A(\mathbf{r}) + \phi_B(\mathbf{r}) - 1 &= 0, \\
\frac{2}{\chi N} \mu_-(\mathbf{r}) - [\phi_A(\mathbf{r}) - \phi_B(\mathbf{r})] &= 0, \\
\frac{1}{\eta N} \mathbf{M}(\mathbf{r}) - \mathbf{S}(\mathbf{r}) &= 0,
\end{aligned} \tag{8}$$

where $\phi_\alpha(\mathbf{r})$ ($\alpha \in A, B$) and $\mathbf{S}(\mathbf{r})$ are the density of the α -block and the orientational order parameter, respectively. The standard saddle point iteration of solving SCFT equations includes

- Step 1: Give initial fields $\mu_+(\mathbf{r}), \mu_-(\mathbf{r}), \mathbf{M}(\mathbf{r})$ and computational domain \mathcal{B} .
- Step 2: Calculate propagators $q_A(\mathbf{r}, s), q_A^\dagger(\mathbf{r}, s), q_B(\mathbf{r}, \mathbf{u}, s)$ and $q_B^\dagger(\mathbf{r}, \mathbf{u}, s)$ from current fields.
- Step 3: Compute the single chain partition function Q , order parameters $\phi_A(\mathbf{r}), \phi_B(\mathbf{r}), \mathbf{S}(\mathbf{r})$ and evaluate the free energy H .
- Step 4: Output the converged results if the error ξ is less than a prescribed value, otherwise update fields $\mu_-(\mathbf{r}), \mu_+(\mathbf{r}), \mathbf{M}(\mathbf{r})$, go back to step 2, where $\xi = E_{\mu_+} + E_{\mu_-} + E_M$, $E_{\mu_+} = \left\| \frac{\delta H}{\delta \mu_+} \right\|$, $E_{\mu_-} = \left\| \frac{\delta H}{\delta \mu_-} \right\|$, $E_M = \sqrt{(\sum_{i,j=1}^3 \left\| \frac{\delta H}{\delta M_{i,j}} \right\|)^2} / 9$, and the norm $\|\mathbf{v}\|$, $\mathbf{v} = (v_1, v_2, \dots, v_n) \in \mathbb{R}^n$ is denoted by $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2} / n$.

3. Numerical methods

In this section, we systematically study the numerical methods of solving SCFT with liquid crystalline interaction potential based on the wormlike chain model. These numerical methods mainly include four parts. Firstly, we introduce ten time discretization schemes for solving high-dimensional partial differential equations (5) and (6), along with corresponding spatial and orientational discretization methods. We will compare these methods based on computational complexity analysis, numerical accuracy, and numerical performance to recommend the relatively optimal algorithm. Secondly, we give several iteration approaches to update the field functions, including the alternative direction iteration (ADI) method, Anderson mixing (AM) method and the adaptive Anderson mixing (AAM) method. Thirdly, we introduce a cascadic multi-level (CML) method to accelerate the SCFT iteration, which has been used in SCFT calculation for flexible polymeric systems by Ceniceros and Fredrickson [20]. Finally, we propose an optimization method to automatically obtain the optimal computational domain.

3.1. Discretization scheme of high-dimensional PDEs for semiflexible propagators

The primary numerical challenge in SCFT iterations is the solving of semiflexible propagators. Flexible propagators satisfy modified diffusion equations with a d -dimensional spatial variable ($d = 2, 3$) and a 1-dimensional time variable. This can be efficiently addressed by using the Fourier pseudo-spectral method and a fourth-order backward difference formula scheme [10, 23, 24]. Solving semiflexible propagators involve discretizing high-dimensional ‘‘convection-diffusion’’ equations with a 3-dimensional spatial, a 2-dimensional orientational, and a 1-dimensional time variable. Due to the addition of orientation in semiflexible propagators, compared to flexible propagators, this presents a significant computational challenge. The goal of this section is to explore the most efficient numerical methods for solving semiflexible propagators.

In this section, we first introduce ten time discretization schemes, which can be classified into four groups, Runge-Kutta (RK) type schemes, backward difference formula (BDF) type schemes, operator splitting (OS) schemes, and upwind type schemes. Furthermore, we employ the spectral deferred correction (SDC) method [25] for the time variable to enhance the accuracy. Under the periodic boundary conditions, the Fourier pseudo-spectral method with exponential convergence is used to discretize spatial variables. Specifically, the finite difference

scheme is employed for spatial discretization when the upwind scheme is used for time discretization. The orientational variable \mathbf{u} is discretized by the spherical harmonic expansion. Before presenting these numerical methods, Eqn. (5) of the semiflexible propagator $q_B(\mathbf{r}, \mathbf{u}, s)$ can be written as

$$\begin{aligned}\frac{\partial}{\partial s} q_B(\mathbf{r}, \mathbf{u}, s) &= \mathcal{L}_1 q_B(\mathbf{r}, \mathbf{u}, s) + \mathcal{L}_2 q_B(\mathbf{r}, \mathbf{u}, s) + \mathcal{L}_3 q_B(\mathbf{r}, \mathbf{u}, s), \\ \mathcal{L}_1 q_B(\mathbf{r}, \mathbf{u}, s) &:= -\beta \mathbf{u} \cdot \nabla_{\mathbf{r}} q_B(\mathbf{r}, \mathbf{u}, s), \\ \mathcal{L}_2 q_B(\mathbf{r}, \mathbf{u}, s) &:= -(w_B(\mathbf{r}) - \mathbf{M}(\mathbf{r}) : [\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I}]) q_B(\mathbf{r}, \mathbf{u}, s), \\ \mathcal{L}_3 q_B(\mathbf{r}, \mathbf{u}, s) &:= \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B(\mathbf{r}, \mathbf{u}, s).\end{aligned}\tag{9}$$

To determine the most suitable method among these algorithms, we conduct an analysis of the computational complexity and time discretization accuracy for each method. Here, we present the semi-discrete scheme of the semiflexible propagator, and a fully discrete scheme is exemplified in Appendix C. Tab. 1 provides symbols and annotations necessary for illustrating numerical methods.

Table 1: Notations of discretization variables

Notation	Explanation
N_r	the number of discretization nodes of spatial variable \mathbf{r}
N_s	the number of discretization nodes of time variable s
N_θ	the number of discretization nodes of angle θ related to orientation \mathbf{u}
N_φ	the number of discretization nodes of angle φ related to orientation \mathbf{u}
Δs	the time s discretization step size
Δh	the space \mathbf{r} discretization step size

In practice, the Fast Fourier Transformation (FFT) is used in Fourier pseudo-spectral method, while the SPHEREPACK 3.0 [26] package is used to implement the fast spherical harmonic transformation and its inverse transformation. The computation complexity of FFT, inverse Fast Fourier Transformation (IFFT), Discrete Cosine Transform (DCT), inverse Discrete Cosine Transform (IDCT) and Spherical Harmonic Transform (SHT) are listed in Tab. 2. The FFT and IFFT algorithms have the same computational complexity, *i.e.*, $m_{FFT} = m_{IFFT}$. Similarly, $m_{DCT} = m_{IDCT}$, $m_{SHT} = m_{ISHT}$.

Table 2: The computational complexity of several transformations

Algorithm	Computational complexity	Notation
FFT (IFFT)	$O(N_r \log N_r)$	m_{FFT} (m_{IFFT})
DCT (IDCT)	$O(N_s \log N_s)$	m_{DCT} (m_{IDCT})
SHT (ISHT)	$O(N_\theta^2 N_\varphi + N_\theta N_\varphi \log N_\varphi)$	m_{SHT} (m_{ISHT})

3.1.1. Runge-Kutta type methods

- *Third-order implicit-explicit (IMEX3) method*

The convection diffusion equations (9) can be expressed as

$$\frac{\partial}{\partial s} q_B(\mathbf{r}, \mathbf{u}, s) = g(q_B(\mathbf{r}, \mathbf{u}, s)) + f(q_B(\mathbf{r}, \mathbf{u}, s)),\tag{10}$$

where $g(q_B(\mathbf{r}, \mathbf{u}, s)) = \mathcal{L}_1(q_B(\mathbf{r}, \mathbf{u}, s))$ and $f(q_B(\mathbf{r}, \mathbf{u}, s)) = \mathcal{L}_2(q_B(\mathbf{r}, \mathbf{u}, s)) + \mathcal{L}_3(q_B(\mathbf{r}, \mathbf{u}, s))$. The IMEX3 [27] method discretizes $g(q_B(\mathbf{r}, \mathbf{u}, s))$ implicitly, and $f(q_B(\mathbf{r}, \mathbf{u}, s))$ explicitly.

Concretely, for g , we use a third-stage diagonally-implicit Runge-Kutta scheme [28] with coefficients $A = (a_{ij}) \in \mathbb{R}^{3 \times 3}$, $\mathbf{b} = (b_i) \in \mathbb{R}^3$, $\mathbf{c} = (c_i) \in \mathbb{R}^3$, $i, j = 1, 2, 3$. The specific values of these coefficients are listed below.

c_1	a_{11}	0	0	0.4358665	0.4358665	0	0
c_2	a_{21}	a_{22}	0	0.7179333	0.2820667	0.4358665	0
c_3	a_{31}	a_{32}	a_{33}	1	1.2084966	-0.6443632	0.4358665
	b_1	b_2	b_3		1.2084966	-0.6443632	0.4358665

For f , we adopt a fourth-stage explicit scheme with $\bar{\mathbf{c}} = (0)$ and coefficient matrices $\bar{A} = (\bar{a}_{ij}) \in \mathbb{R}^{4 \times 4}$, $i, j = 1, 2, 3, 4$, $\bar{\mathbf{b}} = (0, \mathbf{b}) \in \mathbb{R}^4$, as shown below.

0	\bar{a}_{11}	0	0	0	0	0	0	0	0
c_1	\bar{a}_{21}	\bar{a}_{22}	0	0	0.4358665	0.4358665	0	0	0
c_2	\bar{a}_{31}	\bar{a}_{32}	\bar{a}_{33}	0	0.7179333	0.3212789	0.3966544	0	0
c_3	\bar{a}_{41}	\bar{a}_{42}	\bar{a}_{43}	\bar{a}_{44}	1	-0.1058583	0.5529291	0.5529291	0
	0	b_1	b_2	b_3		0	1.2084966	-0.6443632	0.4358665

Let s_n denote the n -th discrete point in time, from s_{n-1} to $s_n = s_{n-1} + \Delta s$, applying the IMEX3 method, $q(\mathbf{r}, \mathbf{u}, s_n)$ can be simplified as $q^n(\mathbf{r}, \mathbf{u})$.

$$q_B^n(\mathbf{r}, \mathbf{u}) = q_B^{n-1}(\mathbf{r}, \mathbf{u}) + \Delta s \sum_{j=1}^3 b_j (K_j + \bar{K}_{j+1}). \quad (11)$$

Assume that $q_i(\mathbf{r}, \mathbf{u})$, $i = 1, 2, 3$, are the uniform interpolation points from s_{n-1} to s_n , then we have

$$K_i = g(q_i(\mathbf{r}, \mathbf{u})), \quad (12)$$

where

$$q_i(\mathbf{r}, \mathbf{u}) = q_B^{n-1}(\mathbf{r}, \mathbf{u}) + \Delta s \sum_{j=1}^i a_{i,j} K_j + \Delta s \sum_{j=1}^i \bar{a}_{i+1,j} \bar{K}_j. \quad (13)$$

and

$$\bar{K}_1 = f(q_B^{n-1}(\mathbf{r}, \mathbf{u})), \quad \bar{K}_{i+1} = f(q_i(\mathbf{r}, \mathbf{u})). \quad (14)$$

The computational complexity of the IMEX3 method is

$$\begin{aligned} C_{\text{IMEX3}} &= 8N_s N_r m_{SHT} + 9N_\varphi N_\theta N_s m_{FFT} + 39N_\varphi N_r N_\theta N_s \\ &+ 16N_\varphi N_\theta N_s + 2N_\theta^2 + 2N_r N_\theta + 20N_\varphi N_\theta + 18N_\varphi N_r N_\theta. \end{aligned} \quad (15)$$

- *Fourth-order Runge-Kutta (RK4) method*

The RK4 method [28] can also be used to solve the Eqn.(9). Firstly, we rewrite the equation (9) into the following form,

$$\frac{\partial}{\partial s} q_B(\mathbf{r}, \mathbf{u}, s) = F(q_B(\mathbf{r}, \mathbf{u}, s)), \quad (16)$$

where $F(q_B(\mathbf{r}, \mathbf{u}, s)) = \mathcal{L}_1 q_B(\mathbf{r}, \mathbf{u}, s) + \mathcal{L}_2 q_B(\mathbf{r}, \mathbf{u}, s) + \mathcal{L}_3 q_B(\mathbf{r}, \mathbf{u}, s)$. Then, from s_{n-1} to $s_n = s_{n-1} + \Delta s$, we denote

$$\begin{aligned} K_1 &= F(q_B^{n-1}(\mathbf{r}, \mathbf{u})), \\ K_2 &= F(q_B^{n-1}(\mathbf{r}, \mathbf{u}) + \Delta s K_1 / 2), \\ K_3 &= F(q_B^{n-1}(\mathbf{r}, \mathbf{u}) + \Delta s K_2 / 2), \\ K_4 &= F(q_B^{n-1}(\mathbf{r}, \mathbf{u}) + \Delta s K_3). \end{aligned} \quad (17)$$

Finally, we obtain

$$q_B^n(\mathbf{r}, \mathbf{u}) = q_B^{n-1}(\mathbf{r}, \mathbf{u}) + \frac{\Delta s}{6} (K_1 + 2K_2 + 2K_3 + K_4). \quad (18)$$

The computational complexity of the RK4 method is

$$\begin{aligned} C_{\text{RK4}} &= 8N_s N_r m_{SHT} + 8N_\varphi N_\theta N_s m_{FFT} + 25N_\varphi N_r N_\theta N_s \\ &+ 2N_\theta^2 + 2N_r N_\theta + 20N_\varphi N_\theta + 26N_\varphi N_r N_\theta. \end{aligned} \quad (19)$$

3.1.2. Backward difference formula (BDF) type methods

The BDF-type approaches, as one of the linear multistep methods, have been commonly utilized in SCFT calculations [10, 23, 24]. Here, we also apply this method to solve Eqn. (9).

