UNIVERSITY OF CALIFORNIA
Santa Barbara

Level Set Strategy for SCFT

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

by

Gaddiel Ouaknin

Committee in Charge:

Professor Frederic Gibou, Chair

Professor Glenn H. Fredrickson

Professor Linda Petzold

Professor Ted Bennett

June 2016

The Dissertation of
Gaddiel Ouaknin is approved:

Professor Glenn H. Fredrickson

Professor Linda Petzold

Professor Ted Bennett

Professor Frederic Gibou, Committee Chairperson

June 2016

Level Set Strategy for SCFT

by

Gaddiel Ouaknin

This thesis is dedicated to my wife Nataliia Shaina and my daughter Yael Tanya.

Acknowledgements

Thank you to Professor Frederic Gibou for his supervision on the level set methods part. Thank you to Professor Glenn Harold Fredrickson for his supervision on the Self Consistent Field Theory (SCFT) part. Thank you to Professor Linda Petzold and Professor Ted Bennett, for being part of my committee, following my progress and reading my thesis.

Thank you to Professor Gibou for his patience, perseverance, encouragment and vital supervision in the research. Thank you to Professor Gibou for his support and enthusiasm during the PhD. Thank you for introducing and teaching me the field of level set methods, shape optimization, sharp boundary conditions, quad/oct tree grids and giving me depth and breadth in the level set theory and implementation.

Thank you to Professor Fredrickson for introducing and teaching me the field of SCFT, for his invaluable advice on polymer theory during the research. Thank you to Professor Fredrickson for his optimism during the PhD. Thank you for sharing and reveal me the rich SCFT theory.

Thank you to all Professor Gibou and Professor Fredrickson current and past group members. Thank you to Dr Nabil Laachi and Dr Kris Delaney for their help on SCFT. Thank you to Dr Laachi for his crucial help on the functional level set derivatives and truthful friendship. Thank you to Dr Delaney for his indispensable help on understanding the underlying physics and fine details in SCFT. Thank you to Dr Mirzadeh and Dr Guittet for having developed `parCASL`. Thank you to Dr Mirzadeh for his critical help installing my code on the Stampede super computer and fruitfull discussions on level set methods.

Thank you to my parents Marc-Alain and Dory Fanny who made this possible. Thank you to my brother Shamgar Maor and sisters Sivane Mikhal and Nin-Gal Neta Shlomtzion for their love.

Thank you to my wife Nataliia Shaina for giving me support, strength, love, energy and that without her the PhD would not have been possible. Thank you to my daughter Yael Tanya who gives me infinite joy.

Gaddiel Ouaknin, Santa Barbara, California, May 2016

Curriculum Vitæ

Gaddiel Ouaknin

Education

University of California Santa Barbara
Ph.D., Mechanical Engineering, 2016.
Field: Computational Sciences Engineering.
Technion-Israel Institute of Technology
MSc, Mechanical Engineering, 2009, Summa Cum Laude.
Technion-Israel Institute of Technology
BSc in Materials Engineering and B.A in Physics, 2006, Cum Laude.

Publications

1. G. Y. Ouaknin and P. Bar-Yoseph, Stochastic Collective Movement of Cells and Fingering Morphology: No Maverick Cells, Biophysical Journal, Volume 97, Issue 7, 2009.
2. G. Y. Ouaknin, N. Laachi, K. Delaney, G. H. Fredrickson, and F. Gibou. Shape optimization for DSA, SPIE Proceedings, Advanced Lithography, 2016.
3. G. Y. Ouaknin, N. Laachi, K. Delaney, G. H. Fredrickson, and F. Gibou. Level set strategy for inverse DSA-lithography, submitted, 2016.
4. G. Y. Ouaknin, N. Laachi, K. Delaney, G. H. Fredrickson, and F. Gibou. Self consistent field theory simulations of polymers on arbitrary domains, submitted, 2016.
5. G. Y. Ouaknin, N. Laachi, D. Bochkov, K. Delaney, G. H. Fredrickson, and F. Gibou. Functional Level-Set Derivatives for Free Boundary Problems with Polymer Self Consistent Field Theory, submitted, 2016.

Teaching

University of California Santa Barbara
Teaching Assistant in Math of Engineering, Numerical Methods, Heat Transfer, Fluid Mechanics and Dynamics.

Technion-Israel Institute of Technology
Teaching Assistant in Finite Element Method, Numerical Methods and Heat Transfer.

Abstract

Level Set Strategy for SCFT

Gaddiel Ouaknin

This thesis investigates the design of sharp interface level set methods in the context of self-consistent field theory (SCFT) in polymer physics. SCFT computes the structure and energy of inhomogeneous self-assembling polymers at thermodynamic equilibrium. Level set methods are based on an implicit representation of free boundaries, which enable motions with arbitrary change in topology. In addition, recent advances on how to impose Robin boundary conditions enables the study of free boundary problems of interest in the community interested in self-assembly.

We first present a computational framework, encoded on a forest of quad/oct-trees in a parallel environment. We then present results of imposing sharp Neumann boundary conditions as was first proposed by de Gennes, which enables SCFT computations of meaningful quantities *at* the boundary of irregular geometries. We then introduce the concept of functional level-set derivative in the context of SCFT and rigorously derive expressions for the change of energy of a diblock copolymer with respect to an enclosing shape. The level-set derivative is then used to embed SCFT into a variable shape simulator, where the internal structure and the enclosing shape are coupled together and evolve in tandem in order to reduce the energy of the diblock copolymer. Finally an algorithm for solving the inverse problem for directed self-assembly is presented.

Contents

List of Figures

Chapter 1

Introduction

Block copolymer self-assembly find a myriad of applications, from the micro electronics industry for patterning in lithography [84, 83] to drug delivery systems [46, 70] to food sciences [63]. In the case of an infinite domain, the emerging structures depend only on the interactions between the blocks. Self-Consistent Field Theory (SCFT) has provided a predictive computational framework that enables the study of such systems [7]. However, this veritable formalism, now over 40 years old, has never succumbed to numerical solutions in other than prescribed domain shapes. In the case of free surface polymers, it is the coupling between the free surface shape and the polymers that determines the thermodynamic equilibrium [32]. We introduce a multiscale simulation framework for free boundary block copolymers, which enables investigations of the interplay between the free surface's geometry and the self-assembled morphology. This represents a new frontier for SCFT and paves the way to enabling the solution of a wide

range of important problems to be addressed, from spreading of block copolymer droplets and films, to predictions of shape evolution during copolymer nanoparticle synthesis, to understanding the mysterious cubosome phases formed by edible surfactants in food science. Furthermore, the methodology we introduce can also be used as an inverse solver in the context of directed self-assembly (DSA), which could be extremely influential as semiconductor chip and storage manufacturers have increasingly turned to DSA as they seek to pattern microelectronic devices at ever smaller scales while limited by expensive photolithography tools that have not kept up with the pace of miniaturization.

The distinct agents of polymeric materials are long chains with organic molecules attached to the backbone of the chain. Diblock copolymers are polymer chains where one fraction of the chain is made of molecule A and the remaining from molecule B. When placed together, they will self assemble into ordered structures due to the competition between entropic and chemical forces. The entropy contributes to the mixing of all the chains and form a disordered phase, whereas the chemical force contributes to the segregation of A and B into two different spatial regions.

The theory of inhomogeneous polymers has been developed through the 60's to the beginning of the 80's. The mathematical model considers an Hamiltonian that takes into account the entropy energy and the interaction energy between the components A and B using the pair (f_A, χ_{AB}). Using an Hubbard-Stratonovitch transform the Hamiltonian is represented through an integral over fields instead of densities. The mean field theory

referred, or self consistent field theory (SCFT), has been initiated by Helfand in 1975 [35, 36] and has become a mature theory.

The use of SCFT as a computational tool has started in the 90's and has had a profound impact. In 1994 [58] the self consistent theory was used as a computational tool to build a phase diagram. This methodology was able to map the parameters (f_A, χ_{AB}) into a phase diagram with BCC, cylindrical and lamellar phases among other. In addition it did find new phases that had not been predicted by simpler theories. For instance the gyroid phase was found to be a stable phase between the lamellar and the cylindrical phases. In addition, it helped to predict how to stabilize particular phases using a blend of diblock-homopolymer [61]. It is used to guide how to play with parameters to obtain long range order and to affect the order itself. Since then SCFT has been used as a computational tool and has been able to predict phase diagrams for many different polymer architectures and component combinations [7] and shapes with success [52, 51, 50, 49, 48, 47]. More Recently, SCFT is used to investigate order in thin films [9, 8, 39]. SCFT is therefore a predictive computational tool that has impacted many fields of science and engineering.

Considerable research has advanced our understanding of which phases develop in the bulk of diblock copolymer melts. Understanding and controlling phase formations in free boundary domains, however, is still in its infancy. Free boundary domains present additional scientific and technological challenges due to the delicate balance between size effects, substrate interactions, surface tension forces, the concentration of the minority

component, incompressibility and other constraints, in addition to the copolymers' natural propensity to segregate into phases. When the polymeric material is confined to a specific shape, its structure and topology are a function of the shape and size of the confinement domain. When polymeric materials are immersed as a non-solvent into a fluid, their shape is free to evolve and in this case the equilibrium shape and phase are both unknown. This is also the case of a polymeric material wetting a substrate, where the bottom substrate is fixed but the upper surface is free. These are extremely challenging problems because of the two-way coupling between the shape of the confining domain or free surface and the possible phase morphologies. This constitutes an infinite dimensional parameter space that is particularly difficult to study. One of the crucial goals of a predictive computational tool in this case is to enable the inverse problem design, i.e. given a targeted design, what is the shape of the confined domain or free surface (in addition to the usual parameters such as the pair (f_A, χ_{AB}) in the case of block copolymer) that will produce that design. However, the understanding of the behavior of diblock copolymers in free boundary domains is crucial for many important applications. In addition to constituting a simple model system that can inform how more general systems self-assemble, block copolymer with a free surface possess crucial commercial and scientific applications. Gyroid phases can be used for drug de livery systems, e.g. the cubosome, which is a cavernous gyroid structure [16], can be tailored for solubilizing and delivering different therapeutic and diagnostic agents and can be used as a controlled-release, intravenous drug delivery engine. Cylindrical phases can be

used to create connections between layers of integrated circuits in the computer industry, with the goal of advancing Moore's Law at sub-20 nanometer resolution and decrease the cost of manufacturing semiconductor components. Yet, without a predictive simulation tools that can consider the interplay between the kinetics and thermodynamics effects of free boundary copolymers, a clear understanding of how to control these phenomena will remain illusive.

In none of previous of computational studies [53, 54, 34, 33, 14, 90, 56] was the shape considered to be arbitrary and no optimization procedures can consider the geometry of the confined domain or free surface as a parameter. In recent studies, researchers have worked towards that goal and have introduced methodology to consider either shape optimization in lithography or for free surfaces. In [53, 54], Latypov proposes an inverse design algorithm where the shape is unknown and the input is a target density. However, this approach uses a linearization of the mean field equations and the shape is expressed through a parametric form and therefore constrained to that specific shape parametrization. In [90], Stasiak *et al.* describe the free surface of a AB diblock copolymer by adding a fictitious polymer C. This approach enables the motion of the free surface of the AB diblock polymer. However, some careful choices of the segregation parameters χ_{AB}, χ_{AC} and χ_{BC} must be provided in order to avoid the case where the C component penetrates AB. In addition, shape optimization can not be performed with this strategy. In [56], the authors use a slightly different strategy than that of Stasiak *et al.* [90] and add terms to the Hamiltonian to represent the vapor phase of the

polymer melt. In [34, 33, 14], a set of proposed strategies for inverse design lithography is presented where they do not solve for the optimal confinement but instead solve for the optimal location of posts inside the domain to obtain a desired pattern. While it is a successful strategy it does not enable the change in the confinement's shape.

We introduce a computational framework that enables the study of free boundary diblock copolymers by combining the level-set formulation [73] for free boundary problems with the self-consistent field theory description of self-assembly of diblock copolymers. The level-set is an implicit non-parametric representation that can describe closed curves of any shapes. We can thus consider optimizations in the parameter space (f_A, χ_{AB}, ϕ) where the geometry is represented by a level set function. In particular, we focus on two important classes of applications. The first concerns the study of free surface copolymers. The second type of problem is in the context of Directed Self- Assembly (DSA), a patterning technique for next generation lithography [52, 51, 50, 49, 48, 47, 53, 54, 34, 33, 14]. In this context, the challenge is to find template shapes that can direct the self-assembly of the diblock copolymer into various cylindrical structures with a targeted topology. In both cases, the computational approach is capable of combining the description of self-assembly from a statistical mechanics approach, while considering a free boundary. The symmetric diblock with a free surface is an excellent example. We consider a simple drop immersed in a fluid where the surface of the polymer behaves as the interface between the polymer and the fluid. We simulate this process with our computational engine where we apply homogeneous Neumann boundary conditions on the interface

[21]. Figure 1.1 gives an example of free surface for symmetric diblock copolymers (in this case, we added to the free energy Hamiltonian a surface tension term). The interplay between the phase and the shape in this case is subtle. In the bulk, a symmetric diblock will have a lamellar morphology [24]. The lamellar morphology means that there is a symmetry in the axes perpendicular to the lamellar direction. The density of the polymer will vary only in the direction orthogonal to the lamellae. On the other hand, when immersed in a fluid as a droplet, surface tension will also play a role. If the shape is driven by surface tension only, the droplet will generate a perfect circle. But in our case, due to the lamellar phase which emerges inside the drop, the polymer will prefer to push in the $\boldsymbol{x}$ direction to minimize stretching by forming domains of uniform thickness and create an oval shape. The interesting points here are (i) that the shape plays a role in the thermodynamic equilibrium phase of the self assembled material that is no less important than the entropy and the chemical forces (ii) the phase plays a role on the shape at equilibrium. Our computational approach demonstrates that it can properly take into account the two-way coupling between these competing effects. In this case, the phase diagram will be a function of the triplet (f_A, χ_{AB}, V), where V is the volume of the polymer, instead of the simpler pair (f_A, χ_{AB}).

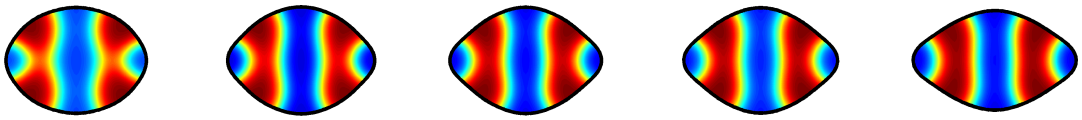


Figure 1.1: Drop Evolution $(f_A, \chi_{AB}) = (0.5, 16)$

Here we conserve the volume of the physical domain and evolve the shape such as to minimize the free energy. We start from a circular shape and end up with a shape elongated in the lamellar direction. The shape velocity has a higher net contribution in the direction of the lamellar structure which forces the material to stretch itself in this direction until the surface tension forces balance out. It does mean that the internal anisotropic lamellar structure is transferred to the shape that follows the symmetry of the internal structure. The cubosome is a fascinating three dimensional structure. The minority region consists of a continuous network of a gyroid-like shape. It has various chemical properties and can be used for many applications. One important of them is drug delivery in the human body.

