Computer Methods in Applied Mechanics and Engineering 439 (2025) 117878

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

On efficient simulation of self-assembling diblock copolymers using a peridynamic-enhanced Fourier spectral method

Farshid Mossaiby [a], Gregor Häfner [b,c], Arman Shojaei [a,*], Alexander Hermann [a], Christian Cyron [a,d], Marcus Müller [b], Stewart Silling [e]

[a] Institute of Material Systems Modeling, Helmholtz-Zentrum Hereon, Max–Planck-Str. 1, Geesthacht 21502, Germany
[b] Georg-August Universität Göttingen, Institut für Theoretische Physik, Friedrich-Hund Platz 1, 37077 Göttingen, Germany
[c] Max Planck School Matter to Life, Jahnstraße 29, 69120 Heidelberg, Germany
[d] Institute for Continuum and Material Mechanics, Hamburg University of Technology, Eissendorfer Str. 42, 21073 Hamburg, Germany
[e] Center for Computing Research, Sandia National Laboratories, P.O. Box 5800, MS-1322, Albuquerque, NM, 87185-1322, USA

ARTICLE INFO

Keywords:
Fourier spectral method
Peridynamics
Stability analysis
Diblock copoloymers
Self-assembly

ABSTRACT

This study introduces a computational framework for simulating the self-assembly of diblock copolymers using a novel peridynamic (PD)-enhanced Fourier spectral method (FSM). Diblock copolymers, composed of two distinct polymer blocks, are capable of forming nanostructured domains with applications in nanoelectronics, photonics, and advanced membranes. Current simulation techniques face challenges in capturing the multiscale dynamics of polymer systems and are often limited by computational inefficiencies. Our approach combines a phase-field model with FSM for spatial discretization and leverages a PD-based diffusion operator to overcome the stability restrictions of explicit time-stepping schemes. This integration allows for larger time steps, ensuring both stability and computational efficiency. The method's scalability is enhanced through parallel implementation using C++ and OpenMP, optimized for multi-core CPUs. Validation through phase diagrams of copolymer melts and simulations of evaporation-induced self-assembly (EISA) processes demonstrates the capability of the proposed method to accurately capture large-scale, dynamic morphologies. Our approach provides a versatile framework and was found in certain examples to improve computational efficiency by more than a factor of 6 compared to forward-Euler FSM approach.

1. Introduction

Polymers, which are long macromolecular chains, have captivated scientists for many years. Advances in polymer systems and structures have led to sophisticated applications, ranging from block polymers as surfactants to highly functional macromolecular systems such as porous polyelectrolytes for catalysis and battery fabrication [1,2]. Controlled polymer synthesis allows for elaborate architectures, such as block copolymers, which are molecules made of distinct covalently bonded blocks. When the repulsion between different blocks are high enough, the material microphase separates into nanoscopic domains [3–5]. Depending on the architecture, diblock copolymers melts can form various equilibrium morphologies like lamellae, cylinders, and spheres. Ternary mixtures like a diblock copolymer solution or a triblock copolymer melt exhibit an even higher variety of morphologies [6,7]. Additionally, non-equilibrium states can be kinetically trapped [8–10], offering a range of applications from nanoelectronics to photonic devices and integral asymmetric copolymer membranes for filter applications [8,11,12].

* Corresponding author.
E-mail address: arman.shojaei@hereon.de (A. Shojaei).

https://doi.org/10.1016/j.cma.2025.117878
Received 16 January 2025; Received in revised form 11 February 2025; Accepted 20 February 2025
Available online 8 March 2025

Recent advancements have significantly improved industrial applications of copolymer membranes. Techniques such as immersion precipitation and non-solvent induced phase separation (NIPS) exploit the macrophase separation of homopolymers and non-solvents to create coarse, sponge-like substructures with a finer selective layer. Enhancements in these techniques, combining the NIPS process with evaporation induced self assembly (EISA) of a block copolymer solution, leverage the self-assembly of the amphiphilic diblock copolymers resulting in an isoporous top layer. By controlling self-assembly through solvent evaporation, vertical-cylindric morphologies are achieved. Combined with NIPS, this process produces an integral asymmetric membrane with highly regular surface pores [8,12,13].

The EISA process is well established experimentally. [8,13,14]. However, computer simulations are needed to quantitatively understand and improve membrane formation [15–19]. The challenge lies in the problem of multiple scales. Polymer systems involve vast numbers of polymer segments, with dynamics occurring on a much smaller length scale than the overall system size, making atomistic simulations challenging. Coarse-graining techniques offer a solution by simplifying the problem, making experimental time scales accessible.

The dynamics of block copolymer melts and solutions can be studied by highly coarse-grained models due to the universality of copolymer assembly [20]. Two strategies exist for the coarse-graining of polymer systems. *Bottom-up* coarse-graining lumps several monomer repeat units into an interaction center. This results in soft interactions of the coarse-grained beads, whose interaction parameters depend on the microscopic details of the systems. On the other hand, *top-down* approaches make use of experimentally measurable parameters, that are invariant under different degrees of coarse graining — scales of length, time and energy. Such parameters implicitly incorporate the chemical details, are measurable in experiments. Particle-based soft coarse-grained models can be implemented using dissipative particle dynamics, or by Monte-Carlo Methods [21]. Another large class of polymer simulations arise from self-consistent field theory (SCFT) [22,23], which is a continuum theory that retains the statistics of polymer configurations.

Common continuum theories for block copolymer systems have their origin in the random-phase approximation (RPA) of SCFT, which expands the free energy in powers of the concentration field around the spatially homogeneous reference state. RPA becomes accurate in the weak segregation limit [3]. The connection between SCFT and continuum models allows for a physical interpretation of the parameters of the continuum model. There are different strategies to derive continuum models from RPA by describing the inverse of the collective structure factor of concentration fluctuations [24]. The Ohta-Kawasaki model (OKM) provides a free-energy functional for diblock copolymer melts [25]. The inverse structure factor is approximated by three contributions: a short- and a long-range contribution, as well as a constant one to match the order–disorder transition. This approach has been generalized by Uneyama and Doi to comprehensively treat blends of block copolymers of different architectures [26]. The resulting free-energy functional – the Uneyama-Doi model (UDM) – is not exact but reduces to the correct limiting cases of single-architecture systems. It has been successfully used to study the equilibrium phases of copolymer-solvent mixtures [27,28]. Since chemical potentials are readily available in the continuum theories, these allow to study the time evolution of nonequilibrium systems using model-B dynamics [29]. Recent applications studied the self-assembly dynamics of amphiphilic molecules in solution [30–32], as well as structure formation of diblock copolymers upon solvent evaporation [17].

The numerical study of the kinetics of structure formation, predicted by this continuum model, involves several challenges: First, the resulting set of partial differential equations are highly non-linear. Second, the functional contains *convolution terms* that are expensive to calculate in real-space. Third, using an explicit forward-Euler scheme to discretize the equations in time requires a prohibitive amount of computational cost due to very small time step required. Implicit schemes can handle much larger time steps, but are very complex to derive and implement. Semi-implicit schemes [17,31,32], which only treat certain terms implicitly, can act as a middle ground, but become more challenging when density-dependent mobilities are of interest [33].

In this work, we address the aforementioned challenges by employing the Fourier spectral method (FSM) for spatial discretization and the forward-Euler explicit scheme for temporal discretization, providing a simple yet general framework for tackling the problem. To enhance computational efficiency, we follow the guidelines of Uneyama and Doi [26,30] for the efficient calculation of convolution terms. Moreover, to overcome the restrictive time step limitations inherent to the forward-Euler scheme, we propose using a nonlocal PD-based diffusion operator as a replacement for the traditional spatial gradient or Laplacian-based operators. PD models, originally formulated in [34], offer a compelling advantage in handling problems involving discontinuities or non-smooth fields by replacing partial differential equations with integro-differential formulations over a finite interaction radius, known as the PD *horizon*. Unlike traditional partial differential operators, which require increasingly fine spatial and temporal resolutions near singularities or sharp interfaces, the nonlocal formulation mitigates these issues by distributing interactions across a finite domain, leading to relaxed numerical-stability constraints [35–37]. The nonlocal nature of the PD operator inherently accommodates the multiscale character of the problem by naturally bridging fine-scale interactions with macroscopic behavior [38]. Moreover, the capability of PD models to incorporate evolving discontinuities, such as sharp interfaces or fracture-like features, without requiring explicit tracking, further enhances their applicability in scenarios where conventional methods often fail [39–45]. In addition, nonlocal diffusion operators, like those applied in reaction–diffusion systems, have demonstrated significant potential in modeling complex interactions across different scales [46,47]. This study introduces a novel computational strategy by integrating a PD operator within a FSM. This hybrid approach enables stable simulations with larger time steps while maintaining high accuracy, significantly enhancing numerical efficiency over conventional explicit forward-Euler schemes or standard convolution-based implementations. Notably, this combination has not been previously applied to the self-assembly of diblock copolymers, representing a valuable contribution to computational materials science. We will mathematically demonstrate that employing the PD operator can substantially increase the maximum time step size compared to the forward-Euler FSM counterpart, thereby enhancing overall performance. We further detail an efficient parallel implementation of our method on multi-core CPUs using C++ and

OpenMP directives, ensuring scalability for large-scale simulations. The outline of the paper is as follows. In the next section, the continuum model is developed and the set of governing equations are derived. The solution strategy is described and justified in Section 3. The details of the computer implementation as well as a step-by-step solution algorithm is presented in Section 4. The performance of the model is examined in Section 5. In this section, we first validate the model, using the known phase diagram for a dicopolymer melt [3,48]. Finally, we investigate the performance of the proposed method in case of an EISA problem. Concluding comments can be found in Section 6.