- *Third-order BDF (BDF3) scheme*

The BDF3 method has been used to solve semiflexible propagators [10], which can be expressed as

$$\begin{aligned} & \frac{11}{6}q_B^n(\mathbf{r}, \mathbf{u}) - 3q_B^{n-1}(\mathbf{r}, \mathbf{u}) + \frac{3}{2}q_B^{n-2}(\mathbf{r}, \mathbf{u}) - \frac{1}{3}q_B^{n-3}(\mathbf{r}, \mathbf{u}) \\ & = \Delta s \mathcal{L}_1 q_B^n(\mathbf{r}, \mathbf{u}) + \Delta s \left(3f^{n-1}(\mathbf{r}, \mathbf{u}) - 3f^{n-2}(\mathbf{r}, \mathbf{u}) + f^{n-3}(\mathbf{r}, \mathbf{u}) \right), \end{aligned} \quad (20)$$

where $f(\mathbf{r}, \mathbf{u}, s) = \mathcal{L}_2 q_B(\mathbf{r}, \mathbf{u}, s) + \mathcal{L}_3 q_B(\mathbf{r}, \mathbf{u}, s)$. In our implementation, initial values of the first three steps are obtained through the IMEX3 scheme. The computational complexity of the BDF3 method is

$$\begin{aligned} C_{\text{BDF3}} &= (2N_s + 16)N_r m_{\text{SHT}} + (2N_s + 12)N_\varphi N_\theta m_{\text{FFT}} + 14N_\varphi N_r N_\theta N_s \\ &+ N_\theta N_r N_s - N_r N_\theta + 20N_\varphi N_\theta + 60N_\varphi N_r N_\theta. \end{aligned} \quad (21)$$

- *Fourth-order BDF (BDF4) scheme*

The BDF4 method used to solve the Gaussian chain model [23] can also be applied to solve wormlike chain model, as illustrated below.

$$\begin{aligned} & \frac{25}{12}q_B^n(\mathbf{r}, \mathbf{u}) - 4q_B^{n-1}(\mathbf{r}, \mathbf{u}) + 3q_B^{n-2}(\mathbf{r}, \mathbf{u}) - \frac{4}{3}q_B^{n-3}(\mathbf{r}, \mathbf{u}) + \frac{1}{4}q_B^{n-4}(\mathbf{r}, \mathbf{u}) \\ & = \Delta s \mathcal{L}_1 q_B^n(\mathbf{r}, \mathbf{u}) + \Delta s \left(4f^{n-1}(\mathbf{r}, \mathbf{u}) - 6f^{n-2}(\mathbf{r}, \mathbf{u}) + 4f^{n-3}(\mathbf{r}, \mathbf{u}) - f^{n-4}(\mathbf{r}, \mathbf{u}) \right), \end{aligned} \quad (22)$$

where $f(\mathbf{r}, \mathbf{u}, s) = \mathcal{L}_2 q_B(\mathbf{r}, \mathbf{u}, s) + \mathcal{L}_3 q_B(\mathbf{r}, \mathbf{u}, s)$. Initial values of the first four steps can be obtained through the above-mentioned RK4 method. The computational complexity of the BDF4 method is

$$\begin{aligned} C_{\text{BDF4}} &= (2N_s + 24)N_r m_{\text{SHT}} + N_\varphi N_\theta (2N_s + 16)m_{\text{FFT}} + 18N_\varphi N_r N_\theta N_s \\ &+ 2N_\theta^2 + 8N_\varphi N_r N_s + 30N_r N_\theta + 20N_\varphi N_\theta + 53N_\varphi N_r N_\theta. \end{aligned} \quad (23)$$

3.1.3. Operator splitting (OS) method

The second-order operator splitting (OS2) method is a widely used technique for solving Gaussian propagator equations [18, 29]. For the semiflexible propagator equation (9), from s_{n-1} to $s_n = s_{n-1} + \Delta s$, the OS2 method can be expressed as [3],

$$q_B^n(\mathbf{r}, \mathbf{u}) = e^{\mathcal{L}_1 \Delta s / 2} e^{\mathcal{L}_2 \Delta s / 2} e^{\mathcal{L}_3 \Delta s} e^{\mathcal{L}_2 \Delta s / 2} e^{\mathcal{L}_1 \Delta s / 2} q_B^{n-1}(\mathbf{r}, \mathbf{u}). \quad (24)$$

The specific steps are as follows:

Step 1 Take FFT on $q_B^{n-1}(\mathbf{r}, \mathbf{u})$ to obtain $\hat{q}_B^{n-1}(\mathbf{k}, \mathbf{u})$, $\hat{q}_B^n(\mathbf{k}, \mathbf{u}) = e^{\mathcal{L}_1 \Delta s / 2} \hat{q}_B^{n-1}(\mathbf{k}, \mathbf{u})$,

Step 2 Perform IFFT on $\hat{q}_B^n(\mathbf{k}, \mathbf{u})$ to compute $q_B^n(\mathbf{r}, \mathbf{u})$, $q_B^n(\mathbf{r}, \mathbf{u}) = e^{\mathcal{L}_2 \Delta s / 2} q_B^n(\mathbf{r}, \mathbf{u})$,

Step 3 Apply SHT on $q_B^n(\mathbf{r}, \mathbf{u})$ to calculate $\hat{q}_{B_{lm}}^n(\mathbf{r})$, $\hat{q}_{B_{lm}}^n(\mathbf{r}) = e^{\mathcal{L}_3 \Delta s} \hat{q}_{B_{lm}}^n(\mathbf{r})$,

Step 4 Use ISHT on $\hat{q}_{B_{lm}}^n(\mathbf{r})$ to get $q_B^n(\mathbf{r}, \mathbf{u})$, $q_B^n(\mathbf{r}, \mathbf{u}) = e^{\mathcal{L}_2 \Delta s / 2} q_B^n(\mathbf{r}, \mathbf{u})$,

Step 5 Execute FFT for $q_B^n(\mathbf{r}, \mathbf{u})$ to obtain $\hat{q}_B^n(\mathbf{k}, \mathbf{u})$, $\hat{q}_B^n(\mathbf{k}, \mathbf{u}) = e^{\mathcal{L}_1 \Delta s / 2} \hat{q}_B^n(\mathbf{k}, \mathbf{u})$, perform IFFT on $\hat{q}_B^n(\mathbf{k}, \mathbf{u})$ to obtain $q_B^{n+1}(\mathbf{r}, \mathbf{u})$.

The computational complexity of the OS2 method is

$$\begin{aligned} C_{\text{OS2}} &= 2N_s N_r m_{\text{SHT}} + 4N_\varphi N_\theta N_s m_{\text{FFT}} + 5N_\varphi N_r N_\theta N_s \\ &+ 2N_\varphi N_r N_s + 2N_r N_\theta + 20N_\varphi N_\theta + 20N_\varphi N_r N_\theta. \end{aligned} \quad (25)$$

3.1.4. Upwind type schemes

Due to the nature of the semiflexible propagator equation being a convection-diffusion equation [5], a common approach to solving such equations is to use upwind schemes to handle the ‘‘convection’’ term, i.e., the spatial differential term. Taking 1-dimensional space as an example, explicit and implicit upwind schemes are provided. Similar methods are followed for two-dimensional and three-dimensional spaces.

- *Explicit upwind (EXUP) scheme*

If $\beta \mathbf{u} \geq 0$,

$$\frac{q_B^{n+1}(\mathbf{r}_j, \mathbf{u}) - q_B^n(\mathbf{r}_j, \mathbf{u})}{\Delta s} + \beta \mathbf{u} \frac{q_B^n(\mathbf{r}_j, \mathbf{u}) - q_B^n(\mathbf{r}_{j-1}, \mathbf{u})}{\Delta h} - \Gamma(\mathbf{r}_j, \mathbf{u}) q_B^n(\mathbf{r}_j, \mathbf{u}) - \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B^n(\mathbf{r}_j, \mathbf{u}) = 0, \quad (26)$$

otherwise $\beta \mathbf{u} < 0$,

$$\frac{q_B^{n+1}(\mathbf{r}_j, \mathbf{u}) - q_B^n(\mathbf{r}_j, \mathbf{u})}{\Delta s} + \beta \mathbf{u} \frac{q_B^n(\mathbf{r}_{j+1}, \mathbf{u}) - q_B^n(\mathbf{r}_j, \mathbf{u})}{\Delta h} - \Gamma(\mathbf{r}_j, \mathbf{u}) q_B^n(\mathbf{r}_j, \mathbf{u}) - \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B^n(\mathbf{r}_j, \mathbf{u}) = 0, \quad (27)$$

where $\Gamma(\mathbf{r}, \mathbf{u}) = -(w_B(\mathbf{r}) - \mathbf{M}(\mathbf{r}) : [\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I}])$. The computational complexity of the EXUP method is

$$C_{\text{EXUP}} = 2N_s N_r m_{\text{SHT}} + 7N_\varphi N_r N_\theta N_s + 2N_\theta^2 + 30N_\varphi N_\theta + 20N_\varphi N_r N_\theta + 2N_\theta. \quad (28)$$

• *Implicit upwind (IMUP) scheme*

If $\beta \mathbf{u} \geq 0$,

$$\frac{q_B^{n+1}(\mathbf{r}_j, \mathbf{u}) - q_B^n(\mathbf{r}_j, \mathbf{u})}{\Delta s} + \beta \mathbf{u} \frac{q_B^{n+1}(\mathbf{r}_j, \mathbf{u}) - q_B^{n+1}(\mathbf{r}_{j-1}, \mathbf{u})}{\Delta h} - \Gamma(\mathbf{r}_j, \mathbf{u}) q_B^n(\mathbf{r}_j, \mathbf{u}) - \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B^n(\mathbf{r}_j, \mathbf{u}) = 0, \quad (29)$$

otherwise $\beta \mathbf{u} < 0$,

$$\frac{q_B^{n+1}(\mathbf{r}_j, \mathbf{u}) - q_B^n(\mathbf{r}_j, \mathbf{u})}{\Delta s} + \beta \mathbf{u} \frac{q_B^{n+1}(\mathbf{r}_{j+1}, \mathbf{u}) - q_B^{n+1}(\mathbf{r}_j, \mathbf{u})}{\Delta h} - \Gamma(\mathbf{r}_j, \mathbf{u}) q_B^n(\mathbf{r}_j, \mathbf{u}) - \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B^n(\mathbf{r}_j, \mathbf{u}) = 0. \quad (30)$$

The computational complexity of the IMUP method is

$$C_{\text{IMUP}} = 2N_s N_r m_{\text{SHT}} + 20N_\varphi N_\theta + 3N_\theta + 2N_r + 2N_\theta^2 + \frac{11}{3} N_\varphi N_r N_\theta N_s + 18N_\varphi N_r N_\theta + \frac{1}{3} N_\varphi N_\theta N_s N_r^3. \quad (31)$$

3.1.5. Spectral deferred correction (SDC) method

The SDC method [25] can be used to improve the time discretization accuracy. The main idea of the SDC method is to use the spectral quadrature [19], such as a Gaussian or a Chebyshev-node interpolatory quadrature, to integrate the time derivative, which can achieve a highly-accuracy numerical solution with a largely reduced number of quadrature points. The SDC method has been used to solve flexible propagators [19]. Here we extend it to solve wormlike chain model. Before solving, an appropriate method is chosen to obtain an initial solution $q_B^{[0]}(\mathbf{r}, \mathbf{u}, s)$.