For the drug delivery to be successful, many things have to happen concurrently. The most important is that it must carry it. The cubosome can do that due to its particular morphology. Its particular cavernous structure allows it to trap the drug inside the network. Then if used with a set of appropriate chemical properties such as hydrophilic, hydrophobic philippic flavors, it delivers the drug in a controlled and sustained way. To predict the morphology of the cubosome at thermodynamic equilibrium, one would

either perform experiments, computations or both. A computation of such a problem requires a variable shape SCFT solver to allow the investigation of the interplay between the shape and the structure. A better understanding would allow to design cubosome with the desired surface ratio and curvature to improve the drug delivery process both for the transport and the storage of the drug.

Another important application of cubosome is in the food industry as one wants to add cubosome to nutritional products to improve their flavor and or conservation qualities. In this case the morphology of the cubosome may affect final flavor of the food product and the interplay shape/structure is of importance.

In figure 1.2, we show an example of a 3D cubosome with a free surface. The shape is evolved to minimize the free energy and we can see the continuous network of A inside the cubosome. Simulations in this case can answer many questions: (i) do we obtain a network structure, (ii) do we obtain a continuous or bicontinuous structure, (iii) what is the surface/volume ratio obtained, (iv) how much the structure obtained is cavernous and (v) how the parameters such as f_A, χ_{AB}, the surface tension and the volume of the cubosome affect all of these properties. From the result presented in figure 1.2 questions (i)-(iii) can be answered for the set of parameters this simulation was run with: we obtained a continuous network structure with a high surface/volume ratio.

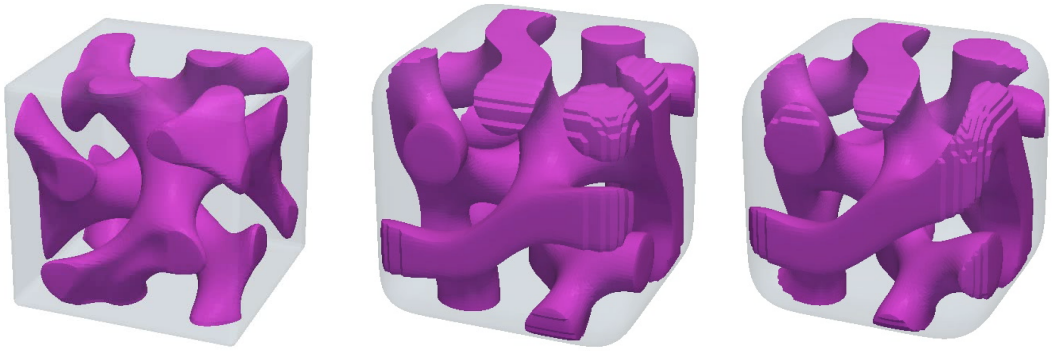


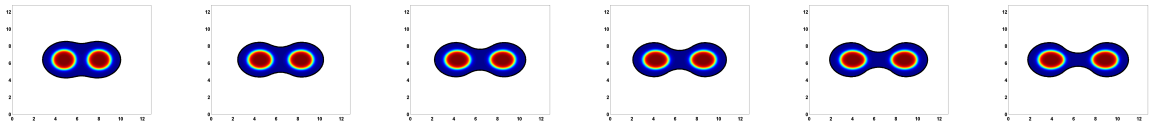
Figure 1.2: Cubosome Evolution $(f_A, \chi_{AB}) = (0.36, 20)$

Cubosomes have a network structure of the cubic phase and consist of cavernous structures separating two channels and a large interfacial area [29]. They are at the same time hydrophobic, hydrophilic and amphiphilic [29]. Due to their unique structure cubosome materials can be used for cubosomal drug delivery such as melanoma therapy [29]. They can be tailored for solubilizing and delivering different therapeutic and diagnostic agents and can be used as controlled release or intravenous drug delivery engine [28, 70]. In a recent study it is shown how self-assembled structures are leveraged for ophthalmic drug delivery [28] and in [70] they exploit the full range of properties of the cubosome. They use the cubosome to sustain a drug and to deliver it in a controlled way. The high ratio surface property is used to carry the drug and the phobic and philic properties to provide a long duration sustained release of an hydrophobic drug. In addition to drug delivery cubosome structures have been found in food materials [63, 89]. A good understanding of the self-assembly process with a free surface will improve our understanding of food materials and then their production. In the case of

the micro electronic industry, one tries to direct the self assembly (DSA) of polymers and to leverage the self-assembly of polymeric materials to create fine features. DSA helps selecting position and orientation of features and to remove defects. A mask is fabricated and a polymeric material is filled inside. In this application, a self-assembled cylindrical phase is targeted. That is, the minority region of component A creates a two dimensional topology of cylinders that is surrounded by the B component. By applying dry or wet etching processes that are selective to A, the A component is removed and small-scale features, e.g. channels, are created. These channels are then filled with conductors/semiconductors, creating components that can be used in circuit design. The challenge in DSA is to obtain the desired center to center distance of the components and complex topologies of the design. Another use of polymer in the electronic industry is to use body centered cubic (BCC) spheres on a thin film with long range order to fabricate high density hard drives [83]. Last but not least is the inverse DSA problem. In this case one wants to enforce a particular topology and morphology. While the pair (f_A, χ_{AB}) enforces the phase, long-range order is hard to obtain as natural defects can emerge. Also in confined domains in some cases the shape can change the phase. Even if we successfully obtain a phase cleaned of defects, we still need to obtain a certain desired topology. So it is an inverse problem where we want to find (f_A, χ_{AB}) as well as the shape that will produce a desired output with specific characteristics. We evolve the shape in such a way that when we the final shape obtained to solve the direct problem

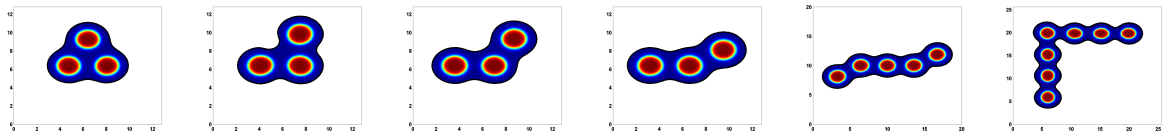
we obtain our density target. That is we solve an inverse problem with respect to the shape.

In figures 1.3 and 1.4 we show examples with different topologies and distance to distance center. The shape is the output of a shape optimization algorithm. These kind of topologies are of high interest in the lithography industry where two or more cylinders need to be put adjacent to each other with a specific distance between them. We can see that the method proposed here is robust for different topologies. It can find the optimal shape for different designs.



(a) $d_{cc} = 3.2R_g$ (b) $d_{cc} = 3.8R_g$ (c) $d_{cc} = 4.2R_g$ (d) $d_{cc} = 4.4R_g$ (e) $d_{cc} = 4.8R_g$ (f) $d_{cc} = 5.0R_g$

Figure 1.3: Different center-to-center distances with $(f_A, \chi_{AB}) = (0.3, 36)$



(a) $\pi/6$ (b) $\pi/2$ (c) $\pi/3$ (d) $\pi/6$ (e) two junctions (f) l-shape

Figure 1.4: Different configurations with $(f_A, \chi_{AB}) = (0.3, 36)$

In the next chapter a level set based computational framework for Neumann boundary conditions is presented [77]. In the subsequent chapter the functional level set deriva-

tive is derived rigorously, a level set based shape optimization strategy is proposed and illustrative examples are presented [75]. Finally in the last chapter the inverse geometric problem is presented along examples with different topologies [76, 78]. At the end of this thesis can be found an appendix with the derivation of the SCFT equations required in this document.

Chapter 2

SCFT on arbitrary domains with Neumann boundary conditions

2.1 Introduction

We introduce a framework for simulating the mesoscale self-assembly of block copolymers in arbitrary confined geometries subject to Neumann boundary conditions. We employ a hybrid finite difference/volume approach to discretize the mean-field equations on an irregular domain represented implicitly by a level-set function. The numerical treatment of the Neumann boundary conditions is *sharp*, i.e it avoids an artificial smearing in the irregular domain boundary. This strategy enables the study of self-assembly in confined domains and enables the computation of physically meaningful quantities at the domain interface. In addition, we employ adaptive grids encoded with Quad-/Oc-trees

in parallel to automatically refine the grid where the statistical fields vary rapidly as well as at the boundary of the confined domain. This approach results in a significant reduction in the number of degrees of freedom and makes the simulations in arbitrary domains using effective boundary conditions computationally efficient in terms of both speed and memory requirement. Finally, in the case of regular periodic domains, where pseudo-spectral approaches are superior to finite differences in terms of CPU time and accuracy, we use the adaptive strategy to store chain propagators, reducing the memory footprint without loss of accuracy in computed physical observables.

Block copolymers are macromolecules comprised of two or more chemically distinct sequences (or "blocks") of repeated and chemically bound monomers. Ensembles of these molecules exhibit fascinating characteristics: in the melt state, they self-assemble into periodic ordered structures in a wide range of morphologies, thus offering the possibility of designing materials with a broad property spectrum [7]. As such, polymeric materials offer many advantages such as scalability, tunability, ease of functionalization, mechanical flexibility, compatibility with various processing methods and a relatively low cost. We focus here on the important class of *diblock copolymers*, in which each polymer chain is made of two components (see figure 2.1): a minority block component A made of monomers a and a majority block component B made of monomers b. Di-

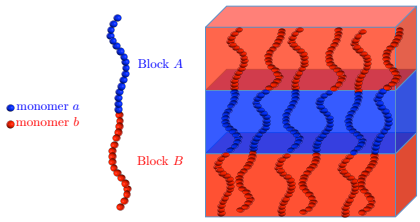


Figure 2.1: Schematic of a diblock copolymer in the lamellar phase.

block copolymers are of particular interest because they produce regular structures with a length scale (5 to 100 nm) that is difficult to access with other patterning techniques. As such, they are used in a myriad of applications in the electronics, energy and health industries [16, 69, 83, 19, 62].

At the mean-field level, the thermodynamic behavior of diblock copolymer melts is dictated by just two dimensionless parameters: the fraction, f_A, of block A along with $\chi_{AB}N$. χ_{AB} is the Flory-Huggins interaction parameter which quantifies the chemical repulsion between the blocks A and B and N is the number of segments (monomers) in the copolymer chain. One of the most important aspects in the study of diblock copolymers is to predict which ordered structures of polymeric materials can form at equilibrium as a function of f_A and $\chi_{AB}N$ [24, 7, 35]. The self consistent field theory (SCFT) [24, 35, 36] is a successful mean-field theory and a mature computational tool for describing inhomogeneous phases in the parameter space $(f_A, \chi_{AB}N)$. It has enabled researchers to predict self-assembly in a wide range of situations [24, 60, 61, 59]. Within this framework, the physical observables such as the densities of the A and B components are expressed in terms of chain propagators, which represent the statistical weight of a polymer chain at a given location and contour length and satisfy a Fokker-Planck equation. In the case of unbounded block copolymer melts, it is justified and convenient to impose periodic boundary conditions and to employ fast pseudo-spectral solvers [81].

In the case of confined domains, other boundary conditions must be imposed. At the atomistic level, the density displays an oscillatory profile in a "boundary layer" with

size below $\sim$ 1nm but at the mesoscopic scale, we are not interested in the details of the solution in this boundary layer. In principle, a Dirichlet boundary condition should be imposed; however the use of a Dirichlet boundary condition is incompatible with the incompressibility assumption [24]. De Gennes thus proposed the use of an effective (Neumann) boundary condition for molten polymers at air or solid interfaces [21, 24].

Currently, the technique that is used to predict self-assembly of block copolymers in confined domains, while still using a fast pseudo-spectral approach, is to impose a wall density to emulate the presence of a physical wall [8, 39, 15, 43, 49, 44]. Effectively, the wall defines a mask that varies smoothly from 0 to 1 over a length scale defined by a parameter describing the width of the "soft" wall. However, while this technique produces accurate solutions inside the irregular domain, it produces fast varying solutions near the boundary that are arbitrarily specified by the mask. In turn, estimating numerically the solution and its derivatives near the boundary, which are needed in the context of free boundary optimization problems [76, 78], produces far less robust results.

Here we introduce a hybrid finite-difference/finite-volume discretizations of the SCFT equations on irregular domains where effective boundary conditions are imposed in a numerically *sharp* fashion. In addition, we present an adaptive mesh refinement technique that focuses the computational resources in regions where the densities vary rapidly. The adaptivity is based on a forest (or collection) of Quad-/Oc-tree data structures in a parallel environment [11, 68]. The paper is organized as follows: in section 3.2, we present the mathematical model describing the theory of copolymer thermodynamics

and the SCFT algorithm. Section 2.3 details the numerical approach, while section 2.4 presents the extension to (potentially massively) parallel environments. Section 2.5 presents a set of numerical results for confined domains and section 2.5.5 presents the numerical results demonstrating that the use of adaptive grids can be beneficial in term of memory requirement, even in the case where one uses a pseudo-spectral approach.

2.2 Self-Consistent Field Theory

In this section we describe the equations of the self-consistent field theory (SCFT) [35, 36] used in the current computational study. A detailed development of these models can be found in [24] and the references therein, and a more physically-oriented derivation can be found in [57].