2. The continuum model

The Uneyama-Doi model [17,26,31,32] treats the normalized local concentration fields, $\phi_c(\mathbf{r})$ as order parameters, where $\mathbf{r}$ is the spatial coordinate and the index c runs over molecular species and their blocks. In this work, we consider one type of diblock copolymer (blocks A and B with block ratio $f_A = 1 - f_B$) and solvents $S, \ldots$, hence $c = A, B, S, \ldots$. The free energy is expressed as a functional of the concentration fields:

$$\begin{aligned}\frac{\mathcal{F}[\{\phi_c(\mathbf{r})\}]}{\sqrt{\bar{\mathcal{N}}}k_{\mathrm{B}}T} = \frac{1}{R_{\mathrm{e}}^3}\int \Bigg[&\sum_{cc'=A,B} A_{cc'}\sqrt{\phi_c(\mathbf{r})}\int \mathcal{G}(\mathbf{r}-\mathbf{r}')\sqrt{\phi_{c'}(\mathbf{r}')}\,\mathrm{d}\mathbf{r}' + \sum_c \frac{R_{\mathrm{e}}^2}{24\phi_c(\mathbf{r})}\left|\nabla\phi_c(\mathbf{r})\right|^2 \\ &+ \sum_c C_c\phi_c(\mathbf{r})\ln\phi_c(\mathbf{r}) - \sqrt{\frac{\phi_A(\mathbf{r})\phi_B(\mathbf{r})}{f_A f_B}} + \frac{1}{2}\sum_{c,c'}\chi_{cc'}N_0\phi_c(\mathbf{r})\phi_{c'}(\mathbf{r}) + \pi(\mathbf{r})N_0\left(\sum_c \phi_c - 1\right)\Bigg]\,\mathrm{d}\mathbf{r}\end{aligned} \tag{1}$$

where we introduced a reference polymerization of the diblock copolymer, N_0, and corresponding mean end-to-end distance, R_{e}, as well as the corresponding invariant degree of polymerization, $\bar{\mathcal{N}}$. k_BT denotes the thermal energy scale, and $A_{cc'}$ and C_c are architecture-dependent coefficients. In the free energy, the first term on the right hand side corresponds to the covalent bonding of the two blocks of the diblock copolymer, giving rise to a long-range interaction, where the convolution kernel $\mathcal{G}(\mathbf{r})$ fulfills the modified Poisson equation, $(-\nabla^2 + \xi_{\mathrm{cut}}^{-2})\mathcal{G}(\mathbf{r}) = \delta(\mathbf{r})$ for cutoff length scale $\xi_{\mathrm{cut}} = 2R_{\mathrm{e}}$. The coefficient matrix is given by:

$$A = \frac{9}{R_{\mathrm{e}}^2 f_A^2 f_B^2}\begin{pmatrix} f_B^2 & -f_A f_B \\ -f_A f_B & f_A^2 \end{pmatrix} \tag{2}$$

matching the diblock copolymer structure factor at small wavenumbers. The second term is a square-gradient penalty that enforces a finite interface width and is derived from the large-wavenumber limit of the structure factor. The third and fourth term describe the entropy of mixing, accounting for the different molecular weights of polymer and solvents:

$$C_c = \frac{N_0}{N_c} \text{ for } c = S, \ldots \tag{3}$$

while the coefficients, C_A and C_B, are chosen to match the order–disorder transition for the blocks of the copolymer

$$C_c = \frac{s(f_c) - f_c}{4f_c(1-f_c)^2} \text{ for } c = A, B \tag{4}$$

with $s(f)$ defined as in the work of Ohta and Kawasaki [25] and approximated by the form:

$$s(f) = 1.572 - 2.702f(1-f) \tag{5}$$

The fifth term in the free energy corresponds to local binary interactions of segments, quantified by the Flory–Huggins parameters, $\chi_{cc'}$ and the final term enforces incompressibility *via* the pressure-like Lagrange multiplier, $\pi(\mathbf{r})$.

Equilibrium morphologies are conformations, $\{\phi_c(\mathbf{r}) \mid c = A, B, S, \ldots\}$, that minimize the free energy. In a nonequilibrium scenario, the chemical potentials, $\mu_c(\mathbf{r}) = \frac{\delta\mathcal{F}}{\delta\phi_c(\mathbf{r})}$ vary in space and their gradients, that give rise to mass fluxes according to model-B dynamics [29]. For the A-block of the copolymer the chemical potential yields:

$$\begin{aligned}\frac{\mu_A(\mathbf{r})R_{\mathrm{e}}^3}{\sqrt{\bar{\mathcal{N}}}k_{\mathrm{B}}T} = &\frac{18}{R_{\mathrm{e}}^2 f_A f_B}\frac{1}{\sqrt{\phi_A(\mathbf{r})}}\int \mathcal{G}(\mathbf{r}-\mathbf{r}')\left[f_B\sqrt{\phi_A(\mathbf{r}')} - \sqrt{f_A f_B}\sqrt{\phi_B(\mathbf{r}')}\right]\mathrm{d}\mathbf{r}' \\ &- \frac{R_{\mathrm{e}}^2}{6}\frac{\nabla^2\sqrt{\phi_A(\mathbf{r})}}{\sqrt{\phi_A(\mathbf{r})}} + \frac{s(f_A)-f_A}{4f_A f_B^2}\ln\phi_A(\mathbf{r}) - \frac{1}{2\sqrt{f_A f_B}}\frac{\sqrt{\phi_B(\mathbf{r})}}{\sqrt{\phi_A(\mathbf{r})}} + \sum_{c'\neq A}\chi_{Ac'}N_P\phi_{c'}(\mathbf{r})\end{aligned} \tag{6}$$

and correspondingly for the B-block by exchanging $A \leftrightarrow B$. For the solvents, $c = S, \ldots$, Eq. (1) yields:

$$\frac{\mu_c(\mathbf{r})R_{\mathrm{e}}^3}{\sqrt{\bar{\mathcal{N}}}k_{\mathrm{B}}T} = \frac{N_0}{N_c}\ln\phi_c(\mathbf{r}) - \frac{R_{\mathrm{e}}^2}{6}\frac{\nabla^2\sqrt{\phi_c(\mathbf{r})}}{\sqrt{\phi_c(\mathbf{r})}} + \sum_{c'\neq c}\chi_{cc'}N_0\phi_c(\mathbf{r}) \tag{7}$$

where constant terms are omitted.

For the time evolution we take model-B dynamics according to the classification of Hohenberg and Halperin [29], which is appropriate for order parameter fields that are locally conserved and thus follow a continuity equation. The incompressibility

constraint is treated explicitly in the free energy and therefore, in the absence of additional constraints, the time evolution reduces to:

$$\begin{aligned}\frac{\partial \phi_c(\mathbf{r},t)}{\partial t} &= -\nabla\cdot\mathbf{j}_c(\mathbf{r},t) = \nabla\cdot\left[\Lambda_c(\mathbf{r},t)\nabla\mu_c(\mathbf{r},t)\right] = \lambda R_e^2\nabla\cdot\left[\phi_c(\mathbf{r},t)\frac{\nabla\mu_c(\mathbf{r},t)R_e^3}{\sqrt{\bar{\mathcal{N}}}k_BT}\right]\\ \Lambda_c(\mathbf{r},t) &= \frac{\lambda R_e^5}{\sqrt{\bar{\mathcal{N}}}k_BT}\phi_c(\mathbf{r},t)\end{aligned} \tag{8}$$

where $\mathbf{j}_c(\mathbf{r})$ are the mass fluxes that are proportional to gradients in the chemical potentials. The proportionality factor is the Onsager coefficient, $\Lambda_c(\mathbf{r},t)$ and the constant λ can be interpreted as the monomer mobility, which we assume equal for every species. It defines the simulation time scale, λ^{-1}.

Finally, the dynamic equations are readily non-dimensionalized by introducing units of time, λ^{-1}, length, R_e, and energy, $\sqrt{\bar{\mathcal{N}}}k_BT$, setting these to unity in the numerical implementation. For better numerical performance near zero densities, Uneyama [26] suggested the following change of variable:

$$\psi_c(\mathbf{r},t) = \sqrt{\phi_c(\mathbf{r},t)} \tag{9}$$

The new chemical potentials can now be calculated as:

$$\tilde{\mu}_c(\mathbf{r}) := \frac{\delta\mathcal{F}}{\delta\psi_c(\mathbf{r})} = \frac{\partial\phi_c(\mathbf{r})}{\partial\psi_c(\mathbf{r})}\frac{\delta\mathcal{F}}{\delta\phi_c(\mathbf{r})} = 2\psi_c(\mathbf{r})\mu_c(\mathbf{r}) \tag{10}$$

This results in the following expression for the new chemical potential $\tilde{\mu}_A$:

$$\begin{aligned}\frac{\tilde{\mu}_A(\mathbf{r})R_e^3}{\sqrt{\bar{\mathcal{N}}}k_BT} &= \frac{36}{R_e^2 f_A f_B}\int \mathcal{G}(\mathbf{r}-\mathbf{r}')\left[f_B\psi_A(\mathbf{r}') - \sqrt{f_A f_B}\psi_B(\mathbf{r}')\right]\mathrm{d}\mathbf{r}'\\ &- \frac{R_e^2}{3}\nabla^2\psi_A(\mathbf{r}) + \frac{s(f_A)-f_A}{2f_A f_B^2}\psi_A(\mathbf{r})\ln\psi_A^2(\mathbf{r}) - \frac{1}{\sqrt{f_A f_B}}\psi_B(\mathbf{r}) + 2\psi_A(\mathbf{r})\sum_{c'\neq A}\chi_{Ac'}N_P\psi_{c'}^2(\mathbf{r})\end{aligned} \tag{11}$$

which can also be use to derive $\tilde{\mu}_A$ by substituting $A \leftrightarrow B$. The remaining chemical potentials $\tilde{\mu}_c, c = S, \ldots$ can be derived as:

$$\frac{\tilde{\mu}_c(\mathbf{r})R_e^3}{\sqrt{\bar{\mathcal{N}}}k_BT} = \frac{2N_0}{N_c}\psi_c(\mathbf{r})\ln\psi_c^2(\mathbf{r}) - \frac{R_e^2}{3}\nabla^2\psi_c(\mathbf{r}) + 2\psi_c(\mathbf{r})\sum_{c'\neq c}\chi_{cc'}N_0\phi_c(\mathbf{r}) \tag{12}$$

The governing Eqs. (8) can be rewritten using the new chemical potentials:

$$\frac{\partial\phi_c(\mathbf{r},t)}{\partial t} = \frac{\lambda R_e^5}{2\sqrt{\bar{\mathcal{N}}}k_BT}\left[\psi_c(\mathbf{r},t)\nabla^2\tilde{\mu}_c(\mathbf{r},t) - \tilde{\mu}_c(\mathbf{r},t)\nabla^2\psi_c(\mathbf{r},t)\right] \tag{13}$$

This concludes the continuum model employed in this work.

3. Solution strategy

In this section, we describe the different steps and methods we use to solve the continuum model in the previous section. Temporal and spatial discretization of (13) and techniques to overcome the inherent difficulties are discussed. Derivation and application of the PD diffusion operator concludes this section.