Firstly, the exact solution of semiflexible propagator can be given by integrating Eqn. (5) along the time variable s ,

$$q_B(\mathbf{r}, \mathbf{u}, s) = q_B(\mathbf{r}, \mathbf{u}, f) + \int_f^s \left[-\beta \mathbf{u} \cdot \nabla_r q_B(\mathbf{r}, \mathbf{u}, \tau) + \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B(\mathbf{r}, \mathbf{u}, \tau) - \left(w_B(\mathbf{r}) - \mathbf{M}(\mathbf{r}) : \left[\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I} \right] \right) q_B(\mathbf{r}, \mathbf{u}, \tau) \right] d\tau. \quad (32)$$

The error between the numerical solution $q_B^{[0]}(\mathbf{r}, \mathbf{u}, s)$ and the exact solution $q_B(\mathbf{r}, \mathbf{u}, s)$ is defined as

$$\delta^{[0]}(\mathbf{r}, \mathbf{u}, s) = q_B(\mathbf{r}, \mathbf{u}, s) - q_B^{[0]}(\mathbf{r}, \mathbf{u}, s), \quad (33)$$

which satisfies the integral equation

$$\delta^{[0]}(\mathbf{r}, \mathbf{u}, s) = \int_f^s \left[-\beta \mathbf{u} \cdot \nabla_r \delta^{[0]}(\mathbf{r}, \mathbf{u}, \tau) + \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 \delta^{[0]}(\mathbf{r}, \mathbf{u}, \tau) - \left(w(\mathbf{r}) - \mathbf{M}(\mathbf{r}) : \left[\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I} \right] \right) \delta^{[0]}(\mathbf{r}, \mathbf{u}, \tau) \right] d\tau + \epsilon^{[0]}(\mathbf{r}, \mathbf{u}, s). \quad (34)$$

where the residual

$$\epsilon^{[0]}(\mathbf{r}, \mathbf{u}, s) = q_B(\mathbf{r}, \mathbf{u}, f) + \int_f^s \left[-\beta \mathbf{u} \cdot \nabla_r q_B^{[0]}(\mathbf{r}, \mathbf{u}, \tau) + \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B^{[0]}(\mathbf{r}, \mathbf{u}, \tau) - \left(w_B(\mathbf{r}) - \mathbf{M}(\mathbf{r}) : \left[\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I} \right] \right) q_B^{[0]}(\mathbf{r}, \mathbf{u}, \tau) \right] d\tau - q_B^{[0]}(\mathbf{r}, \mathbf{u}, s). \quad (35)$$

Then taking the first derivative of Eqn. (34) with respect to s leads to

$$\begin{aligned} \frac{d\delta^{[0]}(\mathbf{r}, \mathbf{u}, s)}{ds} = & \left[-\beta\mathbf{u} \cdot \nabla_{\mathbf{r}}\delta^{[0]}(\mathbf{r}, \mathbf{u}, s) + \frac{1}{2\lambda}\nabla_{\mathbf{u}}^2\delta^{[0]}(\mathbf{r}, \mathbf{u}, s) - \left(w(\mathbf{r}) \right. \right. \\ & \left. \left. - \mathbf{M}(\mathbf{r}) : \left[\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I} \right] \right) \delta^{[0]}(\mathbf{r}, \mathbf{u}, s) \right] + \frac{d\epsilon^{[0]}(\mathbf{r}, \mathbf{u}, s)}{ds}. \end{aligned} \quad (36)$$

$\delta^{[0]}$ and $q_B^{[0]}$ are obtained by using the same numerical method. Then a corrected numerical solution of q_B is

$$q_B^{[1]}(\mathbf{r}, \mathbf{u}, s) = q_B^{[0]}(\mathbf{r}, \mathbf{u}, s) + \delta^{[0]}(\mathbf{r}, \mathbf{u}, s). \quad (37)$$

Finally, repeating the above processes, one can have $q_B^{[2]}, \dots, q_B^{[J]}$, J is the number of deferred corrections. The numerical accuracy of the deferred correction solution along the time variable is

$$\left\| q_B - q_B^{[J]} \right\| = O((\Delta s)^{m(J+1)}), \quad (38)$$

where Δs is the maximum distance of two adjacent Chebyshev-nodes, m is the order of the chosen numerical scheme to solve the Eqns. (5) and (36). In the following, we use the SDC method to improve the upwind type schemes and the OS2 method.

- *Explicit upwind with spectral deferred correction (EXUP+SDC) method*

We can use the first-order EXUP scheme with variable steps $\Delta s_i, i = 1 \dots n$ to solve $q_B^{[0]}(\mathbf{r}, \mathbf{u}, s)$ and $\delta^{[0]}(\mathbf{r}, \mathbf{u}, s)$. Subsequently, by applying the SDC method for one correction, we obtain the second-order EXUP+SDC method, and the computational complexity is

$$\begin{aligned} C_{\text{EXUP+SDC}} = & 6N_s N_r m_{\text{SHT}} + 2N_r N_\varphi N_\theta m_{\text{DCT}} + 30N_\varphi N_\theta \\ & + 33N_\varphi N_r N_\theta N_s + 20N_\varphi N_r N_\theta + 2N_\theta^2 + 7N_s - 1. \end{aligned} \quad (39)$$

- *Implicit upwind with spectral deferred correction (IMUP+SDC) method*

Similarly, $q_B^{[0]}(\mathbf{r}, \mathbf{u}, s)$ and $\delta^{[0]}(\mathbf{r}, \mathbf{u}, s)$ can be solved using a variable-step first-order IMUP method. we apply the SDC method for one correction to obtain the second-order solution with IMUP+SDC scheme. The corresponding computational complexity is

$$\begin{aligned} C_{\text{IMUP+SDC}} = & 6N_s N_r m_{\text{SHT}} + 2N_r N_\varphi N_\theta m_{\text{DCT}} + \frac{2}{3}N_\varphi N_\theta N_s N_r^3 \\ & + 20N_\varphi N_\theta + \frac{79}{3}N_\varphi N_r N_\theta N_s + 18N_\varphi N_r N_\theta + 3N_\theta + 2N_r + 2N_\theta^2. \end{aligned} \quad (40)$$

- *Second-order operator splitting with spectral deferred correction (OS2+SDC) method*

The OS2 scheme is employed to solve $q_B^{[0]}(\mathbf{r}, \mathbf{u}, s)$ and $\delta^{[0]}(\mathbf{r}, \mathbf{u}, s)$. Apply the SDC method for one correction, the resulting OS2+SDC method has fourth-order accuracy. The corresponding computational complexity is

$$\begin{aligned} C_{\text{OS2+SDC}} = & 6N_s N_r m_{\text{SHT}} + 2N_r N_\varphi N_\theta m_{\text{DCT}} + 10N_\varphi N_\theta N_s m_{\text{FFT}} \\ & + 15N_\varphi N_r N_\theta N_s + 4N_\varphi N_r N_s + 2N_r N_\theta + 20N_\varphi N_\theta + 20N_\varphi N_r N_\theta + 2N_\theta. \end{aligned} \quad (41)$$

Although SDC method can improve the efficiency of existing algorithms, its computational complexity and computation time are significantly increased. Therefore, we have not considered correcting other algorithms with higher computational complexity in this paper.

Finally, we summarize the dominant computational complexities of all methods, as listed in Tab. 3. One can sort the computational complexity of the ten schemes: $C_{\text{IMUP+SDC}} > C_{\text{IMUP}} > C_{\text{IMEX3}} > C_{\text{RK4}} > C_{\text{OS2+SDC}} > C_{\text{EXUP+SDC}} > C_{\text{BDF4}} > C_{\text{BDF3}} > C_{\text{OS2}} > C_{\text{EXUP}}$. Therefore, the EXUP method has the least computational amount, the IMUP+SDC method requires the largest computational cost.

Table 3: The dominant computational complexity of all proposed methods.

Algorithm	The dominant term of the computational complexity
EXUP	$2N_s N_r m_{SHT}$
EXUP+SDC	$6N_s N_r m_{SHT} + 2N_r N_\varphi N_\theta m_{DCT}$
IMUP	$2N_s N_r m_{SHT} + \frac{1}{3} N_\varphi N_\theta N_s N_r^3$
IMUP+SDC	$6N_s N_r m_{SHT} + 2N_r N_\varphi N_\theta m_{DCT} + \frac{2}{3} N_\varphi N_\theta N_s N_r^3$
IMEX3	$8N_s N_r m_{SHT} + 9N_\varphi N_\theta N_s m_{FFT}$
RK4	$8N_s N_r m_{SHT} + 8N_\varphi N_\theta N_s m_{FFT}$
BDF3	$(2N_s + 16)N_r m_{SHT} + N_\varphi N_\theta (2N_s + 12)m_{FFT}$
BDF4	$(2N_s + 24)N_r m_{SHT} + N_\varphi N_\theta (2N_s + 16)m_{FFT}$
OS2	$2N_s N_r m_{SHT} + 4N_s N_\varphi N_\theta m_{FFT}$
OS2+SDC	$6N_s N_r m_{SHT} + 2N_r N_\varphi N_\theta m_{DCT} + 10N_\varphi N_\theta N_s m_{FFT}$

3.2. Nonlinear iteration methods

The iterative methods for updating the fields depend on the mathematical structure of SCFT, and the inclusion of liquid crystalline interaction potentials leads to slow convergence of the iterations. To find an efficient and robust algorithm, we discuss several common nonlinear iteration methods, including the alternative direction iteration method, the Anderson mixing method and adaptive Anderson mixing algorithm.

3.2.1. Alternative direction iteration (ADI) method

In the flexible-semiflexible diblock copolymer system, the ordered structure corresponding to a saddle point of Hamilton functional can be obtained by maximizing with respect to μ_+ , and simultaneously minimizing with respect to μ_- and \mathbf{M} . Therefore, we can update μ_+ along with the direction of $\frac{\delta H}{\delta \mu_+}$, while update μ_- and \mathbf{M} along with the directions of $-\frac{\delta H}{\delta \mu_-}$ and $-\frac{\delta H}{\delta \mathbf{M}}$. Concretely, the ADI method [30] to update fields can be written as

$$\frac{\mu_+^{j+1} - \mu_+^j}{\Delta t_1} = (\phi_A^j + \phi_B^j - 1), \quad (42a)$$

$$\frac{\mu_-^{j+1} - \mu_-^j}{\Delta t_2} = -\left(\frac{2}{\chi N} \mu_-^j - (\phi_A^j - \phi_B^j)\right), \quad (42b)$$

$$\frac{\mathbf{M}^{j+1} - \mathbf{M}^j}{\Delta t_3} = -\left(\frac{1}{\eta N} \mathbf{M}^j - \mathbf{S}^j\right), \quad (42c)$$

where Δt_i , $i = 1, 2, 3$ is the iteration step length.

3.2.2. Anderson mixing (AM) approach

AM method was originally proposed by Anderson to solve nonlinear integral equation [31], which makes full use of the historical information of the previous m step to accelerate convergence. Recently, the Anderson mixing method has been widely used in the polymeric SCFT calculation [32–34]. It has greatly accelerated convergence in SCFT calculation of flexible systems. In this work, we try to discuss its performance for the semiflexible polymeric systems with the liquid-crystal interaction potential. Considering a fixed point $\psi(\mathbf{x}) = \mathbf{x}$, the AM method updates the field functions $\mathbf{x} = \{\mu_+, \mu_-, \mathbf{M}\}$ by applying historical information from prior m steps. Algorithm 1 presents the implementation of the Anderson mixing method.

To avoid the instability and improve the convergence speed of AM method, we propose an adaptive AM method. The main idea of the adaptive AM method is to employ the ADI method within the AM framework when an increase in error ξ is detected. Compared with the traditional AM method, this strategy efficiently avoids ineffective iterations and realigns the convergence trajectory towards the optimal path. Details of the adaptive AM are given in Algorithm 2.