2.2.1 SCFT Incompressible Model for AB Diblock Copolymer

The physical system under consideration is a melt composed of AB diblock copolymers. The success of the field-based theory of polymer thermodynamics rests on the use of a Hubbard-Stratonovich transform that converts the partition function from a particles representation to a field representation, hence providing a tractable computational approach. In the mean field approximation, where fluctuations are neglected, the free energy of the melt is given by the Hamiltonian, H, evaluated at the saddle point field configuration, i.e. $\dfrac{\delta H}{\delta w(\boldsymbol{r})} = 0$. We therefore search for the statistical fields w_- and w_+

that give the extremum of the Hamiltonian[1]:

$$H[w_+, w_-] = \frac{1}{V}\int (-w_+(\boldsymbol{r}) + \frac{w_-^2(\boldsymbol{r})}{\chi_{AB}N})\, d\boldsymbol{r} - \ln Q[w_+, w_-], \tag{2.1}$$

where the normalized partition function Q can be formulated as:

$$Q[w_+, w_-] = \frac{1}{V}\int q(s=1, \boldsymbol{r}; [w_+, w_-])\, d\boldsymbol{r}.$$

In these equations, $w_+(\boldsymbol{r})$ acts as a pressure potential that enforces the incompressibility constraint while the exchange potential, $w_-(\boldsymbol{r})$, accounts for the interaction between A and B segments. The functionals $q(s, \boldsymbol{r})$ and $q^\dagger(s, \boldsymbol{r})$ are solutions of the Fokker-Planck equations with initial conditions $q(s = 0, \boldsymbol{r}) = q^\dagger(s = 0, \boldsymbol{r}) = 1$. Here, $s \in [0, 1]$ is a scaled contour variable that describes the location along the backbone of the polymer chain. The chain propagators q and $q^\dagger$, represent the probability of a statistical segment of the chain at location $\boldsymbol{r}$ and contour length s, and satisfy the forward and backward Fokker-Planck equations, respectively:

$$\begin{cases} \partial_s q(s, \boldsymbol{r}) = \bigtriangledown^2 q(s, \boldsymbol{r}) - q(s, \boldsymbol{r})\, w(\boldsymbol{r}, s) & \text{forward.} \\ \\ \partial_s q^\dagger(s, \boldsymbol{r}) = \bigtriangledown^2 q^\dagger(s, \boldsymbol{r}) - q^\dagger(s, \boldsymbol{r})\, w^\dagger(\boldsymbol{r}, s) & \text{backward.} \end{cases} \tag{2.2}$$

These equations are supplemented with periodic boundary conditions in the case of periodic phases and homogeneous Neumann boundary conditions in the case of confined

[1]The Hamiltonian H that we compute is actually H/n where n is the number of polymeric chains.

domains [24]. The forward potential $w(\boldsymbol{r}, s)$ is given by:

$$w(\boldsymbol{r}, s) = \begin{cases} w_A(\boldsymbol{r}) = w_+(\boldsymbol{r}) - w_-(\boldsymbol{r}) & 0 < s < f_A, \\ \\ w_B(\boldsymbol{r}) = w_+(\boldsymbol{r}) + w_-(\boldsymbol{r}) & f_A < s < 1, \end{cases}$$

and the backward potential $w^\dagger(\boldsymbol{r}, s) = w(\boldsymbol{r}, 1-s)$. Finally, the normalized densities of the components A and B (local volume fraction) are obtained through the functional derivatives of Q with respect to $w_A(\boldsymbol{r})$ and $w_B(\boldsymbol{r})$ and can be written as integrals of q and $q^\dagger$ over the contour variable:

$$\begin{cases} \rho_A(\boldsymbol{r}; [w_+, w_-]) = \dfrac{1}{Q[w_+, w_-]} \displaystyle\int_0^{f_A} q^\dagger(1-s, \boldsymbol{r}, [w_+, w_-]) q(s, \boldsymbol{r}, [w_+, w_-])\, ds. \\ \\ \rho_B(\boldsymbol{r}; [w_+, w_-]) = \dfrac{1}{Q[w_+, w_-]} \displaystyle\int_{f_A}^1 q^\dagger(1-s, \boldsymbol{r}, [w_+, w_-]) q(s, \boldsymbol{r}, [w_+, w_-])\, ds. \end{cases} \tag{2.3}$$

The SCFT algorithm (see algorithm 1) searches iteratively for the saddle point of $H[w_+, w_-]$ in terms of the statistical fields. The driving forces of the optimization procedure are obtained through the functional derivatives of H with respect to $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$ and are given by:

$$\begin{cases} f_+(\boldsymbol{r}) = \rho_A(\boldsymbol{r}) + \rho_B(\boldsymbol{r}) - 1 \\ \\ f_-(\boldsymbol{r}) = \dfrac{2w_-(\boldsymbol{r})}{\chi_{AB} N} + \rho_B(\boldsymbol{r}) - \rho_A(\boldsymbol{r}) \end{cases}, \tag{2.4}$$

where w_+ acts as a Lagrange multiplier to enforce incompressibility. For the optimization algorithm we use the simple Euler step:

$$\begin{cases} w_+^{t+1}(\boldsymbol{r}) = w_+^t(\boldsymbol{r}) + \lambda\, f_+(\boldsymbol{r}) \\ \\ w_-^{t+1}(\boldsymbol{r}) = w_-^t(\boldsymbol{r}) - \lambda\, f_-(\boldsymbol{r}) \end{cases}, \tag{2.5}$$

where λ is the fictitious time step of the optimization.

Algorithm 1 SCFT Optimization

Set F_- and F_+ defined in (2.6) arbitrarily large.

while $\max(F_+, F-) > \epsilon$ **do**

1. Solve the two diffusion equations given in (B.4) from $s = 0$ to $s = 1$ with a step Δs.
2. Compute the local volume fractions ρ_A, ρ_B according to (2.3).
3. Compute the driving forces using (2.4).
4. Advance the potentials w using (2.5).
5. Compute the free energy H and compute F_- and F_+ defined in (2.6).

The convergence is established when the L^2-norms:

$$F_- = \left(\frac{1}{V}\int_V f_-^2(\boldsymbol{r})dV\right)^{\frac{1}{2}} \quad \text{and} \quad F_+ = \left(\frac{1}{V}\int_V f_+^2(\boldsymbol{r})dV\right)^{\frac{1}{2}} \tag{2.6}$$

are less than a given tolerance ϵ. The while loop in algorithm 1 is referred to as "the mean field steps". We note that the algorithm does not ensure that one finds the global minimum, but a local minimum only. As any global optimization problem, the outcome depends on the initial seed and simulations with different seeds are usually performed to ensure that a stable phase is obtained [24].

2.2.2 SCFT Weakly Compressible Model with Mask for AB Diblock Copolymer

Imposing Neumann boundary conditions in arbitrary geometries is difficult or impossible in the case of the pseudo-spectral approach. As a consequence, researchers have introduced a technique that emulates the presence of a wall. This is accomplished by considering a normalized local volume fraction ρ_w that augments the pressure field and thus repels the polymeric material inside the confined domain, hence emulating a physical wall. This approach is the so-called "mask" model [8, 39, 15, 43, 49, 44]. Specifically, the Hamiltonian for a confined compressible polymer is given by:

$$H[w_+, w_-, \rho_w] = \frac{1}{V}\int \frac{-0.5(\zeta N)^{-1}}{0.5\chi_{AB}N(\zeta N)^{-1}+1} w_+^2(\boldsymbol{r}) + \frac{\rho_w(\boldsymbol{r})-1}{0.5\chi_{AB}N(\zeta N)^{-1}+1} w_+(\boldsymbol{r})\, d\boldsymbol{r} + \int \frac{w_-^2(\boldsymbol{r})}{\chi_{AB}N}\, d\boldsymbol{r} - \bar{\rho}_d \ln Q[w_+, w_-], \tag{2.7}$$

In this case the total normalized density is $\rho_A + \rho_B + \rho_w$, where ρ_w is the wall normalized density.

Effectively, the wall density defines a mask that takes a value of 1 deep outside the confined domain, 0 deep inside, 0.5 at the wall's boundary and varies smoothly from inside to outside the domain on a length scale defined by δ_w, a parameter of the model describing the width of the "soft" wall. The parameter $(\zeta N)^{-1}$ in equation (2.7) is the compressibility tolerance, which tends to zero as the material becomes incompressible.

In this work, we use for convenience a level-set function $\phi(\boldsymbol{r})$ (see section 2.3) to define the wall density as:

$$\rho_w(\boldsymbol{r}) = \frac{1}{2}\left(\frac{e^{\phi(\boldsymbol{r}/\delta_w)} - e^{-\phi(\boldsymbol{r}/\delta_w)}}{e^{\phi(\boldsymbol{r}/\delta_w)} + e^{-\phi(\boldsymbol{r}/\delta_w)}} + 1\right).$$

The level-set is a non-parametric representation that is convenient in describing arbitrary geometries [73]. It also provides an efficient procedure to compute the signed distance function to the boundary of any domain.

The definitions of the quantities $Q[w_A, w_B]$, $w_A(\boldsymbol{r})$, $w_B(\boldsymbol{r})$, $w(\boldsymbol{r})$ and $w^\dagger(\boldsymbol{r})$ as well as the equations for $q(s, \boldsymbol{r})$ and $q^\dagger(s, \boldsymbol{r})$ are the same as in section 2.2.1. The local volume fractions ρ_A and ρ_B are now:

$$\begin{cases} \rho_A(\boldsymbol{r}; [w_A, w_B]) = & \dfrac{\bar{\rho}_d}{Q[w_A, w_B]} \displaystyle\int_0^{f_A} q^\dagger(1-s, \boldsymbol{r}, [w_A, w_B]) q(s, \boldsymbol{r}, [w_A, w_B])\, ds \\ \\ \rho_B(\boldsymbol{r}; [w_A, w_B]) = & \dfrac{\bar{\rho}_d}{Q[w_A, w_B]} \displaystyle\int_{f_A}^{1} q^\dagger(1-s, \boldsymbol{r}, [w_A, w_B]) q(s, \boldsymbol{r}, [w_A, w_B])\, ds \end{cases}, \quad (2.8)$$

where $\bar{\rho}_d$ is the effective domain fraction defined as:

$$\bar{\rho}_d = \frac{1}{V}\int 1 - \rho_w(\boldsymbol{r})\, d\boldsymbol{r}. \quad (2.9)$$

Finally, the pressure and the chemical forces of the optimization procdedure are given by:

$$\begin{aligned} f_+(\boldsymbol{r}) &= \rho_A(\boldsymbol{r}) + \rho_B(\boldsymbol{r}) + \frac{-(\zeta N)^{-1}}{0.5\chi_{AB}N(\zeta N)^{-1} + 1} w_+(\boldsymbol{r}) + \frac{\rho_w(\boldsymbol{r}) - 1}{0.5\chi_{AB}N(\zeta N)^{-1} + 1}, \\ f_-(\boldsymbol{r}) &= \frac{2w_-(\boldsymbol{r})}{\chi_{AB}N} + \rho_B(\boldsymbol{r}) - \rho_A(\boldsymbol{r}). \end{aligned}$$

The SCFT optimization with the mask is solved with standard periodic boundary conditions using the fast Fourier transform [18] and its optimized `FFTW` implementation [25, 26].

2.3 Sharp Numerical Approach

We present a numerical approach that avoids the need to consider a mask, and therefore approximate the effective Neumann boundary condition suggested by de Gennes [21] in a sharp fashion. In addition, we consider a computational framework on adaptive grids in order to provide efficient computations. Finally, we consider Cartesian adaptive grids on which the geometry and the effective boundary condition are implicitly captured instead of explicitly defined; this strategy avoids the difficulties associated with meshing procedure of body-fitted approaches.

Within the context of predicting the self-assembled structures at equilibrium in arbitrary domains, adaptive grids are desirable when considering sharp approaches since a `FFT` cannot be used. Adaptive grids reduce the number of degrees of freedom, hence reducing both the memory and CPU requirement. Cartesian grids based on Quadtree data structures in two spatial dimensions and Octree data structures in three spatial dimensions are effective at capturing inhomogeneous spatial scales. Figure 2.2(a) gives an example of an Octree Cartesian grid for which the minimum level is 1 and the maximum level is 3, where 'level' corresponds to the number of local refinements. Computational cells are split into four children in 2D and eight children in 3D depending on proper

refinement criteria. We use the level-set formalism [73] to describe irregular confinement domains and also as a way to automatically refine near interfaces between A and B regions. Within this framework, the boundary, Γ, of a confined domain in $\mathbb{R}^n$, where n is the physical dimension, is described implicitly as the zero level-set of a function ϕ in $\mathbb{R}^{n+1}$, i.e. $\Gamma = \{\boldsymbol{r} \in \mathbb{R}^n : \phi(\boldsymbol{r}) = 0\}$. The interior of the confined domain is represented as $\Omega^- = \{\boldsymbol{r} \in \mathbb{R}^n : \phi(\boldsymbol{r}) < 0\}$ and its exterior as $\Omega^+ = \{\boldsymbol{r} \in \mathbb{R}^n : \phi(\boldsymbol{r}) > 0\}$.

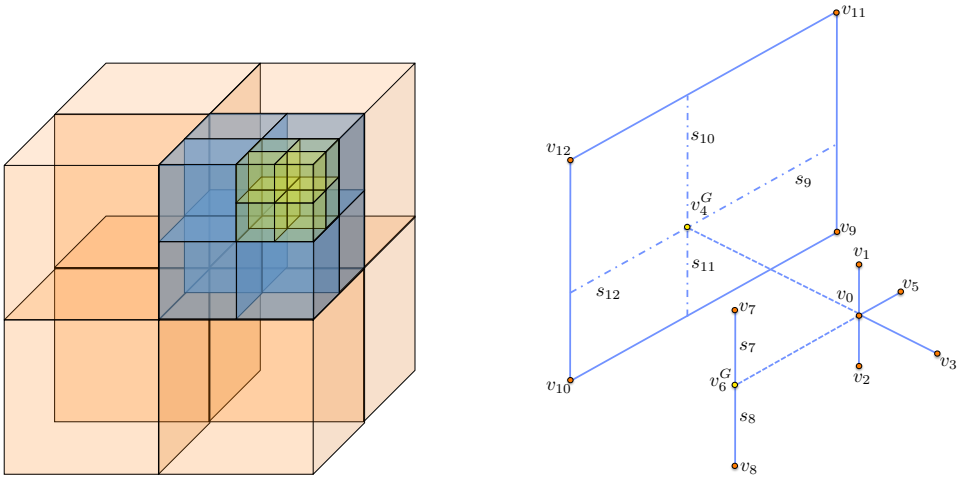
We use the unconditionally stable Crank-Nicholson scheme to solve the equations in (B.4). For example, denoting q^s the solution vector at the contour length s, the discretization of the forward case without the external field is:

$$\frac{q^{s+1} - q^s}{\Delta s} = \frac{1}{2}\left(\widetilde{\nabla}^2 q^{s+1} + \widetilde{\nabla}^2 q^s\right), \tag{2.10}$$

where $\widetilde{\nabla}^2$ refers to the discretization of the Laplace operator approximated as follows: we distinguish the treatment of nodes adjacent to the irregular domain (boundary nodes), which are treated using a finite volume approach as in [79] (see section 2.3.2), to the others (called interior nodes) for which finite differences are used (see section 2.3.1).

2.3.1 Finite Difference Discretization for Interior Nodes

We use the finite difference approximation of standard operators on Octrees introduced in [66]. For the sake of completeness, we will present here the principal results needed for the current study. Typical second-order accurate finite difference approximations require two neighbors in each spatial direction, hence, in the case of adaptive grids, T-junction nodes require the definition of ghost neighborhood nodes. For example, in



(a) A level (1, 3) Octree grid.

(b) General local configuration of a T-junction node v_0.