3.1. Numerical discretization

We start by temporal discretization of (13) using a forward-Euler time-discretization scheme:

$$\phi_c^{n+1}(\mathbf{r}) = \phi_c^n(\mathbf{r}) + \frac{\Delta t\lambda R_e^5}{2\sqrt{\bar{\mathcal{N}}}k_BT}\left[\psi_c^n(\mathbf{r})\nabla^2\tilde{\mu}_c^n(\mathbf{r}) - \tilde{\mu}_c^n(\mathbf{r})\nabla^2\psi_c^n(\mathbf{r})\right] \tag{14}$$

in which the superscripts n and $n+1$ are used to denote terms corresponding to time steps, respectively. For spatial discretization, we use the FSM. FSM usually provides exponential convergence and is rather easy to implement [33]. In this method, the derivatives are calculated using the characteristics of the Fourier transform [49]. These derivatives are usually much better approximations to the exact values, compared to local methods such as finite-difference method.

The computational domain is assumed to be a cuboid with dimensions $l_x\times l_y\times l_z$, discretized using a Cartesian grid with spacings Δx, Δy, and Δz, respectively. Periodic boundary conditions are applied in all directions, as commonly done in many continuum models. These boundary conditions are naturally satisfied in FSM, which is highly convenient. For the case of (14) and the chemical potentials (11) and (12), the sole differential operator involved is the Laplacian, which is straightforward to implement in FSM. Specifically, we replace $\nabla^2 f$, evaluated at all points of the domain, with the following expression:

$$\mathrm{FFT}^{-1}[-|\mathbf{q}|^2 \otimes \mathrm{FFT}[f]] \tag{15}$$

in which FFT[·], FFT^{-1}[·] and $\mathbf{q}$ are 3D fast Fourier transform (FFT), its inverse and the vector of FFT frequencies (or wavevector), respectively. $\otimes$ represents the point-wise multiplication operation. To evaluate the above expression, first f is evaluated on all nodes. Afterwards, a 3D FFT is performed and the result is pointwise multiplied with $-|\mathbf{q}|^2$. A 3D inverse FFT will calculate the approximate values of $\nabla^2 f$ in all nodes of the domain [49]. Interested readers may consult [49] for more detail about calculation of derivatives using FFT.

3.2. Numerical calculation of convolution terms in the chemical potentials

A convolution of two functions, u and v, in 3D space is generally defined as:

$$(u * v)(\mathbf{r}) = \int_{\mathbb{R}^3} u(\mathbf{r}-\mathbf{r}')v(\mathbf{r}')\,\mathrm{d}\mathbf{r}' \tag{16}$$

Such an integral can be computationally very demanding, especially if the values of the convolution are needed in every point in the domain, which is exactly the case with the convolution term in (11):

$$\mathscr{C}(\mathbf{r}) = \int \mathcal{G}(\mathbf{r}-\mathbf{r}')\left[f_B\sqrt{\phi_A}(\mathbf{r}') - \sqrt{f_A f_B}\sqrt{\phi_B}(\mathbf{r}')\right]\,\mathrm{d}\mathbf{r}' \tag{17}$$

Since the kernel $\mathcal{G}$ has finite support, the integration domain is confined to a sphere of radius $\xi_{\text{cut}} = 2R_{\text{e}}$ around $\mathbf{r}$. As the required resolution of the grid is approximately $R_{\text{e}}/10$, this translates to 33 401 evaluation of the integrand for each value of $\mathbf{r}$. An efficient way to calculate the term for all the points in the domain, is to use the *convolution theorem*:

$$(u * v)(\mathbf{r}) = \mathscr{F}^{-1}\{\mathscr{F}\{u\}\mathscr{F}\{v\}\} \tag{18}$$

in which $\mathscr{F}\{\cdot\}$ and $\mathscr{F}\{\cdot\}^{-1}$ denote the Fourier and inverse Fourier transforms, respectively. Accordingly, $\mathscr{C}(\mathbf{r})$ can be expressed as:

$$\mathscr{C}(\mathbf{r}) = \mathscr{F}^{-1}\{\mathscr{F}\{\mathcal{G}\}\mathscr{F}\{f_B\sqrt{\phi_A}(\mathbf{r}') - \sqrt{f_A f_B}\sqrt{\phi_B}(\mathbf{r}')\}\} \tag{19}$$

As we deal with discrete values in a numerical code, to evaluate $\mathscr{C}(\mathbf{r})$ in all points in the domain, one needs to calculate the following expression once:

$$\text{FFT}^{-1}[\text{FFT}[\mathcal{G}] \otimes \text{FFT}[f_B\sqrt{\phi_A} - \sqrt{f_A f_B}\sqrt{\phi_B}]] \tag{20}$$

Based on the definition of $\mathcal{G}$, FFT[$\mathcal{G}$] can be directly computed as:

$$\text{FFT}[\mathcal{G}](\mathbf{q}) = \frac{1}{|\mathbf{q}|^2 + \frac{1}{\xi_{\text{cut}}^2}} \tag{21}$$

which is very convenient, and reduces the needed computational resources compared to other numerical integration methods.

3.3. Numerical regularization

Numerical solutions almost always lead to a deviation from the exact solution, so called *error*. It can typically be decreased by finer numerical discretization, at the expense of computational resources, until it is in an acceptable range. There are special cases where a certain range of approximated values cause additional problems. For example, in the current work, negative densities, $\phi_c(\mathbf{r}) < 0$, are physically meaningless, and the chemical potentials are undefined for such values, because of the appearance of logarithms and square roots. The solution process cannot continue after the first occurrence of such an issue and thus these values need to be avoided. To alleviate this problem, we introduce a lightweight post-processing step, which is executed after calculation of the densities in each time step. In this process, negative density values are replaced such that the impact on the physics of the problem is minimal. The procedure for this *regularization* step is depicted in Algorithm 2. It avoids the negative densities by compensating these with mass from higher depth to keep the total mass of the system constant. We have observed that, among several different schemes tested for regularization steps, those who did not conserve the system mass rapidly lead to numerical instabilities and invalid results. Another argument for choosing the presented scheme was its optimal memory access. For examples, schemes that relied on compensating the negatives densities with *random* neighboring nodes, consumed much more computational resources due to enormous amount of cache misses.

3.4. Peridynamic diffusion operator

In this section, we provide a brief overview of the nonlocal formulation of PD for diffusion-type problems. By comparing it with its local counterpart, we aim to replace the diffusion operator in (8) with its nonlocal equivalent, leveraging the unique advantages of the nonlocal approach. For a comprehensive discussion on the derivation and physical interpretation of nonlocal diffusion problems, we refer the reader to [39,50]. It is worth noting that, since this section focuses on the PD formalism, we adopt its conventional notation — for instance, using $\mathbf{x}$ to represent the position vector, which is denoted as $\mathbf{r}$ in other sections.

To construct the PD model, we consider a region $\mathcal{H}_{\mathbf{x}}$ (Viz Fig. 1) around each point $\mathbf{x}$ in a body $\mathcal{R}$. The neighborhood is typically a ball in 3D (or a disc in 2D) centered at $\mathbf{x}$. At time t, a concentration value $\phi(\mathbf{x}, t)$ is associated with point $\mathbf{x}$. This neighborhood defines the points $\mathbf{x}'$ with which $\mathbf{x}$ interacts nonlocally. These interactions are facilitated through pipe-like conductors called *diffusion*

Algorithm 1 Interface detection algorithm

for $z \leftarrow 1, \dots, N_z/2$ **do** ▷ The other half is treated in a symmetric manner
  $\varrho_{P,z} \leftarrow 0$
  $\varrho_{G,z} \leftarrow 0$
  for $y \leftarrow 1, \dots, N_y; x \leftarrow 1, \dots, N_x$ **do**
    $\varrho_{P,z} \leftarrow \varrho_{P,z} + \rho_{A,z} + \rho_{B,z}$
    $\varrho_{G,z} \leftarrow \varrho_{G,z} + \rho_{G,z}$
  end for
  if $\varrho_{P,z} > \varrho_{G,z}$ **then**
    $z_{int} \leftarrow z$
    return z_{int}
  end if
end for

Algorithm 2 Regularization algorithm

Call Algorithm 1 ▷ Detect the location of the gas interface
for $z \leftarrow 1, \dots, N_z/2$ **do** ▷ The other half is treated in a symmetric manner
  for $y \leftarrow 1, \dots, N_y; x \leftarrow 1, \dots, N_x$ **do**
    if $\phi_{c,z} < 0$ **or** $z < z_{int}$ **then**
      $\phi_{c,z+1} \leftarrow \phi_{c,z} + \phi_{c,z+1}$
      $\phi_{c,z} \leftarrow 0$
    end if
  end for
end for
if $\phi_{c,N_z/2} < 0$ **then**
  $\phi_{c,N_z/2} \leftarrow 0$
end if

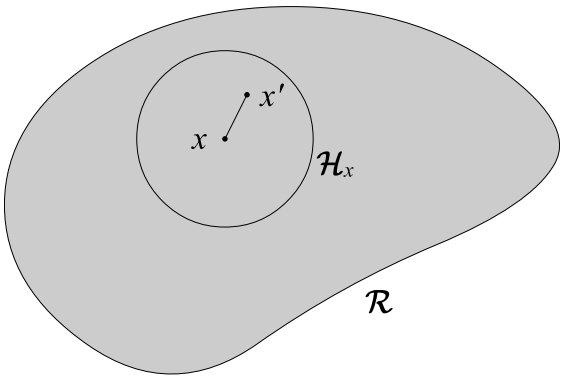

Fig. 1. Representation of the domain $\mathcal{R}$ and the horizon $\mathcal{H}_x$..

bonds, transferring concentration among points (buckets). In the bond-based PD formulation, transport along the bond $\boldsymbol{\xi} := \mathbf{x}' - \mathbf{x}$ is independent of other bonds' transport. Here, we adopt the bond-based PD formulation for our study. The neighborhood of point $\mathbf{x}$ is defined as:

$$\mathcal{H}_{\mathbf{x}} := \left\{ \mathbf{x}' \in \Omega : \|\mathbf{x}' - \mathbf{x}\| \leq \delta \right\} \tag{22}$$

where $\delta > 0$ is the PD horizon. Then governing equation of the corresponding nonlocal diffusion is written as:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \int_{\mathcal{H}_{\mathbf{x}}} J(\mathbf{x}', \mathbf{x}, t)\, \mathrm{d}V_{\mathbf{x}'} + s(\mathbf{x}, t), \quad \mathbf{x} \in \mathcal{B}, t > 0 \tag{23}$$

where J represents the kernel of the integral operator or the concentration flow density and $s(\mathbf{x}, t)$ is the source term. For a straightforward homogeneous diffusion, as proposed in [39], J can be taken as:

$$J(\mathbf{x}', \mathbf{x}, t) = \begin{cases} \kappa \frac{\phi(\mathbf{x}+\boldsymbol{\xi}, t) - \phi(\mathbf{x}, t)}{\|\boldsymbol{\xi}\|} & \|\boldsymbol{\xi}\| > 0 \\ 0 & \|\boldsymbol{\xi}\| = 0 \end{cases} \tag{24}$$

in which κ is recognized as the micro-diffusivity. In comparison, the local diffusion equation can be written as:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = \nabla \cdot [D \nabla \phi(\mathbf{x},t)] + s(\mathbf{x},t) = D \nabla^2 \phi(\mathbf{x},t) + s(\mathbf{x},t) \tag{25}$$

where D represents the diffusion coefficient of the medium (reflecting local Fickian diffusion). The primary objective is to approximate the Laplacian operator in the local diffusion equation using its nonlocal counterpart from (23), as inspired by (25). This substitution aims to exploit the advantages of the nonlocal operator in the transient equation of (13), particularly allowing for larger time steps in time discretization compared to traditional methods, such as the finite difference approach, which approximate the local operator.