3.3. Cascadic multi-level (CML) method

The CML method is an efficient acceleration technique for nonlinear iterations, which was originally introduced to SCFT calculations for flexible polymeric chain systems by Cenicerros and Fredrickson [20]. The idea behind the CML iteration is to rapidly reduce high-frequency errors on fine grids and eliminate low-frequency errors on coarse grids simultaneously through smoothing or relaxation methods. Here we extend the CML strategy to flexible-semiflexible systems.

Algorithm 1: Anderson mixing algorithm

Input: \mathbf{x}_0, ψ, m .

- 1 $\mathbf{x}_1 = \psi(\mathbf{x}_0), F_0 = \psi(\mathbf{x}_0) - \mathbf{x}_0$,
- 2 **for** $k = 1, 2, \dots, n - 1$ **do**
- 3 $m_k = \min(m, k)$,
- 4 $F_k = \psi(\mathbf{x}_k) - \mathbf{x}_k$,
- 5 Minimize $\left\| \sum_{j=0}^{m_k} \alpha_j^{(k)} F_{k-m_k+j} \right\|$ subject to $\sum_{j=0}^{m_k} \alpha_j^{(k)} = 1$,
- 6 $\mathbf{x}_{k+1} = (1 - \beta_k) \sum_{j=0}^{m_k} \alpha_j^{(k)} \mathbf{x}_{k-m_k+j} + \beta_k \sum_{j=0}^{m_k} \alpha_j^{(k)} \psi(\mathbf{x}_{k-m_k+j})$, where β_k is the weight.
- 7 **end**

Output: converged value \mathbf{x}_n .

Algorithm 2: Adaptive Anderson mixing algorithm

Input: \mathbf{x}_0, ψ, m .

- 1 **for** $k = 1, 2, \dots, n - 1$ **do**
- 2 carry out Alg. 1
- 3 **if** $\xi_k > \xi_{k-1}$ **then**
- 4 use the ADI method
- 5 **end**
- 6 carry out Alg. 1
- 7 **end**

Output: converged value \mathbf{x}_n .

Keep the symbols in Tab. 1, and we define $(N_r, N_\theta, N_\varphi, N_s)$ as the finest grid level. One can choose the coarsest level $(N_r^0, N_\theta^0, N_\varphi^0, N_s^0)$ such that $(N_r, N_\theta, N_\varphi, N_s) = (2^L N_r^0, 2^L N_\theta^0, 2^L N_\varphi^0, 2^L N_s^0)$, where L is an integer. Then we can obtain $L + 1$ level grids:

$$(N_r^i, N_\theta^i, N_\varphi^i, N_s^i) = (2^i N_r^0, 2^i N_\theta^0, 2^i N_\varphi^0, 2^i N_s^0), \quad i = 0, 1, \dots, L,$$

with $(N_r^L, N_\theta^L, N_\varphi^L, N_s^L) = (N_r, N_\theta, N_\varphi, N_s)$. On each level i , let \mathcal{W}^i be the initial fields at a given grid $(N_r^i, N_\theta^i, N_\varphi^i, N_s^i)$, ξ_i denote the error tolerance and \mathcal{W}_c^i be the convergence fields. With these notations, the CML method can be described simply as the algorithm 3.

Algorithm 3: Cascadic multi-level method

Input: Extract the coarsest grid fields \mathcal{W}^0 from the current fields.

- 1 **for** $k = 0, 1, 2, \dots, L - 1$ **do**
- 2 **if** $Error > \xi_i$ **then**
- 3 continue SCFT iteration.
- 4 **end**
- 5 **else**
- 6 obtain the fields \mathcal{W}_c^i on the level i .
- 7 **end**
- 8 obtain the fields \mathcal{W}^{i+1} on the level $i + 1$ by interpolating \mathcal{W}_c^i ,
- 9 **end**

Output: Convergent fields \mathcal{W}_c^L on the level L .

In algorithm 3, efficient interpolation involves initially zero-padding the Fourier coefficients of \mathcal{W}^{i-1} to obtain the Fourier coefficients of \mathcal{W}^i on a finer grid $(2N_r^i, 2N_\theta^i, 2N_\varphi^i, 2N_s^i)$, followed by executing IFFT to obtain \mathcal{W}^i .

3.4. Optimizing computational domain

The algorithm previously presented focuses on finding the saddle point of the free energy in a fixed computational domain. Note that Each crystal structure has its own periodicity. Finding the optimal domain for each structure is crucial in SCFT calculations. Because the free energy H is functional with respect to the calculation

domain, the optimization can be achieved by minimizing the free energy relative to the computational domain. Denoting the computational domain as \mathcal{B} , the first-order optimal condition of finding the optimum \mathcal{B} is

$$\left\| \frac{\partial H[\mathcal{B}]}{\partial \mathcal{B}} \right\|_{\ell^\infty} = \max \left| \frac{\partial H[\mathcal{B}]}{\partial b_{ij}} \right| = 0, \quad i, j = 1, \dots, d. \quad (43)$$

In practice, we always choose a proper coordinate system such that $b_{ij} = 0$ when $1 \leq i < j \leq d$. Using this strategy, the variables are required to be optimized reduce to $d(d+1)/2$. There are many optimization methods to solve the unconstrained minimization problem [35]. In this work, we use the Barzilai-Borwein method [36] to update the computational box \mathcal{B} . When the computational domain is adjusted at k -th time, the quantities are marked by the superscript k . For each entry b_{ij} ,

$$b_{ij}^{k+1} = b_{ij}^k - \alpha^k \frac{\partial H[\mathcal{B}^k]}{\partial b_{ij}^k}, \quad (44)$$

where $1 \leq i < j \leq d$. Eqns. (44) can be written as the matrix form

$$\mathcal{B}^{k+1} = \mathcal{B}^k - D^k \frac{\partial H[\mathcal{B}^k]}{\partial \mathcal{B}^k}, \quad (45)$$

where $D^k = \alpha^k I$. To make D^k have the quasi-Newton property [37], we compute α^k such that

$$\alpha^k = \underset{\alpha}{\operatorname{argmin}} \| s^{k-1} - D y^{k-1} \|_{\ell^\infty}, \quad (46)$$

where

$$s^{k-1} = \mathcal{B}^k - \mathcal{B}^{k-1}, \quad y^{k-1} = \frac{\partial H[\mathcal{B}^k]}{\partial \mathcal{B}^k} - \frac{\partial H[\mathcal{B}^{k-1}]}{\partial \mathcal{B}^{k-1}}.$$

Analytically calculating $\frac{\partial H[\mathcal{B}^k]}{\partial \mathcal{B}^k}$ is difficult, thus we compute it by the central difference method. The solution of the Eqn. (46) can be written as

$$\alpha^k = \frac{s^{k-1T} s^{k-1}}{s^{k-1T} y^{k-1}}. \quad (47)$$

Note that the proposed method can optimize the size and shape of the computational area automatically during the optimization process. The iterative procedure of finding the saddle point of the SCFT equations and optimizing the computational domain is summarized as

Step 1 Give an initial box \mathcal{B}^0 and the initial fields, set $k = 0$.

Step 2 Fix box \mathcal{B}^k , perform SCFT iteration until convergence, calculate the free energy $H[\mathcal{B}^k]$ and the derivative $\frac{\partial H[\mathcal{B}^k]}{\partial \mathcal{B}^k}$.

Step 3 If $\left\| \frac{\partial H}{\partial \mathcal{B}^k} \right\|_{\ell^\infty} > \epsilon_H$, compute the step size α^k and optimize \mathcal{B}^k by the Barzilai-Borwein method, set $k = k + 1$, go back to step 2, otherwise output $H[\mathcal{B}^k]$, \mathcal{B}^k .

The completed SCFT iteration procedure, including saddle point iteration, CML method, and optimal computational domain is presented in Fig. 2.

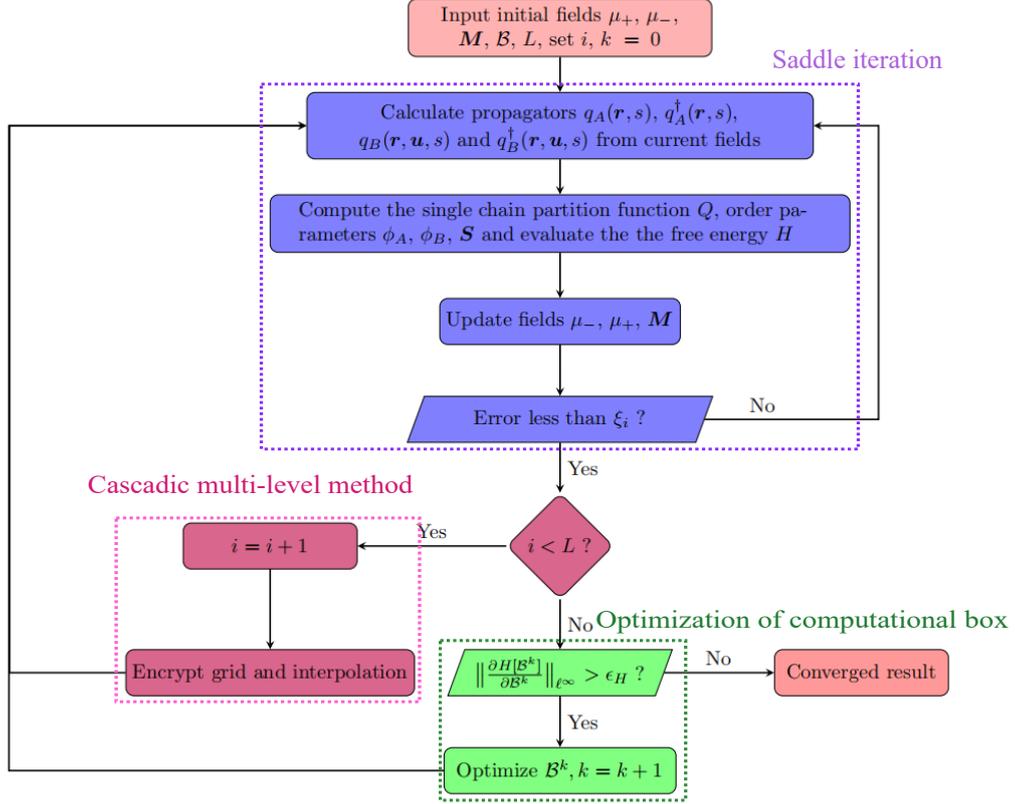


Figure 2: The flowchart of SCFT iteration process.

4. Numerical results and discussion

In this section, we present several numerical examples to demonstrate the performance of our proposed methods. The BDF4 scheme has been used to solve the flexible propagators. All calculations are executed on the CPU through the Intel(R) Xeon(R) Gold 6330 CPU with 28 cores.

4.1. Efficiency of time discretization schemes for high-dimensional PDEs

In this section, all numerical experiments take the case of 4D (space 1D + orientation 2D + time 1D) variables as an example, with parameters listed in Tab. 4 and initial values referenced from [17].

Table 4: The parameters on the 4D calculation.

χN	ηN	f	\mathcal{B}	β	N_r	N_θ	N_φ
16	8	0.6	1.31	2	64	8	17

Fixed spatial and orientational variables are shown in Tab. 4, ensuring accuracy in both spatial and orientational dimensions. To verify the numerical accuracy of ten time discretization schemes, we set $N_s = 20$ and successively refine by bisection, obtaining corresponding errors and ultimately determining the order of accuracy. The error is defined as $\|q^* - q_h\|$, where q^* represents the “numerically exact solution” obtained through the RK4 method with 2560 time discretization nodes, and q_h represents the numerical solution. One can find that the numerical errors and convergence orders for the ten time discretization schemes used to solve the semiflexible propagator in Tab. 5. All these order are consistent with theoretical results. Obviously, the errors of RK4 and OS2+SDC methods can reach 2×10^{-12} with an order of 4, making them the most accurate among all methods.

Table 5: The error and numerical order of ten numerical discretization schemes for solving the semiflexible propagators, with model parameter $\lambda = \infty$.