Figure 2.2: Discretization with an Octree Cartesian Grid.

the case of figure 2.2(b), the degree of freedom q_0, sampled at the node (or vertex) v_0, requires two ghost values q_4 and q_6, located at the virtual vertices v_4 and v_6. In [66, 67], the authors introduced third-order accurate definitions of ghost values:

$$q_6^G = \frac{s_8 q_7 + s_7 q_8}{s_8 + s_7} - \frac{s_8 s_7}{s_2 + s_1}\left(\frac{q_2 - q_0}{s_2} + \frac{q_1 - q_0}{s_1}\right),$$

and

$$\begin{aligned} q_4^G = {} & \frac{s_{11}s_{12}q_{11} + s_{11}s_9 q_{12} + s_{10}s_{12}q_9 + s_{10}s_9 q_{10}}{(s_{10} + s_{11})(s_9 + s_{12})} \\ & - \frac{s_{10}s_{11}}{s_2 + s_1}\left(\frac{q_2 - q_0}{s_2} + \frac{q_1 - q_0}{s_1}\right) \\ & - \frac{s_9 s_{12}}{s_5 + s_6}\left(\frac{q_5 - q_0}{s_5} + \frac{q_6^G - q_0}{s_6}\right), \end{aligned}$$

where the s_i's refer to the distances between the nodes v_0 and v_i. From the definitions of these ghost values, [66] introduced a discretization for the Laplace operator that

produces second-order accurate solutions and second-order accurate gradients:

$$\begin{aligned}\widetilde{\nabla}^2 q_0 \approx & \alpha\left(\frac{q_5 - q_0}{s_5} - \frac{q_0 - q_6}{s_6}\right)\frac{2}{s_5 + s_6} \\ & + \beta\left(\frac{q_3 - q_0}{s_3} - \frac{q_0 - q_4}{s_4}\right)\frac{2}{s_3 + s_4} \\ & + \gamma\left(\frac{q_1 - q_0}{s_1} - \frac{q_0 - q_2}{s_2}\right)\frac{2}{s_1 + s_2},\end{aligned}$$

with

$$\alpha = 1 - \frac{s_{10}s_{11}}{s_4(s_3 + s_4)}, \quad \beta = 1 - \frac{s_9 s_{12}}{s_4(s_3 + s_4)} - \alpha\frac{s_7 s_8}{s_6(s_5 + s_6)} \quad \text{and } \gamma = 1.$$

The approximation of the Laplacian on the grid can thus be written as:

$$\widetilde{\nabla}^2 q = Mq,$$

where M is a matrix that depends only on the Octree data structure (the coefficients of q depend only on the distances s_i). The Crank-Nicholson scheme can thus be written:

$$\left(I - \frac{\Delta s}{2}M\right) q^{s+1} = \left(I + \frac{\Delta s}{2}M\right) q^s.$$

Reintroducing the external field, we solve the Fokker-Planck equation (B.4) with an external potential using the Strang splitting [91]:

$$q^{s+1} = e^{-w\frac{\Delta s}{2}I} M_2^{-1} M_1 e^{-w\frac{\Delta s}{2}I} q^s, \tag{2.11}$$

where $M_1 = \left(I + \frac{\Delta s}{2}M\right)$ and $M_2 = \left(I - \frac{\Delta s}{2}M\right)$. We note that M_2^{-1} has been shown to exist in [66].

2.3.2 Finite Volume Discretization for Boundary Nodes

One of the main challenges in the case of a confined domain is to solve equation (B.4) with homogeneous Neumann boundary conditions at the domain's boundary. In [79, 80], the authors introduced a numerical method for solving a diffusion equation with Robin boundary conditions, which can be trivially adapted to the case of a diffusion equation with effective Neumann boundary conditions and an external potential. For the sake of clarity, we summarize the main steps.

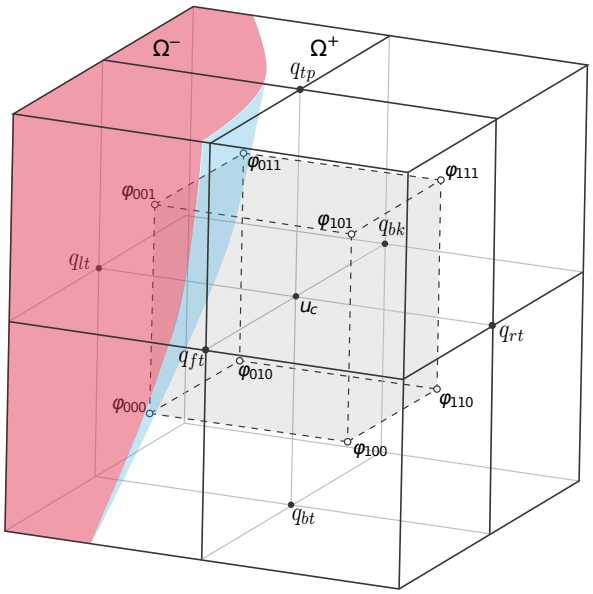


Figure 2.3: Local configuration of a node q_c near a confined domain's boundary.

Referring to figure 5.3 and considering the Crank-Nicholson time stepping, the discretization of the diffusion equation is based on a finite volume approach on the dual

cell, C, with center q_c, giving:

$$\int_{C\cap\Omega^-} \frac{q_c^{s+1}-q_c^s}{\Delta s}\,dV \;=\; \frac{1}{2}\left(\int_{\partial C\cap\Omega^-} \nabla q_c^{s+1}\cdot\boldsymbol{n}\,dA + \int_{\partial C\cap\Omega^-} \nabla q_c^{s}\cdot\boldsymbol{n}\,dA\right),$$

where we have used the homogeneous Neumann interface condition to cancel the boundary term. The surface integrals are then approximated with central differences, giving:

$$\begin{aligned}\int_{\partial C\cap\Omega_-} \bigtriangledown q_c\cdot\boldsymbol{n}dA \;&=\; \frac{q_{rt}-q_c}{h}A_{+x}+\frac{q_{bk}-q_c}{h}A_{+y}+\frac{q_{tp}-q_c}{h}A_{+z}\\ &+\; \frac{q_{lt}-q_c}{h}A_{-x}+\frac{q_{ft}-q_c}{h}A_{-y}+\frac{q_{bt}-q_c}{h}A_{-z},\end{aligned}$$

where the $A_{\pm x}$ refer to the area of the dual cell's faces in the $\pm\,x$-direction; $A_{\pm y}$ and $A_{\pm z}$ are defined similarly. Finally, we approximate the volume integrals with the geometric procedures detailed in [64, 65].

The discretizations of sections 2.3.1 and 2.3.2 produce a linear system of equations for q^{s+1} on which we apply a Strang splitting of the form of equation (5.20). We note that, for efficient implementation, M_1 and M_2 need to be computed only once for a given mesh and can then be used throughout the solution process for q.

2.3.3 Refinement Strategy

Given a level-set function $\phi(\boldsymbol{r})$ which represents a signed distance function from some interface, refining near that interface can be enforced by splitting a cell if:

$$\|\phi(\boldsymbol{r})\| < \text{cell size}\times L, \tag{2.12}$$

where choosing the Lipschitz constant to be $L=1$ gives a grid graded from the interface [67, 42]. The refinement criteria is applied recursively until the maximum preset tree

level is achieved. The level-set technology on Quad/Octree grids [67] is flexible and many variants of this criterion can be employed. For instance, we can in addition choose to refine uniformly inside the region where ϕ is negative by splitting cells if:

$$\phi(\boldsymbol{r}) < \text{cell size} \times L. \tag{2.13}$$

We will use this refinement for generating finer grids near the confined domains' boundary and to generate adaptive grids inside the bulk in order to capture the rapid transition in the solution near regions between the A and B components. Other refinement criteria can be used, e.g. to impose a band of uniform cells near a boundary; we refer the interested reader to [67] for the details. For confined domains, we generate the level-set when we specify the domain's geometry. We set negative values at the nodes inside the confined domain and positive values outside. We then reinitialize the level-set function to transform it into a signed distance function by solving the reinitialization equation [92]:

$$\frac{\partial \phi(\boldsymbol{r},\tau)}{\partial \tau} + \text{sign}\left(\phi(\boldsymbol{r},\tau)\right)\left(\|\nabla\phi(\boldsymbol{r},\tau)\| - 1\right) = 0, \tag{2.14}$$

where the time τ in the equation is a pseudo time.

The refinement strategy adopted inside the bulk exploits the structure of the block copolymer mesophases, which are composed of near homogeneous regions that can easily be clustered (figure 2.4 [left] gives a typical example) and across which the fields undergo large variations (see Figure 2.4 center and right). In particular, the exchange potential is a good indicator of the interface between the A and B regions (see figure 2.4 center).

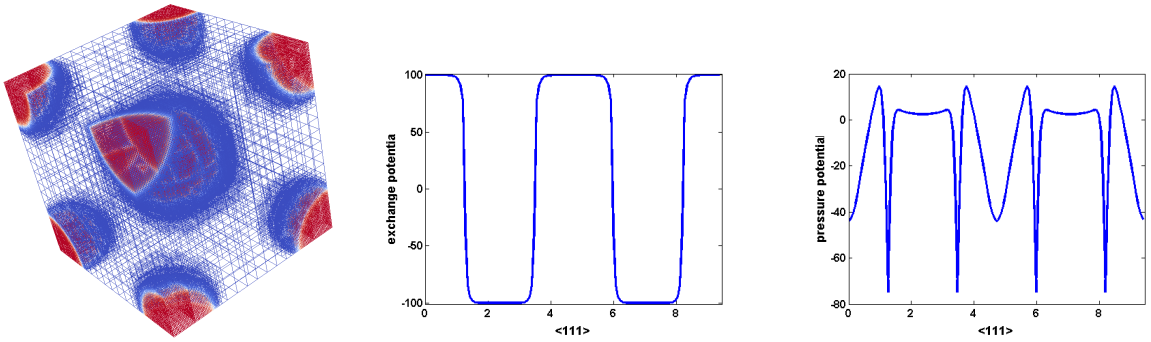


Figure 2.4: Example of a BCC morphology (left) with $f_A = 0.1, \chi_{AB}N = 200$ and the variations of the exchange potential w_- (center) and pressure potential w_+ (right) along the $< 111 >$ direction of the computational domain.

Now that we have outlined the refinement strategies given an interfacial distance function ϕ, we discuss how we construct ϕ. At the saddle point of the free energy H, the interface between the A an B components can be described by $w_- = 0$, so one could use it as an initial condition for the level-set reinitialization equation (2.14). However, $w_- = 0$ does not necessarily represent the interface between A and B in the early stages of the mean field steps. We thus use a different strategy based on the hybrid k-means/level-set algorithm of [30]: k-means is an efficient algorithm to segregate k-components from a data set by iteratively finding the centers of k clusters and assigning each point in the data set to a cluster in such a way that it minimizes the Euclidian distance between the data points to their respective cluster's center [55]. Only a few steps (typically between 2 and 4) of k-means steps are necessary, leading to a linear complexity. Once the k-means procedure is done, the clusters' boundaries are represented by a level-set function ϕ, as described in `Algorithm` 2. Once the level-set function is obtained with this step, it

is used to generate the adaptive grid. This refinement can naturally be embedded in a SCFT solver as described in `Algorithm` 3.

Algorithm 2 Hybrid level-set/k-Means [30]

Step 1: Input a data set $\boldsymbol{w}$. Set k ($k = 2$ in the case of diblock) and take an initial guess for the clusters' centers, c_k

Step 2:

while (c_i change) **do**

1. Compute for each data point j its Euclidian distance from c_i,
2. Assign each of the data points to the cluster with minimum distance.
3. Recompute the centers c_i as the centers of the clusters: $c_i = \frac{1}{N_i}\sum_{j=1}^{N_i} w_j$, where N_i is the number of points in cluster i.

Step 3: Reinitialize the level-set function using the reinitialization equation [92, 67].

As an illustration, figure 2.5 depicts typical adaptive grids that are automatically generated in the case of different morphologies. The lamellar phase is obtained with the pair ($f_A = 0.5, \chi_{AB}N = 20$), the cylinder phase is obtained with the parameters ($f_A = 0.18, \chi_{AB}N = 72$) and the gyroid phase is obtained with ($f_A = 0.36, \chi_{AB}N = 20$).

Algorithm 3 Optimization of H on an adaptive grid

Set F_- and F_+ defined in (2.6) arbitrarily large.

while $\max(F_+, F-) > \epsilon$ **do**

1. Create a binary data set from the potential.

2. Define the level-set function using the hybrid level-set/k-means procedure of `Algorithm` 2.

3. Build the adaptive grid according to the level-set.

4. Remap all the variables from the previous tree to the new tree with interpolation procedures [67].

5. Solve the two diffusion equations given in (B.4) from $s = 0$ to $s = 1$ with a step Δs.

6. Compute the densities ρ_A, ρ_B according to (2.3).

7. Compute the driving forces using (2.4).

8. Advance the potential w using (2.5).

9. Compute the free energy H and compute F_- and F_+ defined in (2.6).

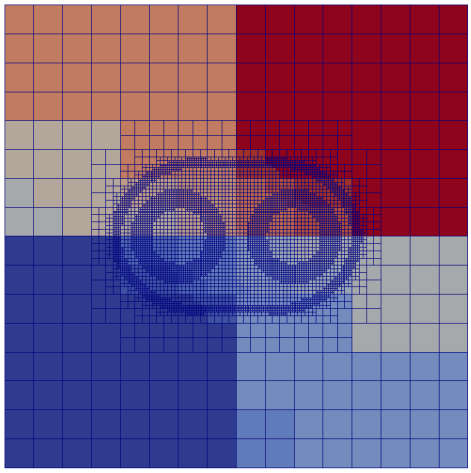
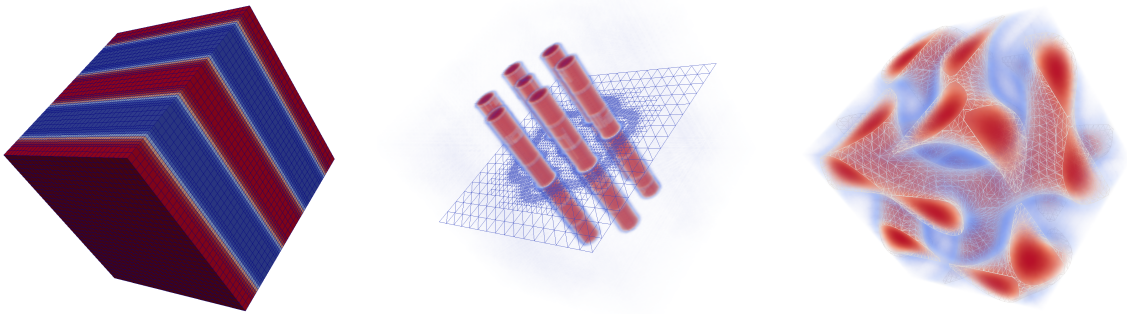


Figure 2.6: An adaptive quadtree grid equally partitioned between 16 processors to achieve load balance where each color represents a different processor.



(a) Lamellar phase (b) cylindrical phase (c) Gyroid phase

Figure 2.5: Adaptive grids automatically generated during an SCFT computation.

2.4 Parallel Computation Strategy - Forest of Octrees

Our solver uses the methodology outlined in [68], which is built on top of the open source `p4est` library [11], providing a suite of parallel algorithms for refining, coarsening, and partitioning Quadtree and Octree meshes. Once the grid is locally refined, as described in section 2.3.3, it is partitioned equitably across processes to achieve load balancing as can be seen in figure 2.6. After partitioning, each process obtains from its neighboring processes a layer of ghost cells, which is later used to update the solution values across processes. Finally, both the local and the ghost cells are utilized to construct a local hierarchical representation of the grid. This is done by starting from the root of all local trees and recursively searching and splitting local cells in the hierarchy data-structure such that all `p4est` cells have a corresponding cell in the local representation (see [68] for details).

Once the local representation is constructed, all the discretization operations can be performed locally. Furthermore, these local calculations can be overlapped efficiently with communication: local computations are performed simultaneously with message passing, such that processors do not need to wait passively for messages from other processors: hence "hiding" communication. The interested reader is referred to [68] for more details. We use the `PETSc` library for all linear algebra operations, for storing discretization matrices and for distributing ghost vectors. Finally, as detailed in [68], this

domain decomposition has been scaled up to 4096 processes (the limit of our account at that time) on the Stampede supercomputer at the Texas Advanced Computing Center (TACC).