Consider $\mathbf{x} \in \mathcal{H}_{\mathbf{x}}$, where a smoothly varying concentration field $\phi(\mathbf{x},t)$ exists. For distances $\|\boldsymbol{\xi}\| \leq \delta$, the concentration field can be approximated through a Taylor series expansion as:

$$\phi(\mathbf{x}',t) = \phi(\mathbf{x},t) + (\nabla \cdot \boldsymbol{\xi})\, \phi(\mathbf{x},t) + \frac{1}{2}(\nabla \cdot \boldsymbol{\xi})^2 \phi(\mathbf{x},t) + \mathcal{O}(\|\boldsymbol{\xi}\|^3) \tag{26}$$

By substituting the preceding equation into the integral term of (23), and employing the Einstein summation convention for repeated indices, we obtain:

$$\int_{\mathcal{H}_{\mathbf{x}}} \kappa \frac{\phi(\mathbf{x}',t) - \phi(\mathbf{x},t)}{\|\boldsymbol{\xi}\|} \, \mathrm{d}V_{\mathbf{x}'} = \left[\int_{\mathcal{H}} \kappa \frac{\xi_i}{\|\boldsymbol{\xi}\|} \, \mathrm{d}V_{\boldsymbol{\xi}} \right] \frac{\partial \phi(\mathbf{x},t)}{\partial x_i} + \frac{1}{2} \left[\int_{\mathcal{H}} \kappa \frac{\xi_i \xi_j}{\|\boldsymbol{\xi}\|} \, \mathrm{d}V_{\boldsymbol{\xi}} \right] \frac{\partial^2 \phi(\mathbf{x},t)}{\partial x_i \partial x_j} + \mathcal{O}(\delta^2) \tag{27}$$

noting that here a change in integration variables is applied and thus $\mathcal{H}$ denotes the spherical neighborhood with radius δ around the origin. The symmetrical nature of the integration process eliminates all antisymmetric terms in (27), leading to their cancellation. As a result, the primary contributors to the expansion are the second-order partial derivatives, while the fourth-order partial derivatives emerge as the next most significant terms impacting the truncation error. In the context of a 3D system, the bond vector $\boldsymbol{\xi}$ can be expressed in spherical coordinates as $\boldsymbol{\xi} = \langle \xi_x, \xi_y, \xi_z \rangle = r\langle \cos\theta \sin\varphi, \sin\theta \sin\varphi, \cos\varphi \rangle$. With this representation, the coefficients of the second-order partial derivatives can be conveniently arranged in a matrix form as follows:

$$\mathbf{M} = \frac{1}{2} \int_0^{\pi} \int_0^{2\pi} \int_0^{\delta} \begin{bmatrix} \cos^2\theta \sin^2\varphi & \sin\theta \cos\theta \sin^2\varphi & \cos\theta \sin\varphi \cos\varphi \\ & \sin^2\theta \sin^2\varphi & \sin\theta \sin\varphi \cos\varphi \\ \text{symm.} & & \cos^2\varphi \end{bmatrix} \kappa r^3 \sin\varphi \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}\varphi \tag{28}$$

Because of the integration limits, all off-diagonal elements in the matrix $\mathbf{M}$ are zero, resulting in identical values along the diagonal. By taking the PD micro-diffusivity as follows:

$$\kappa = \frac{6D}{\pi \delta^4} \tag{29}$$

the components of $\mathbf{M}$ become:

$$M_{ij} = \delta_{ij} D \tag{30}$$

where δ_{ij} denotes the Kronecker delta. As the horizon approaches zero, i.e., when $\delta \to 0$, the Laplacian operator linked to local diffusion is recovered, leading to the following expression:

$$\lim_{\delta \to 0} M_{ij} \frac{\partial^2 \phi(\mathbf{x},t)}{\partial x_i \partial x_j} = D \frac{\partial^2 \phi(\mathbf{x},t)}{\partial x_k \partial x_k} \tag{31}$$

There are various methods for spatial discretization in PD [51]. In this study, we adopt a widely-used approach that leverages a meshfree scheme introduced in [35]. We start by employing the same cuboid Cartesian grid as in the FSM, described in Section 3.1. In this configuration, each grid node represents a cell with a volume of $(\Delta x)^3$. As shown in Fig. 2, the node $\mathbf{x}_i$ interacts with all its neighboring nodes $\mathbf{x}_j$ within its local neighborhood $\mathcal{H}_{\mathbf{x}_i}$. Since we use periodic boundaries, periodicity is enforced in constructing the neighborhoods for nodes near the boundaries, where the neighborhood may partially lie outside the solution domain. To evaluate the integral over each neighborhood, we apply a one-point quadrature rule, with quadrature points at the nodes and quadrature weights determined by the volumes of the neighboring cells (Viz Fig. 2). The discretized form of the governing equation in (23) can then be expressed as follows:

$$\frac{\partial \phi(\mathbf{x}_i,t)}{\partial t} = \sum_{\mathbf{x}_j \in \mathcal{H}_{\mathbf{x}_i}} \boldsymbol{J}(\mathbf{x}_j, \mathbf{x}_i, t)\, V_j + s(\mathbf{x}_i,t) \tag{32}$$

where V_j is the volume associated with node $\mathbf{x}_j$. In light of (32), the PD approximation of the Laplacian of a generic scalar function f, calculated at node $\mathbf{x}_i$ can be expressed, taking $D = 1$ in (29), by:

$$\nabla^2 f(\mathbf{x}_i) \approx \frac{6}{\pi \delta^4} \sum_{\mathbf{x}_j \in \mathcal{H}_{\mathbf{x}_i}} \frac{f(\mathbf{x}_j) - f(\mathbf{x}_i)}{\|\mathbf{x}_j - \mathbf{x}_i\|} V_j \tag{33}$$

where this expression is used in the proposed approach to compute the necessary Laplacian operators appeared in (14), as detailed in the implementation section.

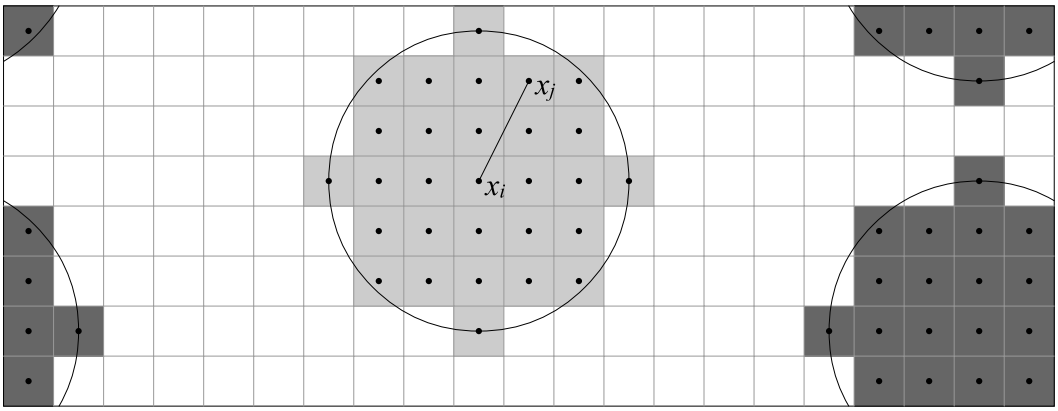

Fig. 2. A schematic representation of the discretization in the PD model, incorporating periodicity in neighborhood construction.

3.5. Stability analysis of the peridynamic diffusion operator

In this section, we perform a stability analysis of the PD diffusion operator and compare it to the traditional CFL condition. For clarity and simplicity, the analysis focuses on the 1D case. This study highlights the advantages of employing the PD diffusion operator.

The 1D PD diffusion equation can be expressed as:

$$\frac{\partial u(x,t)}{\partial t} = \int_{x-\delta}^{x+\delta} C(x'-x)[u(x',t)-u(x,t)]\mathrm{d}x' \tag{34}$$

in which $C(x'-x)$ is a coefficient that characterizes the conductivity of a bond. Using the forward Euler method for temporal discretization and the standard meshfree spatial discretization, the following equation can be derived:

$$\frac{u_k^{n+1}-u_k^n}{\Delta t} = \sum_{j=-m}^{m} C_j[u_{k+j}^n - u_k^n]\Delta x, \quad C_j := C(x_{k+j}-x_k) \tag{35}$$

where the superscripts and subscripts denote the time step and the node number, respectively. To ensure a physically admissible system, we assume:

$$C_{-j} = C_j \geq 0 \tag{36}$$

To perform a von Neumann stability analysis, the following ansatz is considered:

$$u_k^n = \zeta^n e^{\mathrm{i}pk}, \quad \zeta \in \mathbb{C} \tag{37}$$

in which $\mathrm{i}^2 = -1$, p represents the wave number, and the superscript in ζ^n denotes the exponent. Substituting the ansatz into (35) results in:

$$(\zeta^{n+1}-\zeta^n)e^{\mathrm{i}pk} = \zeta^n \Delta t \sum_{j=-m}^{m} C_j[e^{\mathrm{i}p(k+j)} - e^{\mathrm{i}pk}]\Delta x \tag{38}$$

Dividing both sides by $e^{\mathrm{i}pk} > 0$, the equation simplifies to:

$$\zeta - 1 = \Delta t \Delta x \sum_{j=-m}^{m} C_j(e^{ipj} - 1) \tag{39}$$

Employing the assumption (36), the above equation can be rewritten as:

$$\zeta = 1 + \Delta t \Delta x \sum_{j=-m}^{m} C_j \left[\frac{1}{2}(e^{\mathrm{i}pj} + e^{-\mathrm{i}pj}) - 1\right] \tag{40}$$