N_s	RK4		IMEX3		BDF4		BDF3		OS2	
	Error	Order								
20	1.37e-07	-	1.10e-06	-	7.96e-06	-	5.64e-05	-	1.21e-04	-
40	8.40e-09	4.02	1.38e-07	3.00	5.48e-07	3.86	7.43e-06	2.92	3.03e-05	1.99
80	5.18e-10	4.01	1.73e-08	2.99	3.51e-08	3.96	9.46e-07	2.97	7.57e-06	2.00
160	3.21e-11	4.01	2.16e-09	2.99	2.21e-09	3.98	1.18e-07	2.97	1.89e-06	2.00
320	2.00e-12	4.00	2.71e-10	2.99	1.39e-10	3.99	1.49e-08	2.99	4.71e-07	2.01
N_s	EXUP		IMUP		EXUP+SDC		IMUP+SDC		OS2+SDC	
	Error	Order								
20	2.13e-03	-	1.86e-03	-	2.13e-04	-	2.70e-05	-	1.39e-07	-
40	1.03e-03	1.04	9.19e-04	1.02	4.99e-05	2.09	7.01e-06	1.94	8.90e-09	3.97
80	5.05e-04	1.02	4.55e-04	1.01	1.18e-05	2.07	1.78e-06	1.97	5.59e-10	3.99
160	2.49e-04	1.02	2.25e-04	1.01	2.87e-06	2.04	4.50e-07	1.98	3.50e-11	3.99
320	1.22e-04	1.02	1.10e-04	1.02	7.05e-07	2.02	1.11e-07	2.00	2.21e-12	3.98

To further compare the performance of ten different time discretization methods, we apply them to SCFT calculations. The ADI method are chosen as the nonlinear iteration method to guarantee the stability of iteration process. Fig. 3 shows the iteration errors for solving semiflexible propagators using various time discretization schemes within 800 iteration steps. Notably, the EXUP method reduces the error to approximately $\mathcal{O}(10^{-6})$ in just 358 iterations. The RK4 scheme and the IMEX3 method follow closely, both requiring 402 iterations. Compared to other algorithms, these three methods converge the faster and achieve the lower errors.

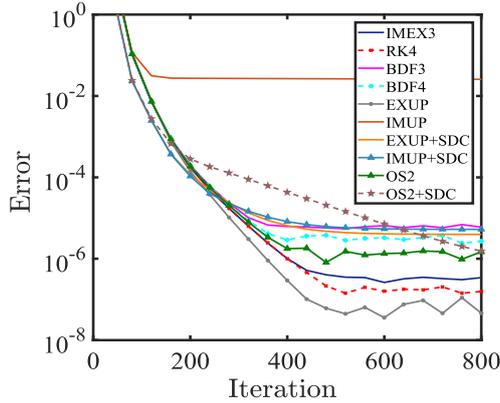


Figure 3: Comparison of ten time discretization schemes in Sec. 3.1, with model parameter $\lambda = 100$.

To further observe the converged accuracy of the three methods, Fig. 4 expresses that the changes of free energy H . Compared to the EXUP method, the RK4 and IMEX3 methods reach lower free energy H when free energy converges, demonstrating greater stability of these two numerical schemes in the SCFT system. From Tab. 3 and Tab. 5, RK4 method presents higher numerical accuracy and lower computational complexity compared to IMEX3. Therefore, we select RK4 method as the time discretization method for the semiflexible propagator.

4.2. Iteration efficiency

In this section, we focus on the effectiveness of nonlinear iterative algorithms. Firstly, we test the performances of these schemes as described in Sec. 3.2 using three iteration methods as follows: (i) ADI scheme, (ii) Anderson mixing method, and (iii) adaptive Anderson mixing method. Secondly, we use the CML technique to accelerate SCFT iterations. The numerical experiments are based on the 4-dimensional case (space 1D + orientation 2D + contour 1D), with parameters $N_r = 64$, $N_\theta = 16$, $N_\psi = 33$, $N_s = 200$. No specified, we set $\beta = 9.8$, $\eta N = 8$, $\lambda = 200$, $f = 0.6$, and $\chi N = 16$.

The convergence of ADI method, AM method and AAM method are shown in Fig. 5. One can find that three iterative methods can reach the saddle points in the SCFT iterations. Among these methods, the ADI method has the slowest converged rate, the AAM method converges fastest and has fewer iteration steps than the AM method

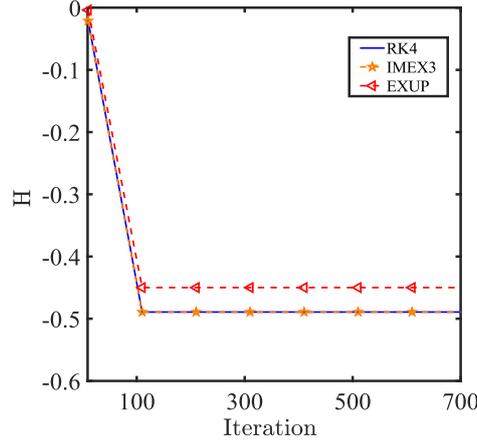


Figure 4: Numerical behavior of free energy H in the RK4 method, the IMEX3 method and the EXUP method, with the parameter $\lambda = 100$.

with parameter $m = 40$. Note that the value of m depends on the case. Specifically, the AAM method achieves an error of 1.7×10^{-7} in only 225 steps, while the AM and ADI methods require 360 steps to achieve errors of 5.3×10^{-7} and 1×10^{-6} , respectively. Therefore, the AAM method is currently the most recommended iterative scheme.

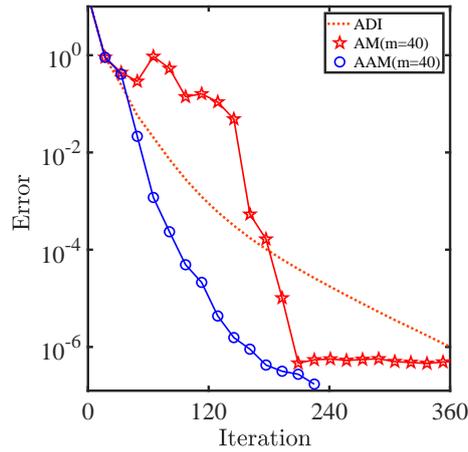


Figure 5: Comparison of the ADI method, AM method and AAM method, with parameters $\chi N = 16$, $\eta N = 8$, $f = 0.6$, $\lambda = 100$, $\beta = 2$, $N_s = 200$, $N_r = 32$, $N_\theta = 8$ and $N_\phi = 17$.

Then we use the CML method to accelerate the AAM method in SCFT iteration. Let the coarsest grid $(N_r^0, N_\phi^0, N_\theta^0, N_s^0)$ be $(16, 9, 4, 50)$ and three levels ($L = 3$) are used in this case. The error tolerance ξ in the three grid levels are set as 1×10^{-4} , 1×10^{-6} and 3×10^{-7} , respectively. Fig. 6 shows the SCFT iteration error for the AAM and its CML version (AAM-CML). At each grid change, there is a significant jump in the error, but it quickly drops to the required error tolerance, eventually achieving the same level of accuracy as AAM method. Although the AAM-CML method takes more steps, its CPU time is significantly reduced, saving 63.88% CPU time compared to AAM method, as shown in Tab. 6.

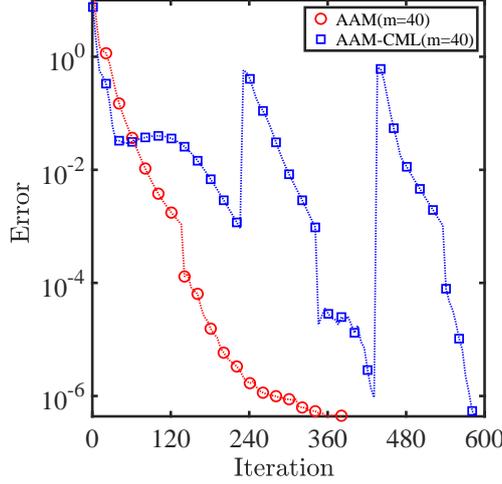


Figure 6: Comparison of the AAM method and the AAM-CML method, with parameters $\chi N = 16$, $\eta N = 8$, $f = 0.6$, $\lambda = 100$, $\beta = 2$, $N_s = 200$, $N_r = 32$, $N_\theta = 8$ and $N_\varphi = 17$.

Table 6: Comparison of the CPU time between the AAM and AAM-CML method.

AAM (t)	AAM-CML (t_c)	$1 - t_c/t$ (%)
4710.18s	1701.12s	63.88%

4.3. Ordered phases

In this section, we investigate self-assembled patterns of flexible-semiflexible diblock copolymers to demonstrate the effectiveness of our approaches in 4D (space 1D), 5D (space 2D), 6D (space 3D) (orientation 2D + time 1D) variables, respectively. The termination criterion for self-consistent field iterations is the total gradient difference $\xi < 1 \times 10^{-5}$ or the energy difference is less than 1×10^{-8} between two sequential iterations.

In the study of flexible-semiflexible diblock copolymers, both nematic and smectic phases have been observed in theoretical [38, 39] and experimental [40] researches. To demonstrate the effectiveness of our algorithm, we select the classic nematic and smectic-C phases as examples for initial values in 4D calculations.

Table 7: The parameters for nematic and smectic-C phases.

Phase	χN	ηN	f	λ	\mathcal{B}	β	N_r	N_θ	N_φ	N_s
Nematic	6	12	0.6	100	1.31	2	32	8	17	100
Smectic-C	16	8	0.6	100	1.31	2	32	8	17	100

Firstly, the model parameters used for the self-assembly of these two structures are listed in Tab. 7. Ordered phases obtained through our calculation are shown in Fig. 7. The density distributions $\phi_A = f$, $\phi_B = 1 - f$, and the orientation distributions $\phi_B(\mathbf{u}, z = 0)$ presents parallel alignment of semiflexible blocks. These characteristics satisfy the unique properties of the nematic phase, and its molecular schematic is shown in Fig. 7 (1c). The density distribution and the orientation distribution peaked at $\theta \in (0, \pi/2)$ in Fig. 7 (2a,2b) indicate that the ordered structure corresponds to the smectic-C phase. The molecular schematic of smectic-C is presented in Fig. 7 (2c).

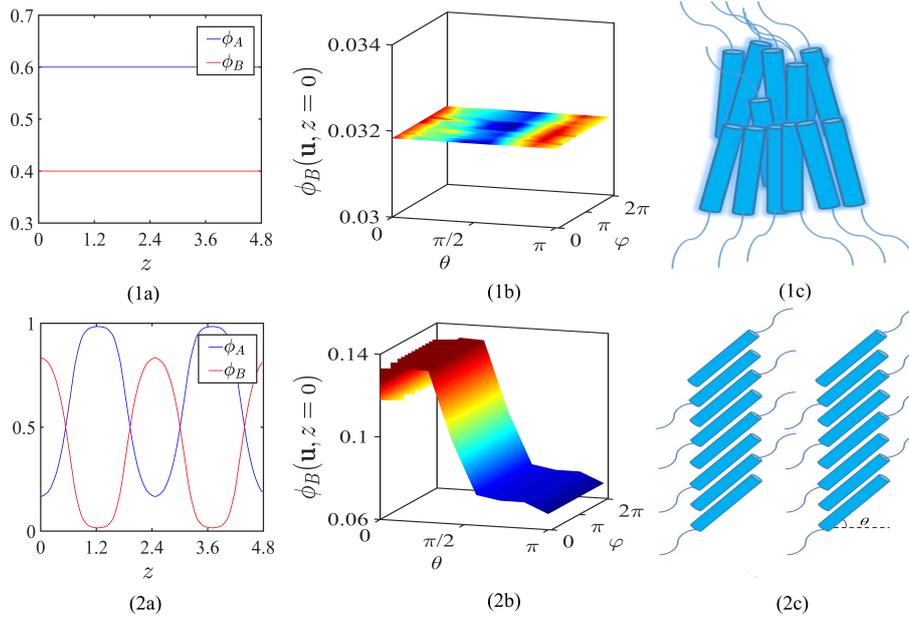


Figure 7: (1) Nematic phase, (2) smectic-C phase. The first, second, and third columns show the density distributions, orientation distribution, and schematic arrangement diagrams, respectively.

In 5D calculations, flexible-semiflexible diblock copolymers can self-assemble into various morphologies of columnar structures. Here we investigate $[4^4]$ and $[3^6]$ tilings with the Archimedean tiling naming rules. In our calculations, monomers B at higher concentrations form vertices, which are then connected to form polygonal tilings. Choosing a vertex as the center, we list the number of edges of each polygon connected to this vertex in a clockwise or counterclockwise order. For instance, $[4^4]$ tiling denotes a cyclic sequence of four squares. The "[]" are used to distinguish these tiling pattern naming from regular numbers. Here we not only reproduce $[4^4]$ and $[3^6]$ tilings but also discover a more complex $[3^4.4; 3^8]$ tiling, with model parameters detailed in Tab. 8. Furthermore, density distribution of monomer B , FFT patterns and orientation distributions are shown in Fig. 8.

Table 8: The parameters of $[4^4]$, $[3^6]$ and $[3^4.4; 3^8]$ tilings.