2.5 Numerical Results

Sections 2.5.1-2.5.4 present numerical examples illustrating the two main novel aspects of this paper: (i) the capability to consider the self-assembly of block copolymer in confined domains with effective boundary conditions and (ii) the computational speedup provided by the parallel environment on Octree Cartesian grids. Section 2.5.5 will present the data compression results for high $\chi_{AB}N$ and low f_A in the case where one uses a pseudo-spectral method on a uniform grid while using the adaptive framework to store the chain propagators.

2.5.1 One Spatial Dimension

We first compare the results of the simulations with different numerical strategies in one spatial dimension in order to highlight the characteristics of each approach. The numerical methods we consider are the present hybrid finite difference/volume discretizations with effective boundary conditions (HFDV), the discrete Fourier transform (DFT) with a formulation that uses a mask (see section 2.2.2), and the discrete cosine transform (DCT). We note that HFDV and DCT use a strictly incompressible model. i.e.

$(\zeta N)^{-1} = 0$, while DFT employs a weakly compressible model, here $(\zeta N)^{-1} = 0.001$, to avoid stiffness in the solution near the mask's boundary.

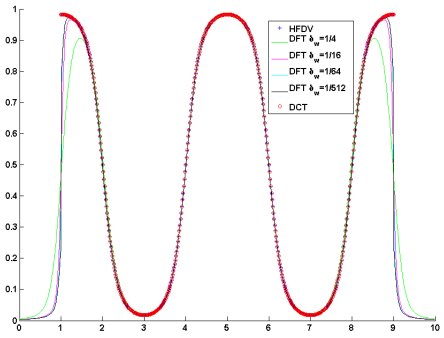
Recall that HFDV can be applied to either periodic or effective boundary conditions, in regular or irregular domains and on uniform or adaptive grids. DFT can be used with periodic boundary conditions but only on regular domains (and uniform grid). Considering irregular domain thus require the use of a mask with a compressible approximation, as discussed in section 2.2.2. DCT can be used with homogeneous Neumann boundary conditions but only on regular domains (and uniform grids); thus DCT cannot be used to consider confined domains. On the other hand, the accuracy of our HFDV is that of a finite difference scheme; it is not spectral as it is the case of DFT. The numerical results we present in one spatial dimension thus seek to answer the following questions:

1. How does HFDV compare to DCT in the case where effective boundary conditions are imposed? Even though DCT is not spectral, it is interesting to confirm that the same density profiles are obtained with HFDV and DCT, and that they have similar convergence behaviors.

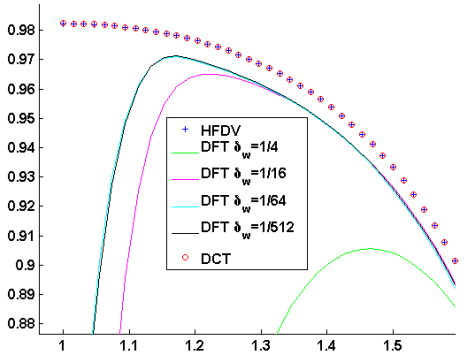
2. Confined domains can be simulated with periodic boundary conditions by using a mask and the compressible assumption. What are the implications of this approximation in terms of the accuracy near the confined walls? In particular, since the width δ_w of the mask's wall is a parameter that can be chosen, what is its effect on the simulated results and how do those results compare to the present method that is sharp?

Figures 2.7 and 2.8 depict the results of several simulations, from which we can answer the questions above. The simulations use a uniform grid with 512 points. The parameters used are $(f_A, \chi_{AB}N) = (.5, 20)$, $\Delta s = 0.0025$, $\lambda = 0.5$. We take $L_x = 8\,R_g$ for HFDV and DCT, and we take $L_x = 10R_g$ and $(\zeta N)^{-1} = 0.001$ for the compressible mask. R_g is the radius of gyration, which represents the root mean squared distance between the segments on the polymeric chain and the center of mass of the polymer chain. Figures 2.7(a) and 2.7(b) depict the density profiles of A. We distinguish two regions, one is near the wall and the other is in the bulk. In the bulk, all methods produce the same results. Near the wall, however, one observe qualitative differences between DFT and the other two approaches. First, figures 2.7(a) and 2.7(b) confirm that both HFDV to DCT correctly imposed the effective boundary condition. Also, the density profiles obtained from HFDV and DCT coincide. Figure 2.7(c) and 2.7(d) illustrate that the forces driving the optimization converge at the same rate until the numerical accuracy of HFDV prevents the computation of forces smaller than what the finite difference approximation enables at this grid resolution. Second, we observe in figures 2.7(a) and 2.7(b) that the effect of the mask in a DFT computation is apparent near the wall, which is assumed to be neutral, i.e. not attractive nor repulsive. Since the mask imposes an arbitrary density profile in a region of length δ_w, the solution in that region is meaningless. A consequence is that the use of the mask prevents the polymer to "wet" the wall of the confined domain. In addition, we observe that decreasing δ_w past 1/64 does not decrease the size of region where the DFT solution

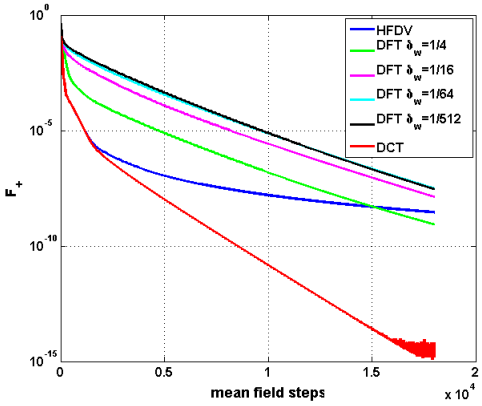
differs from those obtained with DCT and HFDV. Figures 2.7(c) and 2.7(d) also infer that decreasing the size of the wall's thickness slows down the convergence of the forces driving the optimization process due to the added numerical stiffness. Finally, we note that the size of the computed forces are characteristic of the accuracy of the numerical approximations and should not be confused with the accuracy of the simulation. For example, even though the forces in the case of DFT with $\delta_w = 1/4$ depicted in figure 2.7(d) reach the level of the HFDV simulation (and will even eventually be smaller), the resulting density converges to a profile that is arbitrary near the walls. The drawback of the mask representation can be significant in the case where one seeks to define meaningful quantities near the walls. For example, it is interesting to define a shape optimization process that takes into account w_+ near the walls [76, 78]. In a sharp HFDV approach, this quantity is well-defined, whereas it is less clear how this can be computed in the mask approach. Figure 2.8(a) depicts the maximum difference between the density obtained with HFDV on uniform grids with size from 16 to 2048 and the density obtained with a DCT simulation on a grid with 4096 points. Likewise, figure 2.8(b) depicts the results the same analysis for the free energy H. In both cases, good agreement is obtained.



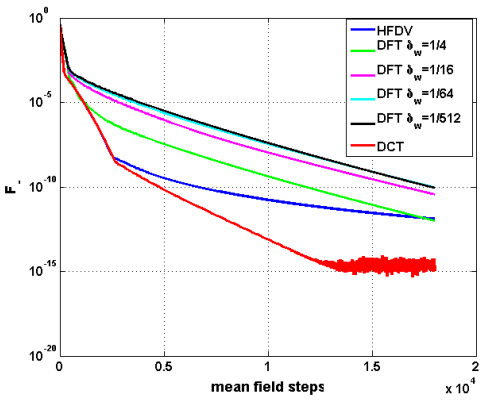
(a) Local volume fraction of A



(b) Zoom near the wall's boundary

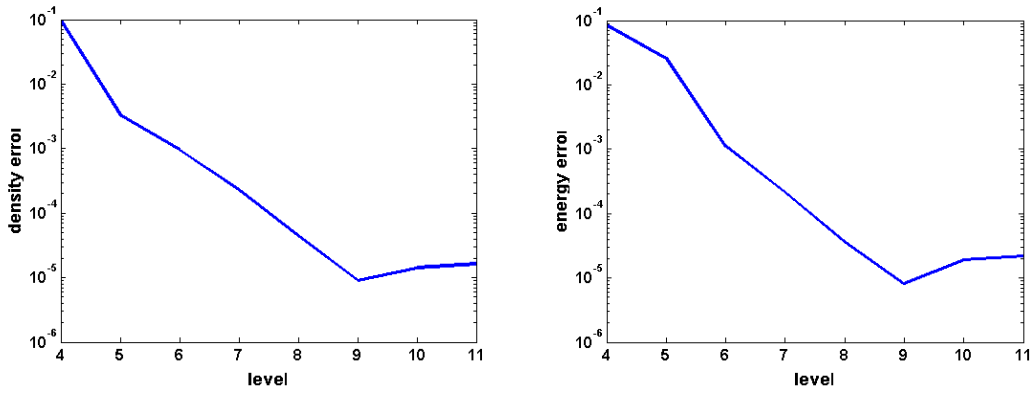


(c) Pressure force w_+



(d) Exchange force w_-

Figure 2.7: Comparison between the different numerical methods considered in the present manuscript.



(a) Error in the density.

(b) Error in the energy.

Figure 2.8: Errors in the density and energy in the L^∞-norm for the HFDV method using a uniform grid with varying levels of resolution from 2^4 to 2^{11}. The solutions are compared with a DCT simulation on a grid with 4096 points.

2.5.2 Two Spatial Dimensions

As a next step, we carried out simulations in two spatial dimensions to study onvergence on non-uniform grids and to compare the present approach to the conventional mask technique. We first consider a diblock copolymer confined in a cylindrical pore (see figure 2.9), which can be simulated in 2D if we assume the pore to be homogeneous in the z-direction. We then simulate a polymer confined in a "wavy" cylindrical pore with regions of high curvature (see figure 2.11). We present in figure 2.10 a detailed study of convergence for both the disk and the "wavy" geometries.

In these investigations, we consider Quadtree grids of size $(\min_{\text{level}}, \max_{\text{level}})$, where $\max_{\text{level}}$ controls the size of the *uniform* grid inside the confined domain and $\min_{\text{level}}$

controls the size of the coarsest cell of the *nonuniform* grid outside the confined domain. We also use Quadtree grids with size $(\min_{\text{level}}, \text{middle}_{\text{level}}, \max_{\text{level}})$, as depicted for example in figure 2.11 (a), where we use the $\min_{\text{level}}$ outside the confined domain, $\max_{\text{level}}$ at the domain's boundary and $\text{middle}_{\text{level}}$ inside the domain. In figure 2.13 (d), we use the $\min_{\text{level}}$ for the grid outside the confined domain, $\max_{\text{level}}$ inside the A minority region, near the AB interface and at the domain's boundary and $\text{middle}_{\text{level}}$ inside the B majority region. The results of figure 2.10(a) indicate that convergence is achieved on a level $(4, 8)$ Quadtree grid in the case where the geometry of the confined domain is a simple cylinder whereas figure 2.10(c) demonstrates that higher resolution is needed in the case of the "wavy" cylinder to capture the solution in regions of high curvature. In both cases, the energies (H) converge, see 2.10(b) and 2.10(d). The insets in figures 2.10(a) and 2.10(c) (resp. 2.10(d) and 2.10(b)) give, for the last mean field step, the differences (in the L^1-norm) in density (resp. energy) between the different grid refinement strategies and a uniform Quadtree grid with level 9 inside the domain. For both geometries, the density and energy converge under grid refinement. It is also interesting to observe the influence of the grid size, especially in the case where fine features occur near the confined domain. For example, in the case of the "wavy" geometry, one obtain better results in the case of a $(4, 7, 9)$ grid than in the case of a $(4, 8)$ grid. It is thus sufficient to consider coarser grids inside the confined domain, so long as a finer grid is placed near the boundary.

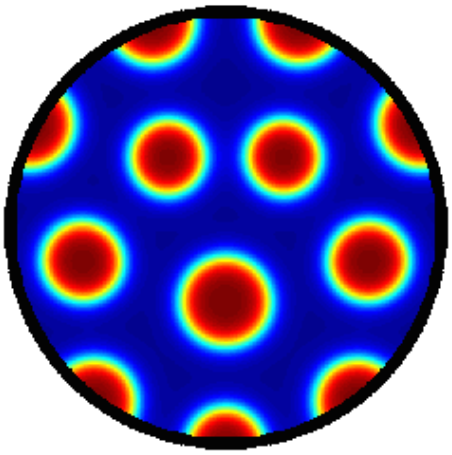
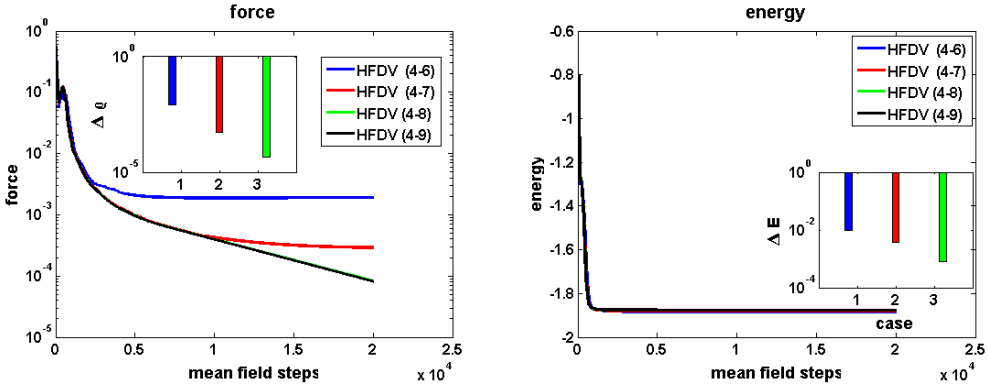


Figure 2.9: Densities obtained with a SCFT simulation in a confined domain using the present sharp computational approach. The parameters used are $(f_A, \chi_{AB}N) = (.3, 25)$, $\Delta s = 0.005$, $L_x = 16\,R_g$, $Diameter = 10.66\,R_g$, $\lambda = 0.25$. The grid levels are $(4, 9)$. The axes are in units of R_g.