Using the definition of the complex cosine function, the equation transforms into:

$$\zeta = 1 + \Delta t \Delta x \sum_{j=-m}^{m} C_j[\cos(pj) - 1] \tag{41}$$

To avoid numerical instability, the condition $|\zeta| \leq 1$ must be satisfied, leading to:

$$-1 \leq 1 + \Delta t \Delta x \sum_{j=-m}^{m} C_j[\cos(pj) - 1] \tag{42}$$

Simplifying further:

$$\Delta t \leq \frac{2}{\Delta x \sum_{j=-m}^{m} C_j [1 - \cos(pj)]} \tag{43}$$

Since $\cos(pj) \geq -1$, the stability criterion is satisfied when:

$$\Delta t \leq \frac{1}{\Delta x \sum_{j=-m}^{m} C_j} \tag{44}$$

which is the stability condition for (35). To compare this with the CFL condition, we analyze the stability criterion in (44) for a specific choice of C_j. It can be shown that C_j can be related to D as [39,50]:

$$C_j = \begin{cases} \frac{2D}{m^2 \Delta x^2 |x_{k+j} - x_k|} = \frac{2D}{m^2 \Delta x^3 |j|} & j \neq 0 \\ 0 & j = 0 \end{cases} \tag{45}$$

Substituting C_j into (44), the stability condition simplifies to:

$$\Delta t \leq \frac{1}{\Delta x \sum_{j=-m}^{m} \frac{2D}{m^2 \Delta x^3 |j|}} = \frac{m^2 \Delta x^2}{4D \sum_{j=1}^{m} \frac{1}{j}} = \frac{\Delta x^2}{2D} \frac{m^2}{2H_m} \tag{46}$$

where $\frac{\Delta x^2}{2D}$ is the CFL stability condition and H_m denotes the mth *harmonic number*. It is evident that:

$$\frac{m^2}{2H_m} > 1, \quad \forall m \in \mathbb{N}, m > 1 \tag{47}$$

Thus, for any PD diffusion operator with $m > 1$, the stability constraint is less restrictive than the CFL condition. Furthermore, as the nonlocality increases:

$$\lim_{m \to \infty} \frac{m^2}{2H_m} \to \infty \tag{48}$$

indicating that the condition becomes progressively more relaxed with larger values of m. Fig. 3 shows the growth rate for $1 \leq m \leq 10$.

Remark 1. It is important to note that, while (48) suggests that increasing m leads to a less restrictive stability condition, the time step size Δt in time discretization cannot be arbitrarily large due to the resulting increase in the local truncation error of the explicit scheme. Furthermore, a larger m increases the number of nodes within the PD horizon, which can make the approach computationally more expensive. In this work, we choose $m = 3$ for two main reasons: (i) To ensure that the nonlocal approximation does not deviate significantly from the local approximation of the Laplacian operator, and (ii) to minimize the computational cost of the PD approach, particularly the cost associated with spatial integration.

Remark 2. In this work, we employ the well-established classical PD diffusion operator originally introduced in [39] due to its simplicity and suitability for uniform grids. However, for simulations involving unstructured grids or complex geometries, alternative nonlocal operators such as Dual-Horizon Peridynamics (DH-PD) [52], the Nonlocal Operator Method (NOM) [53], and the Peridynamic Differential Operator (PDDO) [54–56] could be explored. Future studies incorporating these methods may further extend the applicability of this approach.

4. Implementation

The solver has been developed in standard C++20, and can be compiled on any platform with such a compiler. The code depends only on two libraries, `FFTW` for FFT calculations, and `HDF5` for (parallel) output of results. Both of these libraries are very well known among HPC community for real-world, high-performance application, and are available for many different platforms.

We leverage modern multi-core CPUs, by parallelizing the solver using OpenMP. This type of parallelization can be very effective on multi-core CPUs. However, due to the memory-bound nature of the code a linear speedup is not expected. This is especially true if the code is run on HPC hardware, which are usually equipped with multiple CPU sockets, each with several cores. We will investigate this later in Section 5. In C, C++ and Fortran, the parallelization is done by adding *OpenMP directives* to the loops in the code, which enable the compiler to generate *shared memory* parallel machine instructions. As any other type of parallel program, great care must be taken to avoid *data race*, a situation in which the results are nondeterministic and depend on the (indeterminate) execution order of different threads. Such cases can occur for example in *reduction operations* or when the loop iterations are inter-dependent.

The performance of the solver was tested on a single node in the *Strand* cluster in Helmholtz-Zentrum, Hereon. The node contains two Intel Xeon Platinum 8160 CPU at 2.10 GHz and 384 GB of DDR4 RAM, running CentOS 7. Also, `g++` 11.1.1 was used to compile the code. Algorithms 1–4 list the steps taken to solve the various parts of the problem, both for forward-Euler FSM and proposed methods.

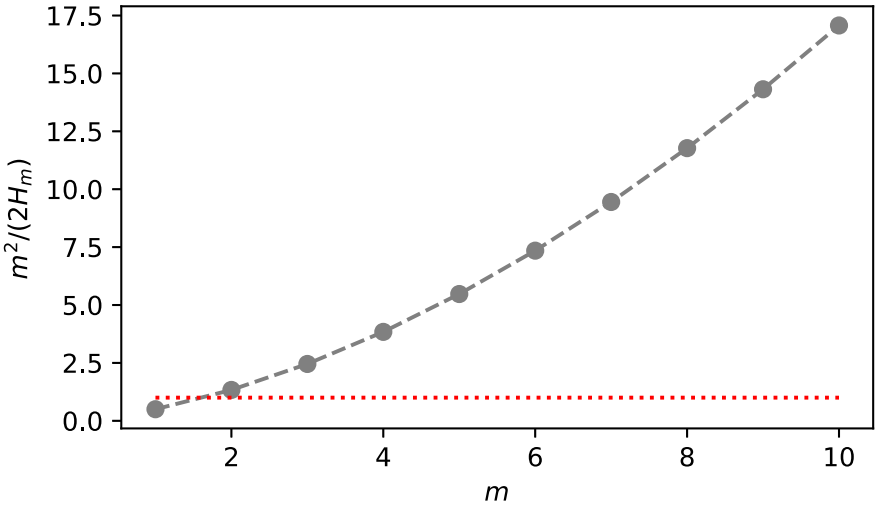

Fig. 3. Growth rate of the term $\frac{m^2}{2H_m}$ for $1 \leq m \leq 10$. The horizontal line represents $\frac{m^2}{2H_m} = 1$. For $m > 1$, the nonlocal operator exhibits a more relaxed stability criterion compared to the CFL condition.

Algorithm 3 Solution algorithm for the reference solution

Read the input parameters
Initialize required data structures
Initialize `FFTW` and `HDF5`
Apply the initial boundary conditions
$n \leftarrow 0$
while $n < n_{max}$ **do**
    Calculate the FFT of ψ_c^n, expressions needed for calculation of convolution terms and $\sum_c \phi_c^n - 1$
    Calculate $\nabla^2 \psi_c^n$, convolution terms (20) and $\nabla^2 \sum_c \phi_c^n - 1$
    Calculate the μ_c^n using (11) and (12) and their FFT
    Calculate the $\nabla^2 \mu_c^n$ using (15)
    Calculate ϕ_c^{n+1} using (14)
    Call Algorithm 2 ▷ Regularization step
    $n \leftarrow n + 1$
end while
Save results

Algorithm 4 Solution algorithm for the proposed method

Read the input parameters
Initialize required data structures
Initialize `FFTW` and `HDF5`
Apply the initial boundary conditions
$n \leftarrow 0$
while $n < n_{max}$ **do**
    Calculate the FFT of ψ_c^n, expressions needed for calculation of convolution terms and $\sum_c \phi_c^n - 1$
    Calculate convolution terms (20) and $\nabla^2 \sum_c \phi_c^n - 1$
    Calculate the μ_c^n using (11) and (12)
    Calculate $\nabla^2 \psi_c^n$ and $\nabla^2 \mu_c^n$ using PD diffusion operator (33)
    Calculate ϕ_c^{n+1} using (14)
    Call Algorithm 2 ▷ Regularization step
    $n \leftarrow n + 1$
end while
Save results

5. Numerical examples

In this section, we will examine the performance of the proposed method. First, we validate the model using the well-known phase diagram for a copolymer melt. Secondly, we investigate the performance of the proposed method in case of an EISA process.

5.1. Example 1: order–disorder transition test for copolymer melts

In this example, we validate the proposed method by testing the order–disorder transition of a neat diblock copolymer melt. Depending on the copolymer's block composition, f_A, and the interaction, $\chi_{AB}N_0$, the diblock copolymer may distribute homogeneously in the system or microphase separate. In the latter case, it takes on a number of distinct morphologies, such as lamellae, cylinders and spheres. Microphase separation occurs when the interaction is high, such that the free-energy decrease by separating the repelling blocks of the copolymer overcomes the loss in translational entropy. The critical interaction strength, above which an initially homogeneous system is unstable and phase separates is given by the spinodal lines in the f_A-$\chi_{AB}N$-phase diagram. It can be analytically calculated from SCFT using RPA [3]. The here-used free-energy functional [31,32] matches this spinodal line exactly and is therefore well suited to test the validity of the software. Fig. 4, shows the spinodal line as obtained by RPA. Below the spinodal line, the fluctuation about the mean concentration are expected to decay in amplitude. Above the spinodal line, fluctuations, approximately on the length scale of the mean end-to-end distance of the polymer, R_e, grow in amplitude. From these, the equilibrium morphology forms, that minimizes the free-energy functional, and the system becomes stationary. The choice of morphology is the result of the balance of surface tension, proportional to the interface area of each morphology, and the entropically disfavored stretching of chains associated with each morphology. In principle, lamellae are stable for symmetric copolymers, $f_A \approx 0.5$, cylinders that arrange in a hexagonal lattice for slight compositional asymmetries, and spheres that arrange in a bcc lattice for strong asymmetries. The detailed phase diagram, as calculated from RPA [3], SCFT [48,57], and for the UDM [26] involves globally minimizing the free-energy functional. In this work, the time evolution of the concentration field, which locally conserves concentration, only locally minimizes the free-energy. Hence, the simulations typically get stuck in a local free-energy minimum – a metastable structure. In this case, morphologies have defects or are unidentifiable which makes a comparison with the equilibrium phase diagram impossible. Here we restrict ourselves to the weak-segregation region, slightly above the spinodal line, where free-energy barriers are small and the equilibrium morphology emerges [58].