Phase	χN	ηN	f	λ	\mathcal{B}	β	N_r	N_θ	N_φ	N_s
$[4^4]$	20	10	0.75	300	$[0, 1.04]^2$	2	20	4	9	100
$[3^6]$	18	9	0.75	300	$[0, 0.52]^2$	2	20	4	9	100
$[3^4.4; 3^8]$	19	9.5	0.75	300	$[0, 0.78]^2$	2	20	4	9	100

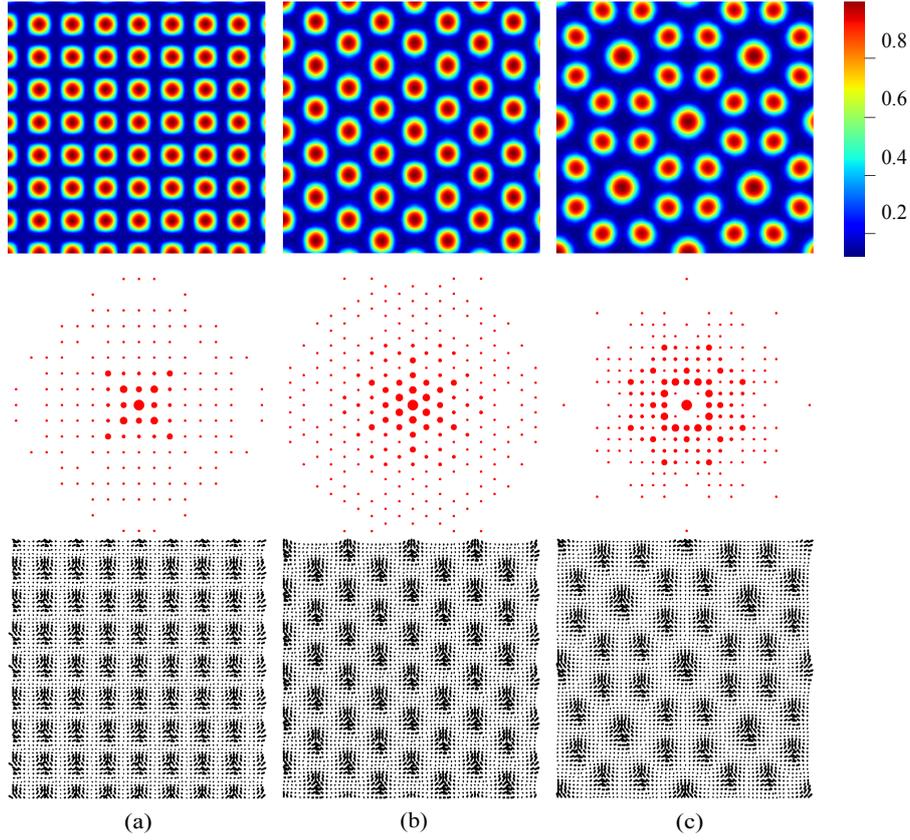


Figure 8: Density distributions of monomer B (first row), FFT patterns (second row), and orientations distributions of semiflexible chains (third row). (a) $[4^4]$, (b) $[3^6]$, (c) $[3^4.4; 3^8]$ tilings.

3D space cubic structures have been successfully predicted in rigid polymers [41, 42]. Based on this, we extend our calculations to 6D case. By selecting appropriate parameters (Tab. 9), we successfully simulate body-centred cubic (BCC), face-centred cubic (FCC), double gyroid (DG), and single gyroid (SG) phases. Their morphologies in 3D space, top views of the density distributions and orientation distributions of these phases are presented in Fig. 9.

Table 9: The parameters for ordered structures.

Phase	χN	ηN	f	λ	\mathcal{B}	β	N_r	N_θ	N_φ	N_s
DG	18	9	0.6	100	$[0, 1.31]^3$	2	16	8	17	100
FCC	14	14	0.7	100	$[0, 1.31]^3$	2	16	8	17	100
BCC	16	8	0.7	100	$[0, 1.31]^3$	2	16	8	17	100
SG	14	14	0.6	100	$[0, 1.31]^3$	2	16	8	17	100

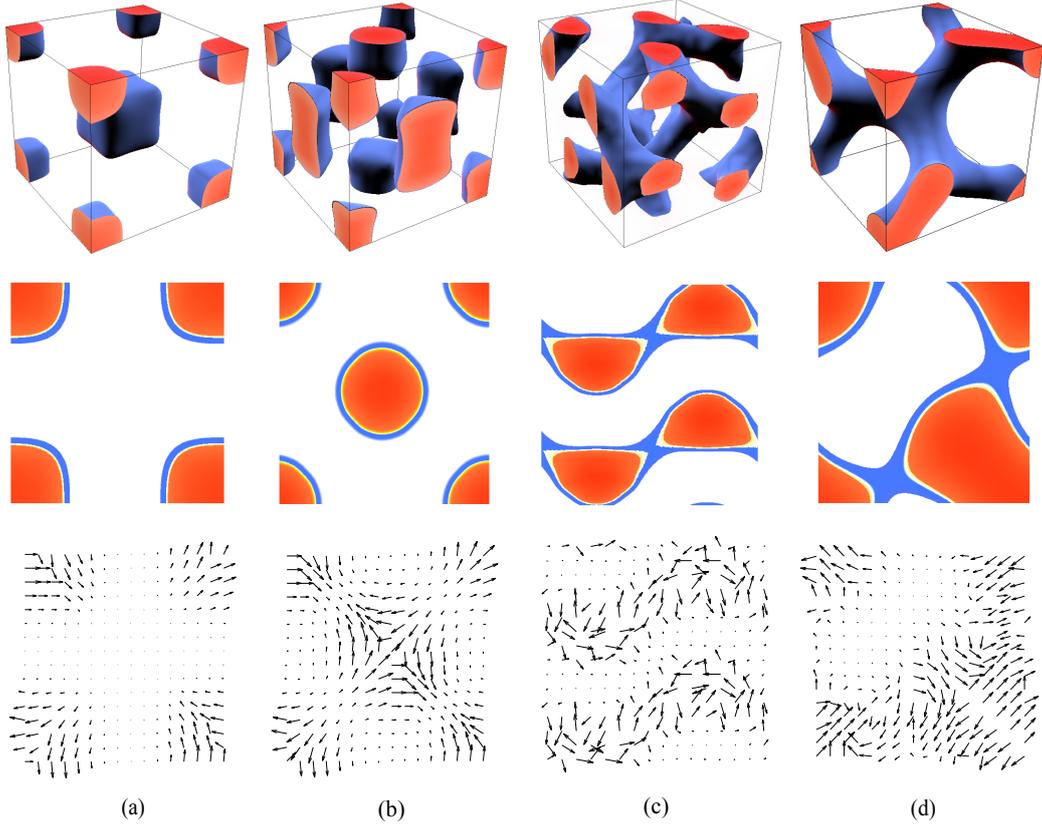


Figure 9: The morphologies of phases (the first row), the top views for morphologies (second row), and the corresponding orientation distributions (the third row) of (a) BCC, (b) FCC, (c) DG and (d) SG, respectively.

4.4. The efficiency of optimizing the computational domain

In this section, we express the efficiency of optimizing computational domain, which can automatically adjust the size and shape of the computational box in the SCFT calculation. The error tolerance ε_H is set as 10^{-4} .

To show that the computational box can be adjusted automatically, we present two cases with different initial values. Fix model parameters $\chi N = 18$, $f = 0.75$, $\eta N = 9$, $\lambda = 300$, $\beta = 2$ and discretization nodes $N_r = 16$, $N_\theta = 8$, $N_\varphi = 0$, $N_s = 100$. We illustrate the effectiveness of optimizing computational domain from the aspects of the angle θ between edges, the length a_1 , a_2 of the computational domain and the free energy H during the iteration of the optimal computational domain.

Using the lamellar phase within a square domain as the initial values, through the SCFT iteration and optimization process, the lamellar phase transitions into a $[3^6]$ phase, as shown in Fig. 10. The variations in θ , a_1 , a_2 , and H during the optimization process are presented in Fig. 11. The constant angles and changing side lengths indicate that the optimal domain for the $[3^6]$ phase should be rectangular rather than square. The convergence of the free energy ensures the stability of the entire optimization process.

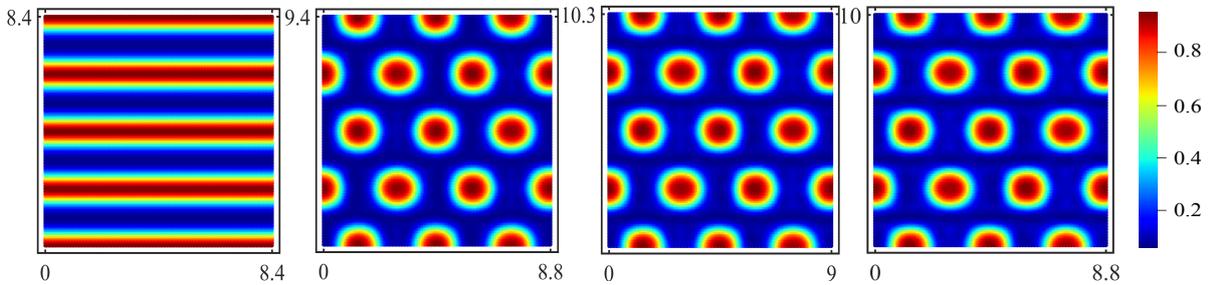


Figure 10: The process of SCFT iteration and optimization starting from the lamellar phase as initial values, transitioning into a $[3^6]$ phase, is shown from left to right. The model parameters $\chi N = 18$, $f = 0.75$, $\eta N = 9$, $\lambda = 300$, $\beta = 2$, discretization nodes $N_r = 20$, $N_\theta = 8$, $N_\varphi = 0$, $N_s = 100$.

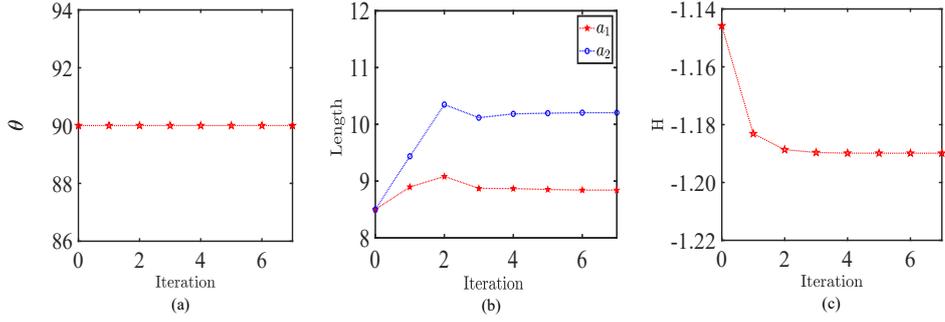


Figure 11: The change process of (a) the angle θ , (b) the box lengths a_1 and a_2 , and (c) the free energy H when lamellar transitions to $[3^6]$.

Starting from the $[3^6]$ phase with rectangular domain, the phase transitions into a $[4^4]$ phase with parallelogram domain, as shown in Fig. 12. This result is consistent with previous findings [10]. As shown in Fig. 13, the variations in angle, side lengths, and free energy provide support for this optimization process.