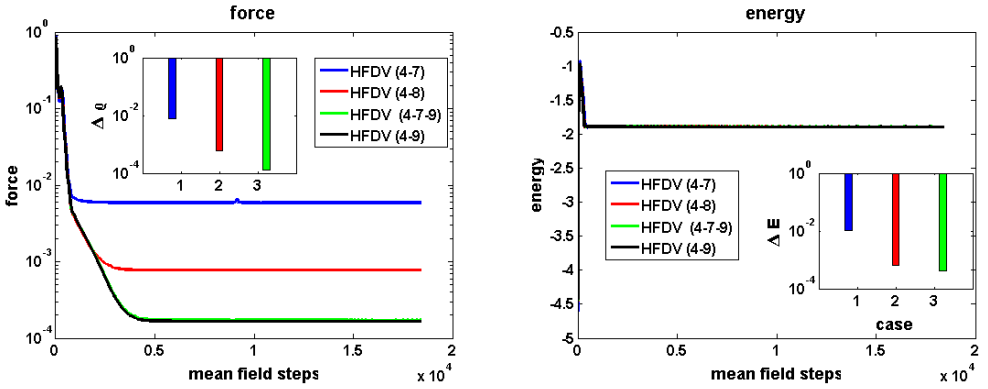
Figure 2.11 illustrates the results of SCFT simulations with both the HFDV method and the mask approache in the case of a confined domain with a "wavy" boundary. This case exemplifies the behavior of the solution near regions of high curvatures. We consider a seed with a radial symmetry to highlight the differences between the two approaches. As in the one-dimensional case, we observe that the use of a mask prevents the polymer to wet the confined domain's boundary. In the case of the sharp approach with effective boundary conditions, the polymer fully contacts the confined domain's boundary. We note again that one can improve the mask results by taking a sharper δ_w. In this case the polymer can penetrate the petals (see figures 2.11(c) versus 2.11(b)) but still not fully wet the walls. In the case of a mask approach, the parameter δ_w must thus be adjusted to the curvature of the confined domain's boundary. Figure 2.12 shows the optimization

forces evolution contrasting the sharp approach with effective boundary condition and with the mask approach.



(a) $\max(F_+, F_-)$ in the case of the disk

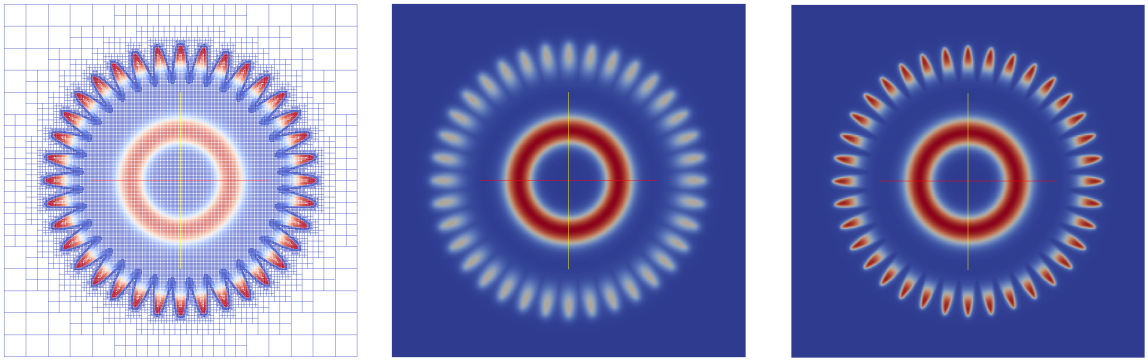
(b) Energy in the case of the disk



(c) $\max(F_+, F_-)$ in the case of the wavy geometry

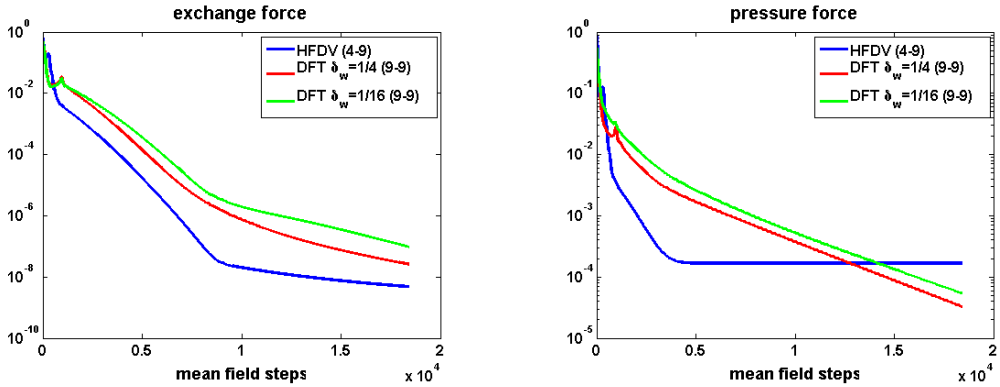
(d) Energy in the case of the wavy geometry

Figure 2.10: Convergence under grid refinement of the present method. The levels of the Quadtrees are $(4, 7)$, $(4, 8)$ and $(4, 9)$ in the case of the disk and $(4, 7)$, $(4, 8)$, $(4, 7, 9)$ and $(4, 9)$ in the case of the wavy geometry. (a) and (c) give the evolution of the L^2 norm of the force. (b) and (d) give the energy evolution as function of the number of mean field steps. The parameters are $f_A = 0.3$, $\chi_{AB}N = 25$, $\Delta s = 0.005$, $L_x = 16\, R_g$ and $\lambda = 0.25$.



(a) HFDV (b) mask $\delta_w = 0.25$ (c) mask $\delta_w = 0.0625$

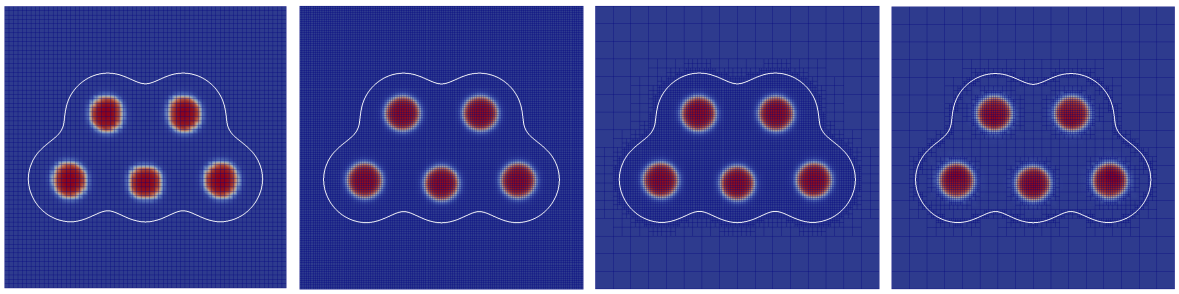
Figure 2.11: Densities of A obtained with HFDV and with a mask approach. For (a), we used a Quadtree with levels 4 outside the confined domain, level 7 inside and 9 near the wall using equation (2.12). (b) and (c) illustrate the effects of δ_w on the density in the mask approach. The parameters used are $f_A = 0.3$, $\chi_{AB}N = 25$, $\Delta s = 0.005$, $L_x = 16\,R_g$, $\lambda = 0.25$ and $\delta_w = 0.25$ and $(\zeta N)^{-1} = 0.001$ with a grid size of 512×512 for the mask. We chose a seed such that we get the same morphology with HFDV and the mask.



(a) F_- (b) F_+

Figure 2.12: Convergence of the mean-field optimization in the case of the "wavy" configuration. The parameters used are $f_A = 0.3$, $\chi_{AB}N = 25$, $\Delta s = 0.005$, $L_x = 16\,R_g$, $\lambda = 0.25$, $\delta_w = 0.25$ and $(\zeta N)^{-1} = 0.001$. The grid size is 512×512.

In figure 2.13 we provide another example where the geometry of a confined domain guides the self-assembly to five disks with radius $0.8\,R_g$, arranged in two rows. We consider four different refinement strategies and study the results of SCFT simulations using the HFDV method. In particular, the refinement shown in figures 2.13 (c) and (d) capture both the self-assembly features and the confinement geometry, but use far less degrees of freedom than the uniform grid of figure 2.13 (a). Table 2.1 clearly highlight that this strategy enables a speedup over a (7, 7) uniform grid without sacrificing accuracy. Other examples and further discussion of refinement criteria are given in section 2.5.5.



(a) (6, 6) grid (b) (7, 7) grid (c) (4, 7 grid) (d) (4, 4, 7) grid

Figure 2.13: SCFT simulation on Quadtrees using different refinement strategies. The parameters are $(f_A, \chi_{AB}N) = (.18, 72)$, $\Delta s = 0.01$, $\lambda = 0.5$ and $L_x = 12.8\,R_g$.

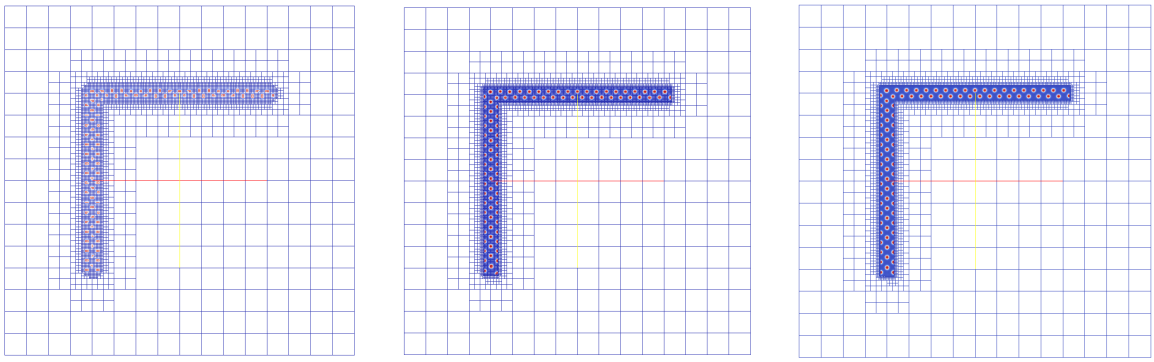
2.5.3 L-Shaped Confined Domain in Two Spatial Dimensions

As a further illustration, we consider a confined domain with a L-shaped geometry and present speedups obtained with both approaches in a parallel environment. In this

	(6, 6)	(7, 7)	(4, 7)	(4,4,7)
$\Delta H/H$	0.17	0	0	0.0016
$\|\Delta\rho\|_2$	0.02	0	0	0.007
$\|\Delta\rho\|_\infty$	0.32	0	0	0.09
N/N_{uniform}	0.25	1	0.42	0.22
t/t_{uniform}	0.19	1	0.46	0.26

Table 2.1: Comparison of quantities obtained with a uniform (7, 7) grid and different adaptive grids. $\Delta H/H$ is the relative energy error, $\Delta\rho$ is the error in density, N is the total degrees of freedom and t is the computational time per mean field step on the adaptive grid. N_{uniform} and t_{uniform} are defined similarly for the uniform (7, 7) grid.

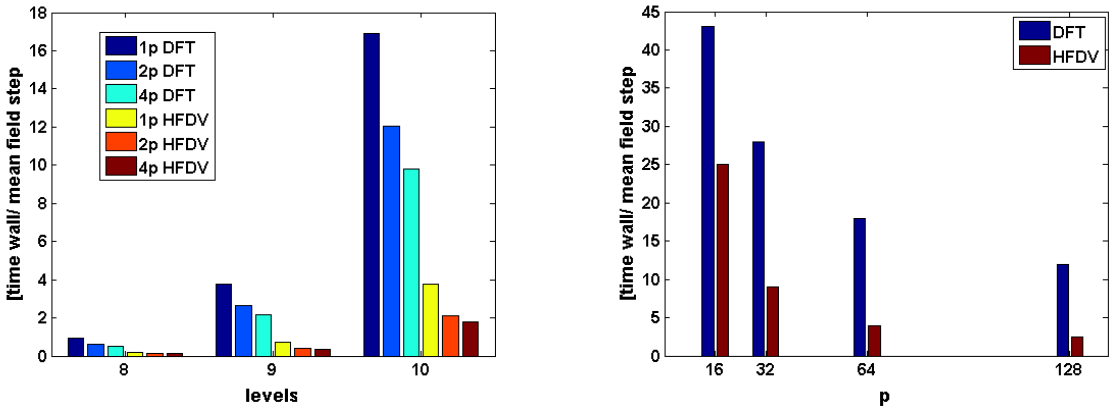
section and figure 2.16, we consider confined domains with geometries that occupy a small part of the computational domain, for which the adaptive finite difference approach is computationally more efficient than the mask approach. For confinement shapes that cover a large part of the computational domain, the mask approach is computationally more efficient, owing to the `FFTW` [25, 26]



(a) Quadtree with levels (4, 8) (b) Quadtree with levels (4, 9) (c) Quadtree with levels (4, 10)

Figure 2.14: Convergence of a SCFT simulation under grid refinement for an L-shaped domain using the present sharp method. The parameters are $\chi_{AB}N = 36$, $f_A = 0.18$, $\Delta s = 0.01$, $\lambda = 0.5$ and $L_x = 128\, R_g$.

Figure 2.14 depicts the results of the sharp approach on adaptive grids, which indicate that a minimum resolution of $(4, 9)$ is needed to capture the fine features of the self-assembly's morphology and the center-to-center distances between features. The effective resolution, i.e. the size of the smallest grid cell corresponds to a 512×512 uniform grid. We have checked that a 512×512 uniform grid is also needed in the case of the mask approach to capture the same features. In such cases, where the confined domain occupies a small portion of the computational domain, the adaptive approach has a computational advantage since the number of degrees of freedom is significantly lower than in the pseudo-spectral approach.



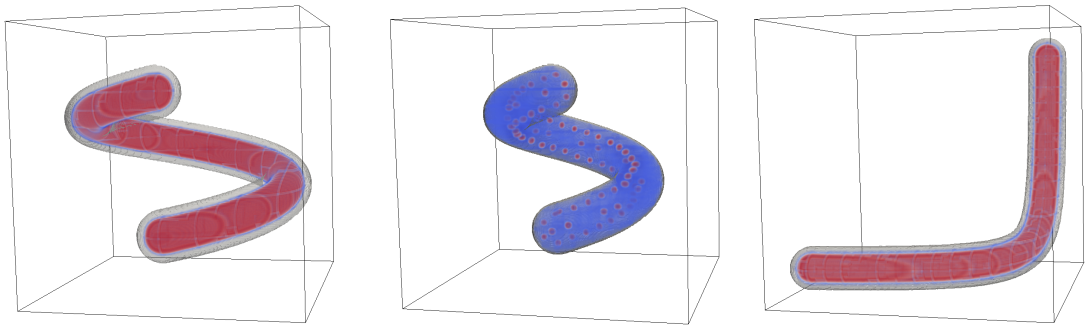
(a) speedup on a pc

(b) speedup on a cluster

Figure 2.15: Comparison of wall time between HFDV and the mask approach (using a DFT) for a SCFT simulation in terms of grid levels and number of processors. (a) Comparison on a shared memory machine. (b) Comparison on a distributed memory cluster. The parameters for these simulations are $\chi_{AB}N = 36, f_A = 0.18, \Delta s = 0.01, \lambda = 0.5, L_x = 128R_g$.

In terms of computational efficiency, figure 2.15 provides a summary of speedups for the two approaches in a parallel environment. We find a net speedup of about 4 in the case of HFDV compared to mask approach for levels (4, 8), (4, 9) and (4, 10) using 1, 2 or 4 processors on a shared memory machine. We also find a speedup for levels (4, 12) on 16, 32, 64 and 128 processors on a distributed memory cluster. Figure 2.15 (a) highlight the weak scaling typical in scientific computing: the speedup with finite difference solver plateaus when we use 4 processors instead of 2 for grid levels 9 and 10 while the mask approach still enjoys a speedup. This is due to the fact that the number of degrees of freedom in the case of the mask approach is much larger than in the case of the adaptive grid. As a consequence, the computational time spent by each process dwarf the communication time. In the case of the adaptive approach, each process is assigned less grid points and in turn the cost of communication is more apparent. Nevertheless, for this example where the confined domain occupies a small region of the entire computational domain, the HFDV approach is more efficient than the mask on both shared and distributed machines. In fact, for this example, HFDV on one processor is faster than DFT on 4 processors. On figure 2.15 (b), we show that for levels (4-12), HFDV is always faster than the solver based on DFT.