For a variety of block ratios, f_A, we test the phase behavior by initializing a homogeneous morphology with small fluctuations, $\phi_A(\mathbf{r}, t=0) = f_A[1+\sigma_A(\mathbf{r})]$ and $\phi_B(\mathbf{r}, t=0) = f_B[1+\sigma_B(\mathbf{r})]$ where $\sigma_A(\mathbf{r})$ and $\sigma_B(\mathbf{r})$ are independent random numbers drawn from the uniform distribution $[-10^{-2}, 10^{-2}]$. We take a system of size $V = 5.542R_\mathrm{e} \times 5.542R_\mathrm{e} \times 1.7R_\mathrm{e}$ on a discrete grid with $56 \times 56 \times 17$ cells and propagate at a time step size of $\Delta t = 2.5 \times 10^{-5}\lambda^{-1}$ to time $t = 50\lambda^{-1}$. We use $\delta/\Delta x = m = 3$ for the PD diffusion operator. At each block composition, we choose two interaction strengths, slightly below and slightly above the spinodal line. The resulting final morphologies are indicated as data points in Fig. 4. Below the spinodal line, we consistently observe homogeneous morphologies, as indicated by the hollow circles in Fig. 4. Above the spinodal we observe the lamellar, cylindrical and spherical morphologies, as indicated by the filled markers. Their location is qualitatively consistent with the known phase diagram for diblock copolymers, [23,26]. Fig. 5 visualizes the resulting non-homogeneous morphologies for $f_A = 0.3$, 0.45 and 0.5. All simulations start out with an exponential growth of small fluctuations. Only after these have grown in amplitude and non-linearities gain influence, ordering into their respective morphology occurs, *i.e.*, spheres for $f_A = 0.3$, cylinders for $f_A = 0.45$ and lamellae for $f_A = 0.5$. It should be noted that there is no scale separation between the periodicity of the morphology and the width of the internal AB interfaces for weak repulsion, which simplifies the computation.

5.2. Example 2: self-assembly of diblock copolymers upon solution evaporation

Integral asymmetric polymer filter membranes are functional macromolecular systems that have a wide range of applications including water purification and catalysis [59,60]. Their synthesis is a two-step procedure consisting of (i) solvent evaporation from a thin film of diblock-copolymer solution and (ii) solvent-non-solvent exchange, accompanied by a freezing of the self-assembled top-layer. This NIPS step creates a sponge-like substructure that provides the membrane with mechanical stability. Here, we simulate the first step, the so-called EISA process, which has been simulated in detail using a range of numerical methods [15,16,18,19], following the approach in [17], where the UDM was previously used.

This example simulation involves five components, the AB diblock copolymer, a gas phase, that is macroscopically phase-separated from the solution to mimic a liquid-vapor interface, a volatile solvent that diffuses from the initially stable homogeneous solution into the gas phase, and a non-volatile solvent that remains in the solution. To mimic the evaporation process, the volatile solvent is converted into the gas phase at the top of the simulation domain. This leads to a retraction of the liquid-vapor interface and an enrichment of the non-volatile diblock copolymer at the top of the solution — formation of a polymer skin. The blocks of the diblock copolymer experience a strong repulsion from each other that is initially mediated by the solvents. Inside the polymer skin the polymer concentration surpasses the critical concentration for microphase separation and micelles form. As the solvent evaporates the region where microphase separation occurs becomes larger and the micelles elongate to form cylinders that are oriented perpendicularly to the interface. This is the case when the minority block, A, has a stronger repulsion towards the gas phase than the majority block does, as has been shown in [17,19]. Here, we use this principle to choose our model parameters.

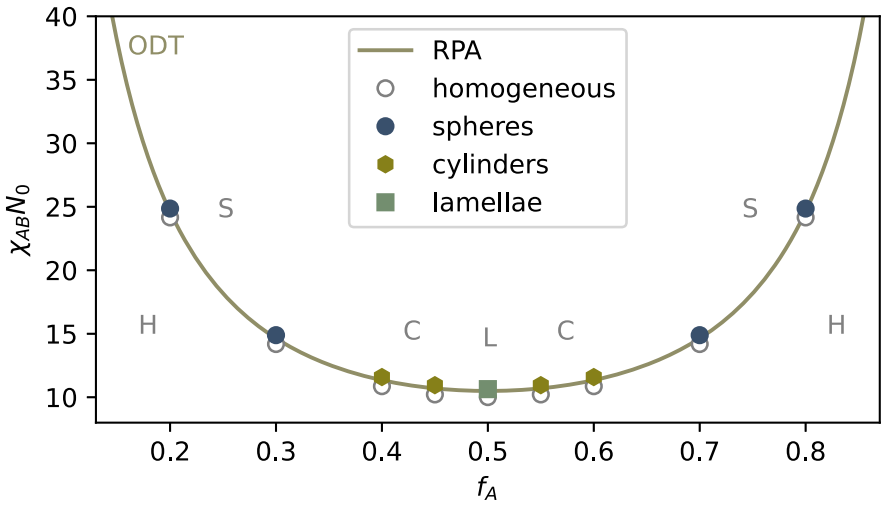

Fig. 4. Phase diagram for diblock copolymer melt in Example 1: The diagram indicates the stability of different morphologies — homogeneous (H), bcc spheres (S), hexagonal cylinders (C) and lamellae (L). The ODT and OOT are calculated by minimizing the free energy per molecule of the respective unit cell. The dashed line represents the theoretical spinodal for the ODT, as obtained by RPA. The proposed method is tested against this curve by starting simulations from a homogeneous initial condition and resulting morphologies are indicated as stated in the legend.

To simulate the EISA process, we take a diblock copolymer with block composition, $f_A = 0.3125$. The matrix of Flory–Huggins parameters is given by

$$\chi N_0 = \begin{array}{c} \begin{array}{ccccc} A & B & S & C & G \end{array} \\ \begin{pmatrix} - & 32 & 16 & 0 & 100 \\ & - & 0 & 16 & 80 \\ & & - & 0 & 0 \\ & & & - & 80 \\ symm. & & & & - \end{pmatrix} \end{array} \begin{array}{c} A \\ B \\ S \\ C \\ G \end{array} \tag{49}$$

Thereby the blocks of the copolymer strongly repel each other. The minority block, A, has a repulsion towards the volatile solvent, S, while the majority block, B, has a strong repulsion towards the non-volatile solvent, C. All non-volatile species, A, B, and C experience a strong repulsion towards the gas phase G, to establish the liquid-vapor interface. In the proposed method for the PD diffusion operator, we set the ratio $\delta/\Delta x = m = 3$. To ensure a fair comparison of performance between the proposed method and FSM, we select the largest stable time steps for each approach, with $\Delta t_{\text{PD}} = 10^{-5}\lambda^{-1}$ for the proposed method and $\Delta t_{\text{FSM}} = 10^{-6}\lambda^{-1}$ for FSM. Both methods are used to simulate the same problem up to a specified final time t.

The simulation results are shown in Figs. 7 and 8. One readily observes the expected self-assembly of the diblock copolymers at the interface into cylinders that are oriented perpendicularly to the solution-gas interface. In addition, the proposed method demonstrates a performance improvement, outperforming forward-Euler FSM by a factor of 6.41. This result highlights the efficiency of the proposed approach and corroborates the conclusions presented in Section 3.5.

To assess the scaling behavior of the proposed method, we perform simulations with varying numbers of OpenMP threads, as illustrated in Fig. 6. The speedup curve deviates increasingly from the ideal linear trend (represented by the dotted line) as the thread count rises, indicating that the method is memory-bound. However, the comparison between 24 and 48 threads (i.e. one versus two CPU sockets) suggests that the memory controller has not yet reached saturation.

6. Conclusion

This work introduces the groundwork for the integration of a PD-enhanced Fourier spectral method (FSM) within continuum-model frameworks of molecular self-assembly. Here, we used the Uneyama-Doi free energy, [17,27,32], describing block copolymer blends and solutions, that effectively captures both short- and long-range interactions. Applying the PD-principles to the model-B dynamics, we demonstrate the numerical viability of the method. We show that the incorporation of a PD-based diffusion operator allows for larger time steps without compromising numerical stability or accuracy. This overcomes the limitations of traditional explicit schemes, reducing computational demands and making experimental time scales accessible. The proposed method offers several key advantages: It improves computational efficiency, achieving performance gains of over sixfold in specific test cases compared to conventional FSM approaches. Demonstrations of the method's effectiveness, including the order–disorder transition and evaporation-induced self-assembly (EISA) process, highlight its versatility in capturing the diverse morphological behaviors. This framework not only advances the study of polymer systems but also provides a valuable tool for addressing similar systems with model-B dynamics. Future research could be used to accelerate problems with similar characteristics.