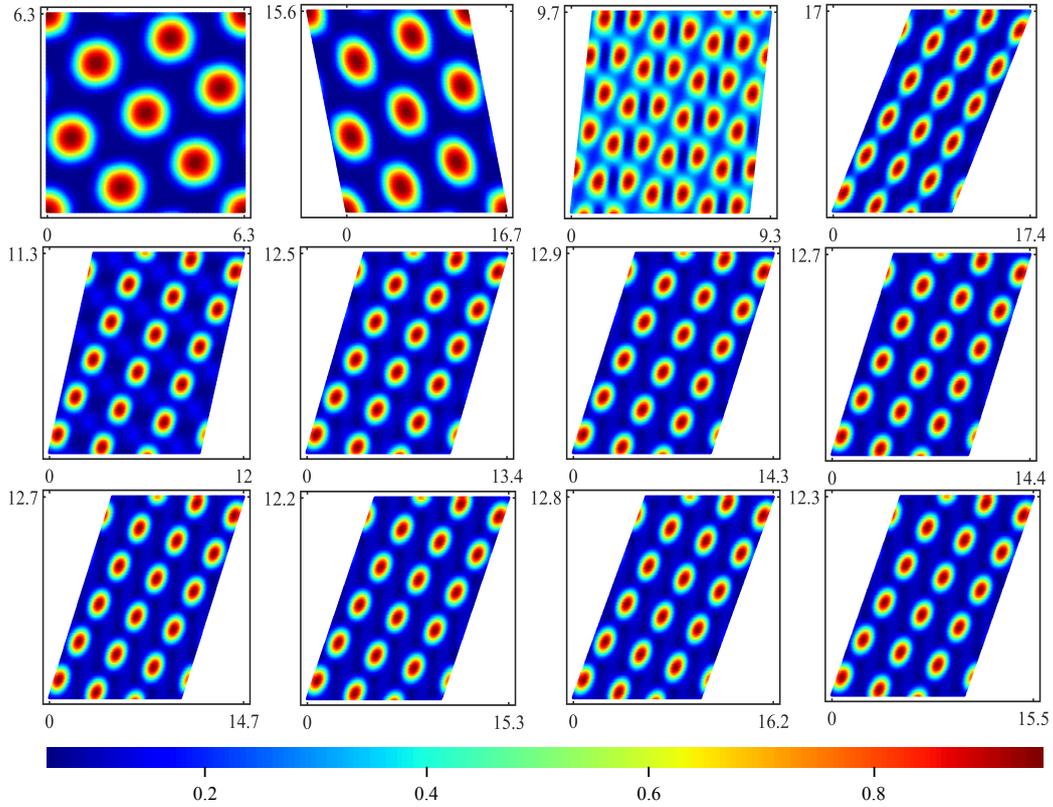


Figure 12: The process of SCFT iteration and optimization starting from the $[3^6]$ phase as initial values, transitioning into a $[4^4]$ phase, is shown from left to right. The model parameters $\chi N = 18$, $f = 0.75$, $\eta N = 9$, $\lambda = 300$, $\beta = 2$; discretization nodes $N_r = 16$, $N_\theta = 8$, $N_\varphi = 0$, $N_s = 100$.

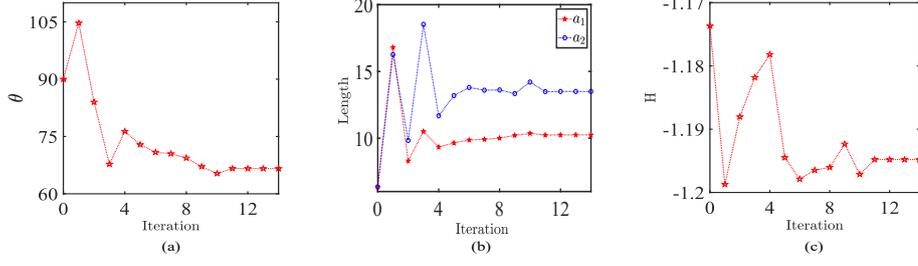


Figure 13: The change process of (a) the angle θ , (b) the box lengths a_1 and a_2 , and (c) the free energy H when $[3^6]$ transitions to $[4^4]$ phases.

5. Conclusion and future work

In this paper, we present and examine ten numerical discrete schemes for solving semiflexible propagators and three nonlinear iteration methods. By comparing the accuracy and computational complexity of these numerical methods, we find that the fourth-order Runge-Kutta scheme is superior to others. Numerical results demonstrate the adaptive Anderson mixing method dramatically improve the convergence rate of the Anderson mixing scheme. To accelerate the SCFT iteration, we adopt the cascading multi-level method to accelerate the adaptive Anderson mixing method, resulting in a notable reduction of CPU time by 63.88%. Meanwhile, we optimize the computational domain during the SCFT calculation to find the optimal period of different phases successfully. Furthermore, we investigate the self-assembly structures of flexible-semiflexible diblock copolymers in 1D, 2D, and 3D spaces. All numerical methods proposed in this paper are adaptable to all liquid crystalline polymer systems. In the future, we aim to apply these numerical methods to more liquid crystalline polymers.

Appendix A. Spherical harmonic expansion

The square integrable function $q(\mathbf{r}, \mathbf{u}, s)$ on unit sphere $S(\mathbf{u} \in S)$ can be expanded by spherical harmonic series as

$$q(\mathbf{r}, \mathbf{u}, s) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} q_{lm}(\mathbf{r}, s) Y_{l,m}(\mathbf{u}), \quad (\text{A.1})$$

with $Y_{l,m}(\mathbf{u})$ is the spherical harmonic function [3], and

$$\begin{aligned} q_{lm}(\mathbf{r}, s) &= \int_S q(\mathbf{r}, \mathbf{u}, s) Y_{l,m}^*(\mathbf{u}) d\mathbf{u} \\ &= \int_0^\pi \int_0^{2\pi} \sin \theta q(\mathbf{r}, \theta, \varphi, s) Y_{l,m}^*(\theta, \varphi) d\varphi d\theta, \end{aligned} \quad (\text{A.2})$$

where $Y_{l,m}^*$ is the complex conjugate of $Y_{l,m}$. The vector \mathbf{u} can be expressed through unit sphere coordinates

$$\mathbf{u} = (x, y, z)^T = (\cos \varphi \sin \theta, \sin \theta \sin \varphi, \cos \theta)^T, \quad \varphi \in [0, 2\pi), \theta \in [0, \pi). \quad (\text{A.3})$$

The spherical harmonic function $Y_{l,m}(\mathbf{u})$ can be expressed as

$$Y_{l,m}(\mathbf{u}) = Y_{l,m}(\theta, \varphi) = (-1)^m \left[\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} \right]^{1/2} P_l^m(\cos \theta) e^{im\varphi}, \quad (\text{A.4})$$

where $P_l^m(x)$ is the associated Legendre function

$$\begin{aligned} P_l^m(x) &= (1-x^2)^{\frac{m}{2}} \frac{d^m}{dx^m} P_l(x), \quad l, m \geq 0, \\ P_l^m(x) &= (-1)^m \frac{(l-|m|)!}{(l+|m|)!} P_l^{|m|}(x), \quad m < 0, \end{aligned} \quad (\text{A.5})$$

with $P_l(x)$ is the Legendre polynomial

$$P_l(x) = \frac{1}{2^l l!} \frac{d^l}{dx^l} (x^2 - 1)^l, \quad l \geq 0. \quad (\text{A.6})$$

Besides, the spherical harmonic function $Y_{l,m}(\mathbf{u})$ is the characteristic function of the Laplace operator $\nabla_{\mathbf{u}}^2$ on the unit sphere,

$$\nabla_{\mathbf{u}}^2 Y_{l,m}(\mathbf{u}) = -l(l+1)Y_{l,m}(\mathbf{u}). \quad (\text{A.7})$$

The spherical harmonic function is a complete orthogonal basis on the sphere, which satisfies orthogonality

$$\begin{aligned} \int_S [Y_{l,m}(\mathbf{u})]^* Y_{l',m'}(\mathbf{u}) d\mathbf{u} &= \int_0^{2\pi} \int_0^\pi \sin\theta [Y_{l,m}(\theta, \varphi)]^* Y_{l',m'}(\theta, \varphi) d\theta d\varphi \\ &= \delta_{ll'} \delta_{mm'}. \end{aligned} \quad (\text{A.8})$$

The SHT and ISHT can be realized by the SPHEREPACK function package [26].

Appendix B. Fourier pseudo-spectral method

The Fourier pseudo-spectral method is a class of numerical methods for solving partial differential equations used in applied mathematics and scientific computing. It is closely related to the Fourier spectral method but complements it with an additional pseudo-spectral basis, representing functions on a quadrature grid. The Fourier pseudo-spectral method simplifies the evaluation of certain operators and can considerably speed up the calculation with FFT. The Fourier pseudo-spectral method is a natural choice for obtaining optimal spatial accuracy under periodic boundary conditions.

Before we show the details of the Fourier pseudo-spectral methods, a brief introduction to the Bravais lattice and reciprocal lattice is necessary. The Bravais lattice is defined by $R_n = n_1 \mathbf{a}_1 + \dots + n_d \mathbf{a}_d$, $n_1, \dots, n_d \in \mathbb{Z}$, the primitive vectors are denoted by $\mathbf{a}_i = (a_{i1}, \dots, a_{id})$, where $i = 1, \dots, d$. d is the geometric dimension of space. Reciprocal vectors $\mathcal{B} = (\mathbf{b}_1, \dots, \mathbf{b}_d)$. The corresponding reciprocal lattice primitive vectors are $\mathbf{b}_i = (b_{i1}, \dots, b_{id})$, $\Lambda^* = k_1 \mathbf{b}_1 + \dots + k_d \mathbf{b}_d$, $k_1, \dots, k_d \in \mathbb{Z}$ is the reciprocal lattice. These two sets of primitive vectors satisfy $\mathbf{a}_i \cdot \mathbf{b}_j = 2\pi \delta_{ij}$, where $i, j = 1, \dots, d$.

Taking the 1D spatial case as an example, the spatial variable r is discretized into grid points r_j , $j = 1, \dots, N$. The periodic function can be expanded as [43]

$$f(r_j) = \sum_{k=-N}^N \hat{f}(k) e^{ikr_j}, \quad (\text{B.1})$$

where the discrete Fourier coefficient \hat{f} can be obtained by FFT. Applying the Fourier pseudo-spectral method to solve the propagator (Eqn. (9)), it can be expanded into

$$\frac{\partial}{\partial s} \sum_{k=-N}^N \hat{q}_A(k, s) e^{ikr} = \nabla_r^2 \sum_{k=-N}^N \hat{q}_A(k, s) e^{ikr} - \sum_{k_1+k_2=-N}^N \hat{w}_A(k_1) e^{ik_1 r} \sum_{k_2=-N}^N \hat{q}_A(k_2, s) e^{ik_2 r}, \quad (\text{B.2})$$

Take the inner product of both sides of Eqn. (B.2) with e^{ikr} , we get

$$\frac{\partial}{\partial s} \hat{q}_A(k, s) = -k^2 \hat{q}_A(k, s) - \sum_{k_1+k_2=-N}^N \hat{w}_A(k_1) \hat{q}_A(k_2, s). \quad (\text{B.3})$$

Since the last term of the above formula creates a convolution that makes the calculation more complicated, we set $G(r, s) = w(r)q(r, s)$, and get the semi-discrete scheme

$$\frac{\partial}{\partial s} \hat{q}_A(k, s) = -k^2 \hat{q}_A(k, s) - \hat{G}(k, s). \quad (\text{B.4})$$

FFT can be performed for the current calculation. Since propagators $q_A(\mathbf{r}, s)$, $q_A^\dagger(\mathbf{r}, s)$, $q_B(\mathbf{r}, \mathbf{u}, s)$ and $q_B^\dagger(\mathbf{r}, \mathbf{u}, s)$ (see Eqns. (4)-(7)) satisfy the periodic boundary conditions on the spatial variable, we use the Fourier pseudo-spectral method to solve them.

Appendix C. Full discretization scheme of the four-dimensional semiflexible propagator equation

Here we present the full discretization scheme with 1-dimensional space variable r , 2-dimensional orientation variable $\mathbf{u} = (\theta, \varphi)$, and 1-dimensional time variable s for equation

$$\frac{\partial}{\partial s} q_B(r, \mathbf{u}, s) = -\beta \cos \theta \nabla_r q_B(r, \mathbf{u}, s) + \Gamma(r, \mathbf{u}) q_B(r, \mathbf{u}, s) + \frac{1}{2\lambda} \nabla_{\mathbf{u}}^2 q_B(r, \mathbf{u}, s), \quad 1 - f \leq s \leq 1, \quad (\text{C.1})$$

where $\Gamma(r, \mathbf{u}) = -(w_B(r) - \mathbf{M}(r) : [\mathbf{u}\mathbf{u} - \frac{1}{3}\mathbf{I}])$. By expanding $q_B(r, \mathbf{u}, s)$ in a Fourier series with respect to the spatial variable r and in a spherical harmonic series with respect to the orientation variable \mathbf{u} , the equation can be formulated as

$$\sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l \frac{\partial \tilde{q}_B(k, l, m, s)}{\partial s} e^{ikr} Y_{l,m}(\mathbf{u}) = M_1 + M_2 + M_3, \quad (\text{C.2})$$

where $2N + 1$ is the number of discrete points in space, $L + 1$ is the number of discrete nodes of θ , and

$$\begin{aligned} M_1 &= \sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l -\frac{l(l+1)}{2\lambda} \tilde{q}_B(k, l, m, s) e^{ikr} Y_{l,m}(\mathbf{u}), \\ M_2 &= \sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l \tilde{\Gamma}_B(k, l, m, s) e^{ikr} Y_{l,m}(\mathbf{u}), \\ M_3 &= \sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l -i\beta \tilde{U}_B(k, l, m, s) e^{ikr} Y_{l,m}(\mathbf{u}). \end{aligned} \quad (\text{C.3})$$