2.5.4 Three Spatial Dimensions



(a) Helix with $(f_A, \chi_{AB}N) = (.25, 40)$

(b) Helix with $(f_A, \chi_{AB}N) = (.1, 54)$

(c) 3D "L-shaped" domain

Figure 2.16: Densities obtained with a SCFT simulation using HFDV in parallel on a (4-8) Octree. The parameters for (a) are $(f_A, \chi_{AB}N) = (.25, 40)$, $\Delta s = 0.0025$ and $\lambda = 0.25$. The parameters for (b) are $(f_A, \chi_{AB}N) = (.1, 54)$, $\Delta s = 0.01$ and $\lambda = 0.5$. The parameters for (c) are $(f_A, \chi_{AB}N) = (.3, 25)$, $\Delta s = 0.01$ and $\lambda = 0.5$. In all cases $L_x = 64\, R_g$.

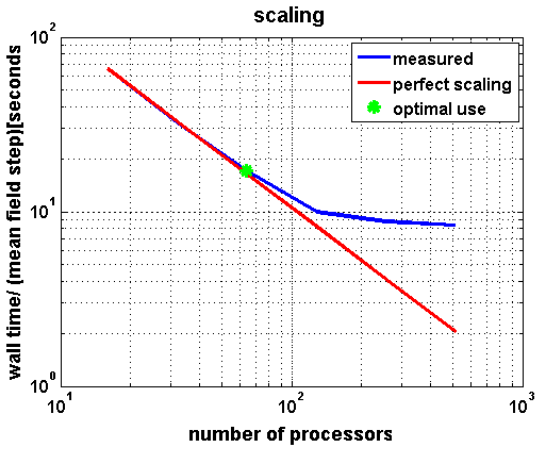


Figure 2.17: Parallel scaling for the example of figure 2.16.

In this section, we present results of simulations in three spatial dimensions using the HFDV approach. The first example considers an helix-shaped confined domain. Figure 2.16 depicts the confined domain and the densities obtained with different pairs of $(f_A, \chi_{AB}N)$, which lead to a self-assembled materials that follows the structure of the confined domain (figure 2.16(a)) or the generation of spheres (2.16(b)). Figure 2.17 shows the parallel scaling on a distributed memory cluster for 100 mean-field steps on a level $(4, 8)$ Octree grid, which corresponds to 1,547,301 degrees of freedom and an effective resolution of a 256^3 uniform grid. One can observe a linear scaling up to 64 processors, followed by a plateau. This illustrates a typical weak scaling, where the cost of communication dominates the computation beyond a given number of processes for a particular problem size. For this particular problem size, the optimal number of processes is 64. For a simulation with 64 processors, the average computational time for one mean field step is 15 seconds for the HFDV approach versus 60 seconds for the pseudo spectral approach in the case of figure 2.16(a). The factor of 4 speedup for HFDV is rooted in the sparsity of the Quadtree grid that requires only for 9% of a 256^3 uniform grid (i.e. 16.8 M degrees of freedom). For the example on figure (2.16(b)) the average time for a mean field step on 128 processecors is 2 seconds for the HFDV approach versus 12 seconds for the `FFTW` on the corresponding 256^3 uniform grid, hence a speed-up of 6 in favor of HFDV.

We consider a final example in three spatial dimensions on a level $(4, 8)$ Octree grid, with a "3D L-shaped" confined domain depicted in figure 2.16(c). Again in this case the

average time for a mean field step on 128 processors is 2 seconds for the HFDV approach versus 12 seconds for the `FFTW`, hence a speed-up of 6 in favor of HFDV.

We emphasize again that the computational superiority of the adaptive finite difference approach is for confined domains that occupy a small portion of the computational domain.

2.5.5 Hybrid Uniform/Adaptive Pseudo-Spectral Approach

In the case where the computational domain is periodic and densely filled with block copolymer, i.e., bulk periodic mesophases, it is advantageous to consider a pseudo-spectral approach because of its computational efficiency. In this case, one can take advantage of the nature of the SCFT algorithm and the data structure hierarchy to compress the storage requirements of a simulation. There are two kinds of iterations in a SCFT simulation: those associated with the diffusion solver and those associated with the mean field optimization. The mean field iterations only require the potentials $w_-(\boldsymbol{r})$ and $w_+(\boldsymbol{r})$, the forces $f_+(\boldsymbol{r})$ and $f_-(\boldsymbol{r})$ and the densities $\rho_A(\boldsymbol{r})$ and $\rho_B(\boldsymbol{r})$. These quantities are all $\boldsymbol{r}$-dependent, but s-independent. On the other hand, even though the diffusion iterations evolve quantities $q(s,\boldsymbol{r})$ and $q^\dagger(s,\boldsymbol{r})$ from $s=0$ to $s=1$, one needs to store the entire sequence $q(s,\boldsymbol{r})$ and $q^\dagger(1-s,\boldsymbol{r})$ in s for each grid point in order to compute the densities and the forces. The key point is that one can use a uniform grid to solve the diffusion iterations (e.g. in order to use a `FFT` algorithm for fast computations) but the adaptive grid is enough to store the data, hence reducing the memory footprint.

The densities, forces and potentials are subsequently only computed at the nodes of the adaptive grid. The values on the uniform grid that are required are interpolated from the adaptive grid using the procedures described in [67]. Conversely the storage does not require interpolations since the Quad-/Oc-tree nodes corresponds to grid points on the uniform grid. This strategy combines the advantages of a spectral convergence and the reduction of memory on adaptive grids. This strategy is illustrated in `Algorithm` 4.

Algorithm 4 Diffusion Iteration FFT on uniform grid but storage of q on an AMR grid

1. $q_{\boldsymbol{n}}^{s+1/3} = \exp(-w_{\boldsymbol{n}}\frac{\Delta s}{2})q_{\boldsymbol{n}}^{s}$, $\boldsymbol{n} \in$ Uniform Grid
2. $\hat{q}_{\boldsymbol{k}}^{s+1/3} = \texttt{FFT}(q_{\boldsymbol{n}}^{s+1/3})$, $\boldsymbol{k}, \boldsymbol{n} \in$ Uniform Grid
3. $\hat{q}_{\boldsymbol{k}}^{s+2/3} = \exp(-4d\pi^2k^2)(\hat{q}_{\boldsymbol{k}}^{s+1/3})$, $\boldsymbol{k}, \boldsymbol{n} \in$ Uniform Grid
4. $q_{\boldsymbol{n}}^{s+2/3} = \texttt{IFFT}(\hat{q}_{\boldsymbol{k}}^{s+2/3})$, $\boldsymbol{k}, \boldsymbol{n} \in$ Uniform grid
5. $q_{\boldsymbol{n}}^{s+1} = \exp(-w_{\boldsymbol{n}}\frac{\Delta s}{2})q_{\boldsymbol{n}}^{s+2/3}$ $\boldsymbol{n} \in$ Uniform Grid
6. store $q_{\boldsymbol{n}}^{s+1}$ $\boldsymbol{n} \in$ Adaptive Grid

In a distributed environment the mapping from the uniform `FFT` grid to the AMR grid is complicated by data distribution. An efficient strategy is required to map from the uniform grid to the adaptive grid and to send and receive off-process data. In this case, we use parallel index sets and scattering contexts of the PETSC library [4] to manage parallel vectors related to unstructured grids. To each node on the adaptive grid we calculate the node on the uniform grid on global numbering. The scattering context stores two parallel index set vectors (not of the same size) and two parallel data

vectors. The scattering uses these two pairs of vectors to transfer and map off-processor information from the adaptive grid to the uniform grid without any interpolation. We note that it is still necessary to interpolate the potential from the adaptive grid to the uniform grid but this is done only once at each mean field step instead of at each diffusion iteration.

Refinement Criteria	tree levels	α	$\|\cdot\|_\infty$	$\|\cdot\|_2$	$\Delta H/H[\%]$
Criterion 1	(6-6)	1	0	0	0
Criterion 1	(5-5)	8	1.07×10^{-1}	1.7×10^{-2}	5×10^{-3}
Criterion 2	(4-6)	3.72	1.4×10^{-2}	6×10^{-4}	7×10^{-2}
Criterion 3	(4-6)	3.16	2.7×10^{-3}	2.07×10^{-4}	5×10^{-2}
Criterion 4	(4-6)	2.62	1.5×10^{-3}	1.2×10^{-4}	1×10^{-2}

Table 2.2: Simulation results comparing the density $\rho_A(\boldsymbol{r})$ obtained with a pseudo-spectral solver and the storage on an adaptive grid with the density obtained with the full pseudo-spectral solver for a BCC sphere phase in a 3D unit cell. The grid for the pseudo spectral solver is (6, 6) and the parameters used are $(f_A, \chi_{AB}N) = (.14, 100)$ and $L_x = 5.4559\, R_g$. The compression ratio α is defined as $N_{uniform}/N_{amr}$, where $N_{uniform}$ and N_{amr} are the number of grid points on the uniform and adaptive grids, respectively.

Table 2.2 compares the densities of the converged field between different methods and demonstrates that when using the pseudo-spectral method with an adaptive storage, memory compression is obtained while accuracy, expressed both as the L_∞ and L_2 norms of the error in the density, is kept. Using `Algorithm` 2 with w_- and w_+, we obtain two functions $\phi(w_-)$ and $\phi(w_+)$, from which we defined the following refinement criteria:

- **Criterion 1**: A uniform grid is imposed in the entire computational domain.

- **Criterion 2**: The highest resolution is imposed only near the boundary between the minority and majority regions using equation (2.12) with $\phi(w_-)$.

- **Criterion 3**: Same as criterion 2 and in addition, the grid is uniform inside the minority region.

- **Criterion 4**: The highest resolution is imposed using equation (2.12) with both $\phi(w_-)$ and $\phi(w_+)$ and the grid is uniform inside the minority region.

In all cases the Octree grid provides a compression of the data and the error is below 2%. Specifically, we define the compression ratio as $\alpha = \frac{N_{uniform}}{N_{amr}}$, where N and N_{amr} are the number of grid points on the uniform and adaptive grids, respectively. We also used this computational strategy with a full mean field optimization for an Octree with $(f_A, \chi_{AB}N) = (.1, 200)$ (see figure 2.4 (a)) and found a compression ratio of about 5 without sacrificing accuracy.

2.6 Conclusion

We have presented a hybrid finite difference/volume method for conducting self-consistent field theory simulations of self-assembly of diblock copolymers under confinement with an effective Neumann boundary condition. In addition, we have implemented adaptive grids encoded by Quad-/Oc-trees and extended the method to parallel environments for increased computational efficiency. The advantage of this approach is that it provides a computationally and memory efficient way to tackle self-assembly within

irregular confined domains in two and three spatial dimensions. The method is further shown to be an attractive alternative to the use of a mask technique that imposes uniform resolution and mandates a weakly compressible model. In particular, the development of a sharp numerical approach approximating the effective boundary condition enables robust computations of the solution and its derivative near the boundary. Future work will explore the use of this promising HFDV technique to investigate diblock copolymer assembly in a domain with a free surface.

Chapter 3

Functional Level Set Derivative

We derive functional level-set derivatives for the Hamiltonian arising in self-consistent field theory, which are required to solve free boundary problems in the self-assembly of polymeric systems such as block copolymer melts.. In particular, we consider in detail the case of Dirichlet, Neumann and Robin boundary conditions. We provide numerical examples that illustrate how these shape derivatives can be used to find equilibrium and metastable structures of block copolymer melts with a free surface in both two and three spatial dimensions.

3.1 Introduction

The use of polymers in confined situations is ubiquitous in science and engineering, ranging from novel lithography techniques at the sub 22-nanometer scale for next generation computer chips to the study of drug delivery carriers with controlled release to the

study of thin films [47, 50, 49, 52, 84, 83, 46, 70]. Self-consistent field theory (SCFT) provides an accurate description of the self-assembly of dense collections of long polymers at equilibrium by considering a Fokker-Planck equation that describes the probability of polymer segments to be at a certain location and given chain contour location. This technique has been successfully used in the case of periodic as well as confined domains. The more difficult case of free boundaries is still in its infancy, although some authors have recently provided specialized solutions in the context of directed self-assembly, by considering a parametric description of the free boundary [54, 53]. We focus here on the versatile free boundary description proposed by Ouaknin *et al.* [77, 78, 76], which is based on an implicit level-set representation [73, 71, 86].

An important question in the context of free surfaces is how the self-assembled solution responds to a change in the free boundary's *shape*. Answering this question requires to either explicitly solve the direct problem several times in order to *a posteriori* estimate the sensitivity or to derive appropriate functional derivatives with respect to the level set to *a priori* estimate it. The advantage of an *a priori* approach is that it enables the development and implementation of algorithms that optimize for the coupling between the fields and the shape. In the context of SCFT, functional shape derivatives have been derived for orthorhombic shapes only [6]. In the present paper, we derive functional shape derivatives with arbitrary shapes.

Shape derivatives have been derived for many systems described by partial differential equations with various boundary conditions; a rigorous derivation of the methods and

examples can be found in the seminal book of Sokolowski and Zolesio [88]. The level-set framework has also been used to study shape optimization of a variety of physical [74, 10, 1, 13], but no work exists for free boundary polymer self-assembly within the framework of SCFT. The present work fills this gap in the literature.

The paper is structured as follows: in section 3.2 we present the SCFT equations, in section 3.3 we derive functional level-set derivatives for SCFT for various boundary conditions and in section 3.4 we present examples of shape optimization using these level-set derivatives in both two and three spatial dimensions.