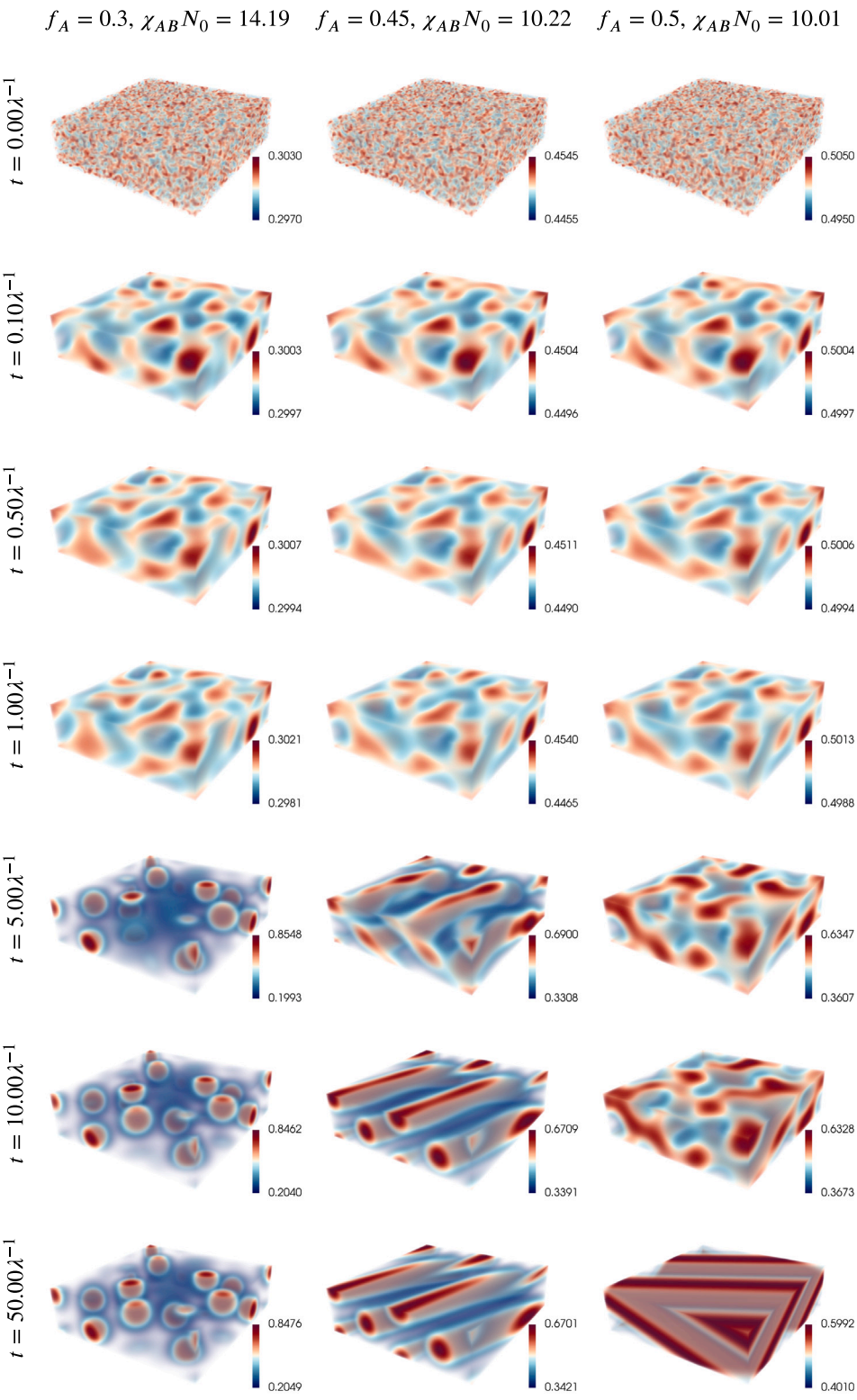

Fig. 5. Time evolution of minority block density ϕ_A in Example 1: (left) $f_A = 0.3$, $\chi_{AB}N_0 = 14.19$, (middle) $f_A = 0.45$, $\chi_{AB}N_0 = 10.22$, (right) $f_A = 0.5$, $\chi_{AB}N_0 = 10.01$.

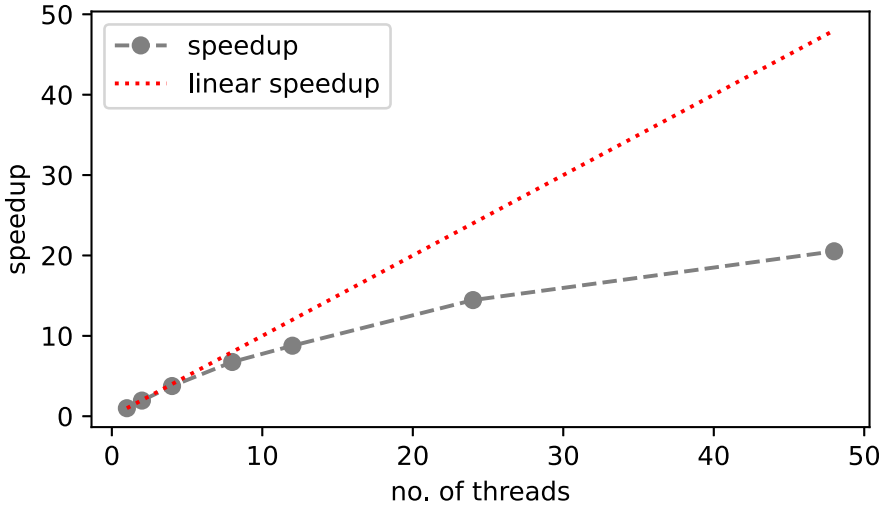

Fig. 6. Speedup of the proposed method over the serial code in Example 2; the dotted line depicts linear speedup.

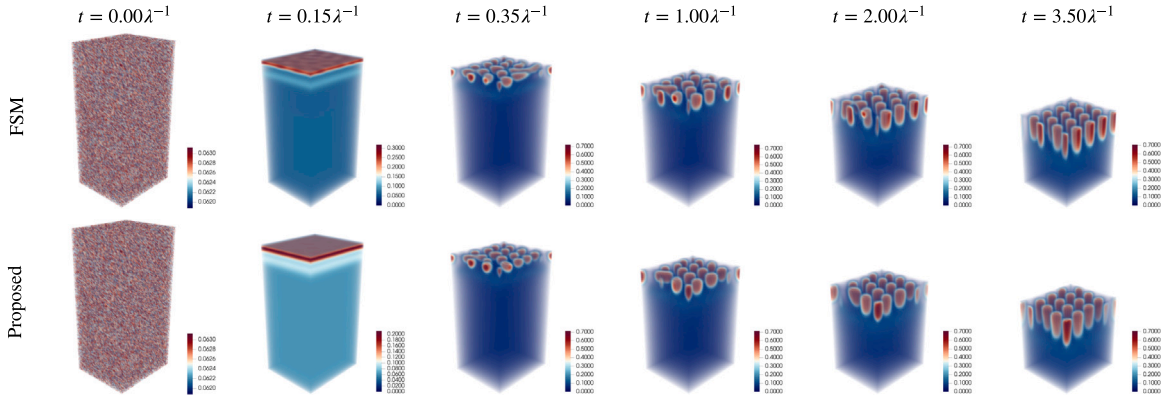

Fig. 7. Time evolution of minority block density ϕ_A in Example 2: (top) FSM, (bottom) Proposed.

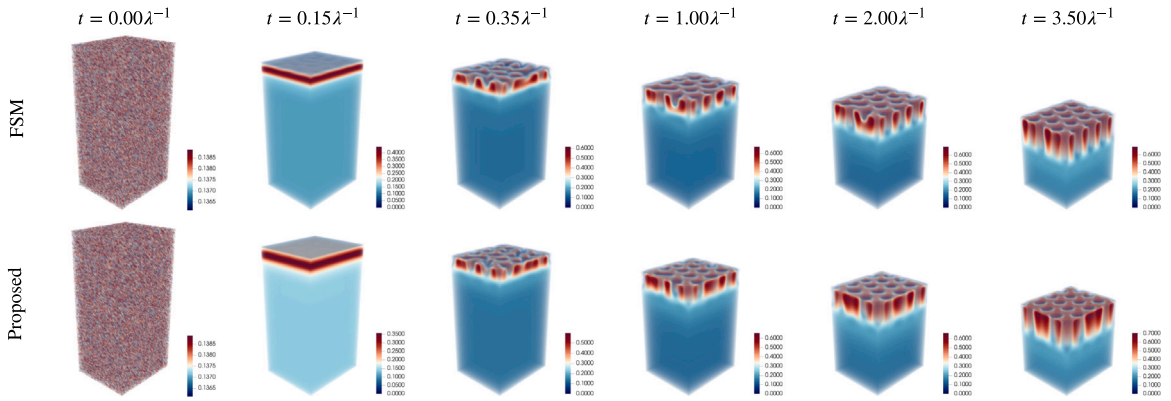

Fig. 8. Time evolution of majority block density ϕ_B in Example 2: (top) FSM, (bottom) Proposed.

CRediT authorship contribution statement

Farshid Mossaiby: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Gregor Häfner:** Writing – review & editing, Writing – original

draft, Visualization, Validation, Software, Investigation. **Arman Shojaei:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Alexander Hermann:** Writing – review & editing, Writing – original draft, Methodology, Investigation. **Christian Cyron:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Marcus Müller:** Writing – review & editing, Validation, Project administration, Funding acquisition. **Stewart Silling:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used OpenAI ChatGPT in order to improve language and readability of the text. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Financial support was provided by the Bundesministerium für Bildung und Forschung (BMBF) within the project 16ME0658K MExMeMo as well as by the European Union – NextGenerationEU. Note that the above content expresses solely the opinion of the author(s) and not necessarily the one of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for the above content. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 470246804.

This article has been authored by an employee of National Technology and Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/downloads/doe-public-access-plan.

Data availability

Data will be made available on request.

References

[1] M. Karayianni, S. Pispas, Block copolymer solution self-assembly: recent advances, emerging trends, and applications, J. Polym. Sci. 59 (17) (2021) 1874–1898, http://dx.doi.org/10.1002/pol.20210430.
[2] W. Zhang, Q. Zhao, J. Yuan, Porous polyelectrolytes: the interplay of charge and pores for new functionalities, Angew. Chem. Int. Ed. 57 (23) (2018) 6754–6773, http://dx.doi.org/10.1002/anie.201710272.
[3] L. Leibler, Theory of microphase separation in block copolymers, Macromolecules 13 (6) (1980) 1602–1617, http://dx.doi.org/10.1021/ma60078a047.
[4] F.S. Bates, G.H. Fredrickson, Block copolymer thermodynamics: Theory and experiment, Annu. Rev. Phys. Chem. 41 (1) (1990) 525–557, URL http://www.annualreviews.org/doi/10.1146/annurev.pc.41.100190.002521.
[5] C.M. Bates, F.S. Bates, 50Th anniversary perspective: Block polymers-pure potential, Macromolecules 50 (1) (2017) 3–22, http://dx.doi.org/10.1021/acs.macromol.6b02355.
[6] P.K. Janert, M. Schick, Phase behavior of ternary homopolymer/diblock blends: Microphase unbinding in the symmetric system, Macromolecules 30 (13) (1997) 3916–3920.
[7] M.W. Matsen, Gyroid versus double-diamond in ABC triblock copolymer melts, J. Chem. Phys. 108 (2) (1998) 785–796.
[8] V. Abetz, K. Kremer, M. Müller, G. Reiter, Functional macromolecular systems: kinetic pathways to obtain tailored structures, Macromol. Chem. Phys. 220 (2) (2019) 1800334, http://dx.doi.org/10.1002/macp.201800334.
[9] S. Chandran, J. Baschnagel, D. Cangialosi, K. Fukao, E. Glynos, L.M.C. Janssen, M. Müller, M. Muthukumar, U. Steiner, J. Xu, S. Napolitano, G. Reiter, Processing pathways decide polymer properties at the molecular level, Macromolecules 52 (19) (2019) 7146–7156, http://dx.doi.org/10.1021/acs.macromol.9b01195.
[10] M. Müller, Process-directed self-assembly of copolymers: Results of and challenges for simulation studies, Prog. Polym. Sci. 101 (2020) 101198, http://dx.doi.org/10.1016/j.progpolymsci.2019.101198.
[11] H.-C. Kim, S.-M. Park, W.D. Hinsberg, Block copolymer based nanostructures: materials, processes, and applications to electronics, Chem. Rev. 110 (1) (2010) 146–177, http://dx.doi.org/10.1021/cr900159v.
[12] M. Müller, V. Abetz, Nonequilibrium processes in polymer membrane formation: theory and experiment, Chem. Rev. 121 (22) (2021) 14189–14231, http://dx.doi.org/10.1021/acs.chemrev.1c00029.
[13] K.-V. Peinemann, V. Abetz, P.F.W. Simon, Asymmetric superstructure formed in a block copolymer via phase separation, Nat. Mater. 6 (12) (2007) 992–996, http://dx.doi.org/10.1038/nmat2038.
[14] S.H. Kim, M.J. Misner, T. Xu, M. Kimura, T.P. Russell, Highly oriented and ordered arrays from block copolymers via solvent evaporation, Adv. Mater. 16 (3) (2004) 226–231.