The coefficients $\tilde{q}_B(k, l, m, s)$, $\tilde{U}_B(k, l, m, s)$, $\tilde{\Gamma}_B(k, l, m, s)$ are obtained by first performing Fourier transforms on $q_B(r, \mathbf{u}, s)$, $q_B(r, \mathbf{u}, s) \cos \theta$ and $\Gamma(r, \mathbf{u}) q_B(r, \mathbf{u}, s)$, followed by spherical harmonic transforms, respectively. Using the testing function $v = e^{ik'r} Y_{l',m'}(\theta, \varphi)$, Eqn.(C.2) can be written as

$$\begin{aligned} & \sum_{p=1}^{2N} \sum_{k=-N}^{N-1} \sum_{j=1}^{L+1} \sum_{l=0}^L \sum_{n=1}^{2L+1} \sum_{m=-l}^l \frac{\partial \tilde{q}_B(k, l, m, s)}{\partial s} e^{ikr_p} e^{-ik'r_p} Y_{l,m}(\theta_j, \varphi_n) Y_{l',m'}^*(\theta_j, \varphi_n) \\ &= \sum_{p=1}^{2N} \sum_{k=-N}^{N-1} \sum_{j=1}^{L+1} \sum_{l=0}^L \sum_{n=1}^{2L+1} \sum_{m=-l}^l -\frac{l(l+1)}{2\lambda} \tilde{q}_B(k, l, m, s) e^{ikr_p} e^{-ik'r_p} Y_{l,m}(\theta_j, \varphi_n) Y_{l',m'}^*(\theta_j, \varphi_n) \\ &+ \sum_{p=1}^{2N} \sum_{k=-N}^{N-1} \sum_{j=1}^{L+1} \sum_{l=0}^L \sum_{n=1}^{2L+1} \sum_{m=-l}^l \tilde{\Gamma}_B(k, l, m, s) e^{ikr_p} e^{-ik'r_p} Y_{l,m}(\theta_j, \varphi_n) Y_{l',m'}^*(\theta_j, \varphi_n) \\ &- \sum_{p=1}^{2N} \sum_{k=-N}^{N-1} \sum_{j=1}^{L+1} \sum_{l=0}^L \sum_{n=1}^{2L+1} \sum_{m=-l}^l i\beta \tilde{U}_B(k, l, m, s) e^{ikr_p} e^{-ik'r_p} Y_{l,m}(\theta_j, \varphi_n) Y_{l',m'}^*(\theta_j, \varphi_n), \end{aligned} \quad (\text{C.4})$$

where $\theta_j = \frac{\pi j}{L+1}$, $j = 1, 2, \dots, L+1$, $\varphi_n = \frac{2\pi n}{2L+1}$, $n = 1, 2, \dots, 2L+1$, $r_p = \frac{|B|p}{2N}$, $p = 1, 2, \dots, 2N$, and the periodic boundary condition makes $q_B(r_0, \mathbf{u}, s) = q_B(r_{2N}, \mathbf{u}, s)$.

Define the discrete form of inner-product as

$$(q_B, v) = \sum_{p=1}^{2N} \sum_{j=1}^{L+1} \sum_{n=1}^{2L+1} q_B(r_p, \theta_j, \varphi_n) v^*(r_p, \theta_j, \varphi_n), \quad (\text{C.5})$$

where v^* is the complex conjugate of v . Thus, Eqn.(C.4) can be written as

$$\begin{aligned}
& \sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l \frac{\partial \tilde{q}_B(k, l, m, s)}{\partial s} (e^{ikr} Y_{l,m}(\theta, \varphi), e^{ik'r} Y_{l',m'}(\theta, \varphi)) \\
&= \sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l -\frac{l(l+1)}{2\lambda} \tilde{q}_B(k, l, m, s) (e^{ikr} Y_{l,m}(\theta, \varphi), e^{ik'r} Y_{l',m'}(\theta, \varphi)) \\
&+ \sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l \tilde{\Gamma}_B(k, l, m, s) (e^{ikr} Y_{l,m}(\theta, \varphi), e^{ik'r} Y_{l',m'}(\theta, \varphi)) \\
&- \sum_{k=-N}^{N-1} \sum_{l=0}^L \sum_{m=-l}^l i\beta \tilde{U}_B(k, l, m, s) (e^{ikr} Y_{l,m}(\theta, \varphi), e^{ik'r} Y_{l',m'}(\theta, \varphi)).
\end{aligned} \tag{C.6}$$

Simplifying it leads to the full discretization form

$$\frac{\partial \tilde{q}_B(k, l, m, s)}{\partial s} = -\frac{l(l+1)}{2\lambda} \tilde{q}_B(k, l, m, s) + \tilde{\Gamma}_B(k, l, m, s) - i\beta \tilde{U}_B(k, l, m, s). \tag{C.7}$$

References

- [1] P.G.De. Gennes and J. Prost. *The Physics of Liquid Crystal*. Oxford University Press, 1993.
- [2] F. Schmid. Self-consistent-field theories for complex fluids. *J. Phys. Condens. Matter*, 10(37):8105–8138, 1998.
- [3] G.H. Fredrickson. *The Equilibrium Theory of Inhomogeneous Polymers*. Oxford University Press, 2006.
- [4] Y. Jiang and J.Z.Y. Chen. Isotropic-Nematic interface in a lyotropic system of wormlike chains with the onsager interaction. *Macromolecules*, 43(24):10668–10678, 2010.
- [5] X. Yao, H. Zhang, and J.Z.Y. Chen. Topological defects in two-dimensional liquid crystals confined by a box. *Phys. Rev. E*, 97:052707, 2018.
- [6] M. Matsen and M. Schick. Stable and unstable phases of a linear multiblock copolymer melt. *Macromolecules*, 27(24):7157–7163, 1994.
- [7] F. Drolet and G.H. Fredrickson. Combinatorial screening of complex block copolymer assembly with self-consistent field theory. *Phys. Rev. Lett.*, 83(21):4317–4320, 1999.
- [8] M.W. Matsen. Melts of semiflexible diblock copolymer. *J. Chem. Phys.*, 104(19):7758–7764, 1996.
- [9] S. Li, Y. Jiang, and J.Z.Y. Chen. Phase transitions in semiflexible-rod diblock copolymers: A self-consistent field theory. *Soft Matter*, 10:8932–8944, 2014.
- [10] Y. Jiang and J.Z.Y. Chen. Self-consistent field theory and numerical scheme for calculating the phase diagram of wormlike diblock copolymers. *Phys. Rev. E*, 88(4):042603, 2013.
- [11] W. Song, P. Tang, F. Qiu, Y. Yang, and A.-C. Shi. Phase behavior of semiflexible-coil diblock copolymers: A hybrid numerical scft approach. *Soft Matter*, 7:929–938, 2011.
- [12] Z. He, X. Wang, P. Zhang, A.-C. Shi, and K. Jiang. Theory of polygonal phases self-assembled from t-shaped liquid crystalline molecules. *Macromolecules*, 57(5):2154–2164, 2024.
- [13] F. Liu, P. Tang, H. Zhang, and Y. Yang. Archimedean tiling patterns self-assembled from x-shaped rod-coil copolymers with hydrogen bonds. *Macromolecules*, 51(19):7807–7816, 2018.
- [14] W. Song, P. Tang, H. Zhang, Y. Yang, and A.-C. Shi. New numerical implementation of self-consistent field theory for semiflexible polymers. *Macromolecules*, 42(16):6300–6309, 2009.
- [15] N. Saito, K. Takahashi, and Y. YunokiK. The statistical mechanical theory of stiff chains. *J. Phys. Soc. Jpn.*, 22(1):219–226, 1967.
- [16] Y. Jiang and J.Z.Y. Chen. Influence of chain rigidity on the phase behavior of wormlike diblock copolymers. *Phys. Rev. Lett.*, 110(13):138305, 2013.
- [17] Q. Liang, K. Jiang, and P. Zhang. Efficient numerical schemes for solving the self-consistent field equations of flexible-semiflexible diblock copolymers. *Math.Methods Appl. Sci.*, 38(18):4553–4563, 2015.
- [18] Y. Jiang, W. Zhang, and J.Z.Y. Chen. Dependence of the disorder-lamellar stability boundary of a melt of asymmetric wormlike A B diblock copolymers on the chain rigidity. *Phys. Rev. E*, 84:041803, 2011.
- [19] H.D. Ceniceros. Efficient order-adaptive methods for polymer self-consistent field theory. *J. Comput. Phys.*, 386:9–21, 2019.
- [20] H.D. Ceniceros and G.H. Fredrickson. Numerical solution of polymer self-consistent field theory. *Multiscale Model. Simul.*, 2(3):452–474, 2004.
- [21] K. Jiang, W. Xu, and P. Zhang. Analytic structure of the SCFT energy functional of multicomponent block copolymers. *Commun Comput Phys.*, 17(5):1360–1387, 2015.
- [22] T.L. Chantawansri, A.W. Bosse, A. Hexemer, H.D. Ceniceros, C.J. Garcia-Cervera, E.J. Kramer, and G.H. Fredrickson. Self-consistent field theory simulations of block copolymer assembly on a sphere. *Phys. Rev. E*, 75(3):031802, 2007.
- [23] E.W. Cochran, C.J. Garcia-Cervera, and G.H. Fredrickson. Stability of the gyroid phase in diblock copolymers at strong segregation. *Macromolecules*, 39(7):2449–2451, 2006.
- [24] K. Jiang, J. Zhang, and Q. Liang. Self-assembly of asymmetrically interacting ABC star triblock copolymer melts. *J. Phys. Chem. B*, 119(45):14551–14562, 2015.
- [25] A. Duttand, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numer. Math.*, 40(2):241–266, 2000.
- [26] J.C. Adams and P.N. Swarztrauber. SPHEREPACK 3.0: A model development facility. *Mon. Wea. Rev.*, 127(8):1872–1878, 1999.

- [27] U.M. Ascher, S.-J. Ruuth, and B.T.R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32(3):797–823, 1995.
- [28] G. Wanner and E. Hairer. *Solving ordinary differential equations II*. Springer Berlin Heidelberg, 1996.
- [29] J. Gao, P. Tang, and Y. Yang. Non-lamellae structures of coil–semiflexible diblock copolymers. *Soft Matter*, 9(1):69–81, 2013.
- [30] B.D. Olsen and R.A. Segalman. Self-assembly of rod–coil block copolymers. *Mater. Sci. Eng. R Rep.*, 62(2):37–66, 2008.
- [31] D.G. Anderson. Iterative procedures for nonlinear integral equations. *J. ACM*, 12(4):547–560, 1965.
- [32] A. Akash, D.C. Morse, F.S. Bates, and K.D. Dorfman. Accelerating self-consistent field theory of block polymers in a variable unit cell. *J. Chem. Phys.*, 146(24):244902, 06 2017.
- [33] P.P. Pratapa, P. Suryanarayana, and J.E. Pask. Anderson acceleration of the jacobi iterative method: An efficient alternative to krylov methods for large, sparse linear systems. *J. Comput. Phys.*, 306:43–54, 2016.
- [34] S. Pollock, L.G. Rebholz, and M. Xiao. Anderson-accelerated convergence of picard iterations for incompressible navier–stokes equations. *SIAM J. Numer. Anal.*, 57(2):615–637, 2019.
- [35] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [36] J. Barzilai and J.M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.
- [37] YH. Dai. A new analysis on the barzilai-borwein gradient method. *J. Oper. Res. Soc. China*, 1(2):187–198, 2013.
- [38] M.W. Matsen and C. Barrett. Liquid-crystalline behavior of rod-coil diblock copolymers. *J. Chem. Phys.*, 109(10):4108–4118, 1998.
- [39] A.N. Semenov and A. Subbotin. Theory of smectic ordering in melts of polymers containing mesogenic units. *Sov. phys. JETP*, 101:1233–1245, 1992.
- [40] J.T. Chen, E.L. Thomas, C.K. Ober, and G.-P. Mao. Self-assembled smectic phases in rod-coil block copolymers. *Science*, 273(5273):343–346, 1996.
- [41] A.K. Khandpur, S. Foerster, F. Bates, I. Hamley, A. Ryan, W. Bras, K. Almdal, and K. Mortensen. Polyisoprene-polystyrene diblock copolymer phase diagram near the order-disorder transition. *Macromolecules*, 28(26):8796–8806, 1995.
- [42] K. Jiang, C. Wang, Y. Huang, and P. Zhang. Discovery of new metastable patterns in diblock copolymers. *Commun Comput Phys.*, 14(2):443–460, 2013.
- [43] K. Jiang, Y. Huang, and P. Zhang. Spectral method for exploring patterns of diblock copolymers. *J. Comput. Phys.*, 229:7796–7805, 2010.