[15] S.P. Paradiso, K.T. Delaney, C.J. GarcÃa-Cervera, H.D. Ceniceros, G.H. Fredrickson, Block copolymer self assembly during rapid solvent evaporation: insights into cylinder growth and stability, ACS Macro Lett. 3 (1) (2014) 16–20, http://dx.doi.org/10.1021/mz400572r.
[16] A.V. Berezkin, C.M. Papadakis, I.I. Potemkin, Vertical domain orientation in cylinder-forming diblock copolymer films upon solvent vapor annealing, Macromolecules 49 (1) (2016) 415–424, http://dx.doi.org/10.1021/acs.macromol.5b01771.
[17] O. Dreyer, G. Ibbeken, L. Schneider, N. Blagojevic, M. Radjabian, V. Abetz, M. Müller, Simulation of solvent evaporation from a diblock copolymer film: orientation of the cylindrical mesophase, Macromolecules 55 (17) (2022) 7564–7582, http://dx.doi.org/10.1021/acs.macromol.2c00612.
[18] N. Blagojevic, M. Müller, Simulation of membrane fabrication via solvent evaporation and nonsolvent-induced phase separation, ACS Appl. Mater. Interfaces 15 (50) (2023-12-20) 57913–57927, http://dx.doi.org/10.1021/acsami.3c03126.
[19] N. Blagojevic, S. Das, J. Xie, O. Dreyer, M. Radjabian, M. Held, V. Abetz, M. Müller, Toward predicting the formation of integral-asymmetric, isoporous diblock copolymer membranes, Adv. Mater. n/a (n/a) (2024) 2404560, http://dx.doi.org/10.1002/adma.202404560.
[20] M. Müller, Studying amphiphilic self-assembly with soft coarse-grained models, J. Stat. Phys. 145 (4) (2011) 967–1016, http://dx.doi.org/10.1007/s10955-011-0302-z.
[21] K.C. Daoulas, M. Müller, Single chain in mean field simulations: quasi-instantaneous field approximation and quantitative comparison with Monte Carlo simulations, J. Chem. Phys. 125 (18) (2006) 184904, http://dx.doi.org/10.1063/1.2364506.
[22] E. Helfand, Theory of inhomogeneous polymers: fundamentals of the Gaussian random-walk model, J. Chem. Phys. 62 (3) (1975) 999–1005, http://dx.doi.org/10.1063/1.430517.
[23] M.W. Matsen, M. Schick, Stable and unstable phases of a diblock copolymer melt, Phys. Rev. Lett. 72 (16) (1994) 2660–2663, http://dx.doi.org/10.1103/PhysRevLett.72.2660.
[24] M. Müller, J.C.O. Rey, Continuum models for directed self-assembly, Mol. Syst. Des. Eng. 3 (2) (2018) 295–313, http://dx.doi.org/10.1039/C7ME00109F.
[25] T. Ohta, K. Kawasaki, Equilibrium morphology of block copolymer melts, Macromolecules 19 (10) (1986) 2621–2632.
[26] T. Uneyama, M. Doi, Density functional theory for block copolymer melts and blends, Macromolecules 38 (1) (2005) 196–205.
[27] T. Uneyama, M. Doi, Calculation of the micellar structure of polymer surfactant on the basis of the density functional theory, Macromolecules 38 (13) (2005) 5817–5825, http://dx.doi.org/10.1021/ma0507746.
[28] H. Qi, C. Zhong, Density functional theory studies on the microphase separation of amphiphilic comb copolymers in a selective solvent, J. Phys. Chem. B 112 (35) (2008) 10841–10847, http://dx.doi.org/10.1021/jp0774950.
[29] P.C. Hohenberg, B.I. Halperin, Theory of dynamic critical phenomena, Rev. Modern Phys. 49 (3) (1977) 435.
[30] T. Uneyama, Density functional simulation of spontaneous formation of vesicle in block copolymer solutions, J. Chem. Phys. 126 (11) (2007).
[31] G. Häfner, M. Müller, Reaction-driven assembly: Controlling changes in membrane topology by reaction cycles, Soft Matter 19 (38) (2023) 7281–7292, http://dx.doi.org/10.1039/D3SM00876B.
[32] G. Häfner, M. Müller, Reaction-driven diffusiophoresis of liquid condensates: potential mechanisms for intracellular organization, ACS Nano (2024) http://dx.doi.org/10.1021/acsnano.3c12842.
[33] S. Biner, Programming Phase-Field Modeling, Springer International Publishing, 2018, URL https://books.google.com/books?id=h6nOugEACAAJ.
[34] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, J. Mech. Phys. Solids 48 (1) (2000) 175–209.
[35] S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics, Comput. Struct. 83 (17–18) (2005) 1526–1535.
[36] E. Lees, S. Rokkam, S. Shanbhag, M. Gunzburger, The electroneutrality constraint in nonlocal models, J. Chem. Phys. 147 (12) (2017).
[37] F. Bobaru, M. Duangpanya, A peridynamic formulation for transient heat conduction in bodies with evolving discontinuities, J. Comput. Phys. 231 (7) (2012) 2764–2785.
[38] S.A. Silling, R.B. Lehoucq, Peridynamic theory of solid mechanics, Adv. Appl. Mech. 44 (2010) 73–168.
[39] F. Bobaru, M. Duangpanya, The peridynamic formulation for transient heat conduction, Int. J. Heat Mass Transfer 53 (19–20) (2010) 4047–4059.
[40] P. Diehl, S. Prudhomme, M. Lévesque, A review of benchmark experiments for the validation of peridynamics models, J. Peridynamics Nonlocal Model. 1 (2019) 14–35.
[41] C. Tamur, S. Li, A bond-based peridynamics modeling of polymeric material fracture under finite deformation, Comput. Methods Appl. Mech. Engrg. 414 (2023) 116132.
[42] F.S. Vieira, A.L. Araújo, Peridynamic topology optimization to improve fracture resistance of structures, Comput. Methods Appl. Mech. Engrg. 433 (2025) 117455.
[43] A. Shojaei, A. Hermann, P. Seleson, S.A. Silling, T. Rabczuk, C.J. Cyron, Peridynamic elastic waves in two-dimensional unbounded domains: Construction of nonlocal Dirichlet-type absorbing boundary conditions, Comput. Methods Appl. Mech. Engrg. 407 (2023) 115948.
[44] G. Ongaro, A. Shojaei, F. Mossaiby, A. Hermann, C.J. Cyron, P. Trovalusci, Multi-adaptive spatial discretization of bond-based peridynamics, Int. J. Fract. 244 (1) (2023) 1–24.
[45] G. Ongaro, P. Seleson, U. Galvanetto, T. Ni, M. Zaccariotto, Overall equilibrium in the coupling of peridynamics and classical continuum mechanics, Comput. Methods Appl. Mech. Engrg. 381 (2021) 113515.
[46] A. Hermann, A. Shojaei, D. Höche, S. Jafarzadeh, F. Bobaru, C.J. Cyron, Nonlocal nernst-Planck-Poisson system for modeling electrochemical corrosion in biodegradable magnesium implants, J. Peridynamics Nonlocal Model. 7 (1) (2025) 1–32.
[47] C. Tian, S. Fan, J. Du, Z. Zhou, Z. Chen, F. Bobaru, A peridynamic model for advection–reaction–diffusion problems, Comput. Methods Appl. Mech. Engrg. 415 (2023) 116206.
[48] M.W. Matsen, F.S. Bates, Unifying weak- and strong-segregation block copolymer theories, Macromolecules 29 (4) (1996) 1091–1098, http://dx.doi.org/10.1021/ma951138i.
[49] S.G. Johnson, Notes on FFT-based differentiation, in: MIT Applied Mathematics, Tech. Rep., 2011.
[50] A. Shojaei, A. Hermann, P. Seleson, C.J. Cyron, Dirichlet absorbing boundary conditions for classical and peridynamic diffusion-type models, Comput. Mech. 66 (2020) 773–793.
[51] A. Shojaei, A. Hermann, C.J. Cyron, P. Seleson, S.A. Silling, A hybrid meshfree discretization to improve the numerical performance of peridynamic models, Comput. Methods Appl. Mech. Engrg. 391 (2022) 114544.
[52] H. Ren, X. Zhuang, T. Rabczuk, Dual-horizon peridynamics: A stable solution to varying horizons, Comput. Methods Appl. Mech. Engrg. 318 (2017) 762–782.
[53] H. Ren, X. Zhuang, T. Rabczuk, Nonlocal operator method with numerical integration for gradient solid, Comput. Struct. 233 (2020) 106235.
[54] E. Madenci, A. Barut, M. Futch, Peridynamic differential operator and its applications, Comput. Methods Appl. Mech. Engrg. 304 (2016) 408–451.
[55] A. Shojaei, U. Galvanetto, T. Rabczuk, A. Jenabi, M. Zaccariotto, A generalized finite difference method based on the peridynamic differential operator for the solution of problems in bounded and unbounded domains, Comput. Methods Appl. Mech. Engrg. 343 (2019) 100–126.
[56] S. Bazazzadeh, A. Shojaei, M. Zaccariotto, U. Galvanetto, Application of the peridynamic differential operator to the solution of sloshing problems in tanks, Eng. Comput. 36 (1) (2019) 45–83.
[57] M.W. Matsen, T.M. Beardsley, J.D. Willis, Fluctuation-corrected phase diagrams for diblock copolymer melts, Phys. Rev. Lett. 130 (24) (2023) 248101, http://dx.doi.org/10.1103/PhysRevLett.130.248101.
[58] W. Li, P.F. Nealey, J.J. de Pablo, M. Müller, Defect removal in the course of directed self-assembly is facilitated in the vicinity of the order-disorder transition, Phys. Rev. Lett. 113 (16) (2014) 168301, http://dx.doi.org/10.1103/PhysRevLett.113.168301.
[59] E. Drioli, A. Brunetti, G. Di Profio, G. Barbieri, Process intensification strategies and membrane engineering, Green Chem. 14 (6) (2012) 1561–1572.
[60] A. Lee, J.W. Elam, S.B. Darling, Membrane materials for water purification: design, development, and application, Environ. Sci.: Water Res. Technol. 2 (1) (2016) 17–42.