3.2 Self-Consistent Field Theory Model

In this section, we describe the governing equations for self-consistent field theory [24, 35, 36, 57] in the context of an incompressible melt of diblock copolymers composed of two monomer species A and B. This physical system is characterized by two parameters: (1) f_A, the fraction of block A and (2) χ_{AB}, the repulsion interaction between A and B polymer segments. The field-based theory of polymer thermodynamics uses a Hubbard-Stratonovich transform to convert the partition function from a particles representation to a field representation. If one further imposes the mean-field, or self-consistent field, approximation, the SCFT equations are obtained by finding for the statistical fields w_-

and w_+ that optimize the Hamiltonian[1]:

$$H[w_+, w_-] = \frac{1}{V}\int_\Omega \Big(- w_+(\boldsymbol{r}) + \frac{w_-^2(\boldsymbol{r})}{\chi_{AB}}\Big)\, d\boldsymbol{r} - \ln Q[w_+(\boldsymbol{r}), w_-(\boldsymbol{r})], \tag{3.1}$$

where the normalized partition function Q can be formulated as:

$$Q[w_+, w_-] = \frac{1}{V}\int_\Omega q(\boldsymbol{r}, s=1; [w_+(\boldsymbol{r}), w_-(\boldsymbol{r})])\, d\boldsymbol{r}.$$

In these equations, $w_+(\boldsymbol{r})$ acts as a pressure potential that enforces the incompressibility constraint while the exchange potential, $w_-(\boldsymbol{r})$, describes the interaction between A and B. The functionals $q(\boldsymbol{r}, s=1; [w_+, w_-])$ and $q^\dagger(\boldsymbol{r}, s=1; [w_+, w_-])$ are the solution of a Fokker Planck equation, which has to be solved from $s=0$ to $s=1$, with initial conditions $q(s=0, \boldsymbol{r}) = q^\dagger(s=0, \boldsymbol{r}) = 1$, where s is the contour chain. The functionals q and $q^\dagger$, called the chain propagators, represent the statistical weight of piece of chain at location $\boldsymbol{r}$ and contour length s and satisfy the following forward and backward Fokker-Planck equations:

$$\begin{cases} \partial_s q(s, \boldsymbol{r}) = \Delta q(s, \boldsymbol{r}) - q(s, \boldsymbol{r})\, w(\boldsymbol{r}, s) \qquad \text{forward.} \\ \\ \partial_s q^\dagger(s, \boldsymbol{r}) = \Delta q^\dagger(s, \boldsymbol{r}) - q^\dagger(s, \boldsymbol{r})\, w^\dagger(\boldsymbol{r}, s) \quad \text{backward.} \end{cases} \tag{3.2}$$

The boundary conditions at the free boundary that we consider in this study are:

1. Homogeneous Dirichlet:

$$q(s, \boldsymbol{r}) = 0 \quad \text{and} \quad q^\dagger(s, \boldsymbol{r}) = 0.$$

[1]This is actually H/n, where n is the number of molecular chains. Likewise χ_{AB} is in fact $\chi_{AB}N$ where N is the number of molecular segments per chain.

2. Homogeneous Neumann:

$$\frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}} = 0 \quad \text{and} \quad \frac{\partial q^{\dagger}(s,\boldsymbol{r})}{\partial \boldsymbol{n}} = 0.$$

3. Homogeneous Robin:

$$q(s,\boldsymbol{r}) + \alpha_A \frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}} = 0, \qquad \text{for } 0 < s < f_A \qquad \text{and}$$

$$q(s,\boldsymbol{r}) + \alpha_B \frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}} = 0, \qquad \text{for } f_A < s < 1,$$

$$q^{\dagger}(s,\boldsymbol{r}) + \alpha_B \frac{\partial q^{\dagger}(s,\boldsymbol{r})}{\partial \boldsymbol{n}} = 0, \quad \text{for } 0 < s < 1 - f_A \quad \text{and}$$

$$q^{\dagger}(s,\boldsymbol{r}) + \alpha_A \frac{\partial q^{\dagger}(s,\boldsymbol{r})}{\partial \boldsymbol{n}} = 0, \quad \text{for } 1 - f_A < s < 1.$$

where we can denote $\alpha(s)$ for α_A or α_B in the Robin boundary condition for q and $q^{\dagger}$ which describe the attraction or the repulsion from the wall for the A and B species [24]. The forward potential $w(\boldsymbol{s},\boldsymbol{r})$ is calculated as follows:

$$w(s,\boldsymbol{r}) = \begin{cases} w_A(\boldsymbol{r}) = w_+(\boldsymbol{r}) - w_-(\boldsymbol{r}) & 0 < s < f_A \\ \\ w_B(\boldsymbol{r}) = w_+(\boldsymbol{r}) + w_-(\boldsymbol{r}) & f_A < s < 1 \end{cases}, \tag{3.3}$$

and $w^\dagger(s,\boldsymbol{r})$ is defined as $w^\dagger(s,\boldsymbol{r}) = w(1-s,\boldsymbol{r})$. The normalized densities of the components A and B are obtained through the functional derivatives with respect to $w_A(\boldsymbol{r})$ and $w_B(\boldsymbol{r})$, $\dfrac{\delta H}{\delta w_A(\boldsymbol{r})}, \dfrac{\delta H}{\delta w_B(\boldsymbol{r})}$ and are computed using q and $q^\dagger$:

$$\rho_A(\boldsymbol{r};[w_+,w_-]) = \frac{1}{Q[w_+,w_-]}\int_0^{f_A} q^\dagger(1-s,\boldsymbol{r},[w_+,w_-])q(s,\boldsymbol{r},[w_+,w_-])\,ds$$

$$\rho_B(\boldsymbol{r};[w_+,w_-]) = \frac{1}{Q[w_+,w_-]}\int_{f_A}^{1} q^\dagger(1-s,\boldsymbol{r},[w_+,w_-])q(s,\boldsymbol{r},[w_+,w_-])\,ds$$

In the mean-field or SCFT approximation, where fluctuations are neglected, the free energy of the melt is then given by the Hamiltonian at the saddle point, i.e. when $\dfrac{\delta H}{\delta w(\boldsymbol{r})} = 0$. The forces of the optimization process are obtained through the functional derivatives with respect to the statistical fields, $\dfrac{\delta H}{\delta w_+(\boldsymbol{r})}, \dfrac{\delta H}{\delta w_-(\boldsymbol{r})}$, and are given by:

$$\begin{cases} f_+(\boldsymbol{r}) = \rho_A(\boldsymbol{r}) + \rho_B(\boldsymbol{r}) - 1 \\ \\ f_-(\boldsymbol{r}) = \dfrac{2w_-(\boldsymbol{r})}{\chi_{AB}} + \rho_B(\boldsymbol{r}) - \rho_A(\boldsymbol{r}) \end{cases} . \tag{3.4}$$

3.3 Level-Set Representation and Functional Level-Set Derivatives

In this work, the shape is described through a level-set function, $\phi(\boldsymbol{r})$, which describes an irregular domain, Ω, in an implicit way (see figure 3.1): $\phi(\boldsymbol{r})$ is negative inside Ω, positive outside, and zero at its boundary Γ. The Fokker-Planck equations are thus solved in Ω where $\phi < 0$ with the boundary conditions imposed at the interface Γ, where

$\phi(\boldsymbol{r}) = 0$. The volume of the polymer V is equal to $|\Omega| = \int \mathcal{H}(-\phi(\boldsymbol{r}))d\boldsymbol{r}$ where $\mathcal{H}$ is the Heaviside function. In this study we choose $\phi(\boldsymbol{r})$ to be a signed distance function, i.e. $|\nabla\phi(\boldsymbol{r})| = 1$. The outward normal to Ω is thus defined as $\boldsymbol{n} = \nabla\phi(\boldsymbol{r})$, the local mean curvature is $\kappa(\boldsymbol{r}) = \nabla \cdot \nabla\phi(\boldsymbol{r})$ and $\dfrac{\partial q(\boldsymbol{r})}{\partial \boldsymbol{n}} = \nabla q \cdot \nabla\phi$.

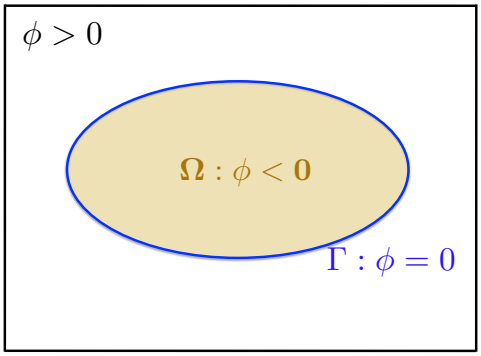


Figure 3.1: Level-set representation of an irregular domain Ω with boundary Γ.

With such a representation, the Hamiltonian is a functional of $\phi(\boldsymbol{r})$, in addition to the pressure field $w_+(\boldsymbol{r})$ and the exchange field $w_-(\boldsymbol{r})$; thus one writes $H = H[w_+, w_-, \phi]$. To quantify how the Hamiltonian H responds to a change in the irregular domain's shape, we need to find how Q changes and in turn how the chain propagators $q(s = 1, \boldsymbol{r})$ and $q^\dagger(s = 1, \boldsymbol{r})$ vary. The change in the shape is expressed through a perturbation of the interface Γ via the level-set function $\phi(\boldsymbol{r}) \to \phi(\boldsymbol{r}) + \delta\phi(\boldsymbol{r})$ (see figure 3.2). In turn we need to find how $q(s = 1, \boldsymbol{r}) \to q(s = 1, \boldsymbol{r}) + \delta q(s = 1, \boldsymbol{r})$ and $q^\dagger(s = 1, \boldsymbol{r}) \to q^\dagger(s = 1, \boldsymbol{r}) + \delta q^\dagger(s = 1, \boldsymbol{r})$. In what follows, we derive the shape derivative for H in the weak form and provide in the appendix the derivation in the strong form.

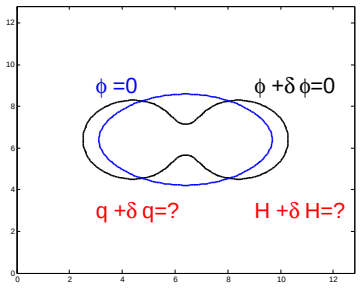


Figure 3.2: Schematic illustrating how $q(\boldsymbol{r}, s)$, the solution of $\partial_s q = \nabla^2 q - wq$, varies to $q+\delta q$, and by consequence how the free energy H varies to $H + \delta H$, as the shape is perturbed from ϕ to $\phi + \delta\phi$.

3.3.1 Shape Derivative Theorems

For completeness, we recall the shape derivative theorems that are required for the current study and we refer the interested reader to [88] for detailed proofs.

Theorem 1: Consider a functional, $J[u(\boldsymbol{r})] = \int_\Omega u(\boldsymbol{r}) d\boldsymbol{r}$, defined in Ω. Given a change $\delta\phi(\boldsymbol{r})$ of the shape, the corresponding change of the functional is given by:

$$\delta J[u(\boldsymbol{r})] = \int_\Omega \delta u(\boldsymbol{r}) d\boldsymbol{r} - \int_\Gamma u(\boldsymbol{r}) \delta\phi(\boldsymbol{r}) d\Gamma,$$

where the integral over Γ is over the original shape's boundary (unchanged).

Theorem 2: Consider a functional, $J[u(\Gamma)] = \int_\Gamma u(\Gamma)d\Gamma$, defined on Γ. Given a change $\delta\phi(\boldsymbol{r})$ of the shape, the change of the functional is given by:

$$\delta J[u(\Gamma)] = \int_\Gamma \delta u(\Gamma)d\Gamma - \int_\Gamma \kappa(\boldsymbol{r})u(\Gamma)\delta\phi(\boldsymbol{r})d\Gamma.$$

In the particular case that $u(\Gamma) = u(\boldsymbol{r})\Big|_\Gamma$ and $\delta u(\Gamma) = \delta u(\boldsymbol{r})\Big|_\Gamma - \dfrac{\partial u}{\partial \boldsymbol{n}}(\boldsymbol{r})\Big|_\Gamma \delta\phi(\boldsymbol{r})$, the change of the functional can be rewritten as:

$$\delta J(u) = \int_\Gamma \delta u(\boldsymbol{r})\Big|_\Gamma d\Gamma - \int_\Gamma \Big(\frac{\partial u(\boldsymbol{r})}{\partial \boldsymbol{n}}\Big|_\Gamma + \kappa(\boldsymbol{r})u(\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r})d\Gamma,$$

where $\kappa(\boldsymbol{r})$ is the curvature of Γ and as in Theorem 1 the integrals over Γ are over the original boundary shape (unperturbed).

Theorem 3: This theorem gives a formula for integrating by parts on the surface Γ:

$$\int_\Gamma \nabla f(\boldsymbol{r})\cdot\nabla g(\boldsymbol{r})d\Gamma = \int_\Gamma -f(\boldsymbol{r})\cdot\Delta g(\boldsymbol{r})d\Gamma + \int_\Gamma [\frac{\partial f(\boldsymbol{r})}{\partial \boldsymbol{n}}\frac{\partial g(\boldsymbol{r})}{\partial \boldsymbol{n}} + \frac{\partial^2 g}{\partial \boldsymbol{n}^2} + \kappa(\boldsymbol{r})f\frac{\partial g}{\partial \boldsymbol{n}}]d\Gamma,$$

We note that we have assumed in theorems 1-3 that the level-set function $\phi(\boldsymbol{r})$ is a distance function i.e, $\boldsymbol{n} = \nabla\phi$, $|\nabla\phi| = 1$ and more importantly that the change of the shape is measured by $\delta\phi(\boldsymbol{r})$.

3.3.2 Overview

In [12], Cea introduced a method based on a Lagrangian functional to calculate the shape derivative of a functional $J[\phi, u]$, where $u(\boldsymbol{r})$ is solution of a partial differential

equation $E(\phi, u(\boldsymbol{r})) = 0$. Details of the Cea method are given in [12] and a summary of it is given in Appendix D. This approach rests on the idea that one can optimize a functional $J[u]$ under the constraint $E(u(\boldsymbol{r})) = 0$ by finding the stationary points of the Lagrangian functional $L = J + \int_\Omega \lambda(\boldsymbol{r})E(u)d\boldsymbol{r}$, where $\lambda(\boldsymbol{r})$ is a dual variable inside the integral. As described in [12], we consider the change, $\delta L_{\delta u}$, of L in response to the change, δu, in u and the change, $\delta L_{\delta\phi}$, of L in response to the change, $\delta\phi$, in ϕ, i.e. $\delta L_{\delta u} = L(\phi, u + \delta u) - L(\phi, u)$ and $\delta L_{\delta\phi} = L(\phi + \delta\phi, u) - L(\phi)$. Setting $\delta L_{\delta u} = \delta L_{\delta\lambda} = 0$ we obtain an adjoint problem for λ and an expression for $\delta L_{\delta\phi}$.

In sections 3.3.3-3.3.4, we use the method introduced by Cea, i.e we derive the functional derivatives in the weak form by treating the δu terms through the Lagrangian. In appendix B, we use Theorems 1-2 and we treat the terms corresponding to δu explicitly (strong form). We will also use theorem 3 to show the equivalence between the (weak form) expressions found in section 3.3 and the (strong form) expressions of appendix B.

3.3.3 Functional Level-Set Derivatives for the Hamiltonian H

Considering shape optimization in the context of self-consistent field theory iscomplicated by the fact that the functional H not only depends on ϕ but also on the fields $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$. We thus have:

$$\delta H = \int_\Omega [\frac{\delta H}{\delta\phi(\boldsymbol{r})}\delta\phi(\boldsymbol{r}) + \frac{\delta H}{\delta w_+(\boldsymbol{r})}\delta w_+(\boldsymbol{r}) + \frac{\delta H}{\delta w_-(\boldsymbol{r})}\delta w_-(\boldsymbol{r})]d\boldsymbol{r}.$$

Section 3.2 provides expressions for $\dfrac{\delta H}{\delta w_+(\boldsymbol{r})}$ and $\dfrac{\delta H}{\delta w_-(\boldsymbol{r})}$. In what follows, we denote

$$\delta H_{\delta\phi} = H[\phi + \delta\phi, w_+, w_-] - H[\phi, w_+, w_-] = \int_\Omega \frac{\delta H}{\delta\phi(\boldsymbol{r})}\delta\phi(\boldsymbol{r})d\boldsymbol{r}$$

and derive expressions for $\dfrac{\delta H}{\delta\phi(\boldsymbol{r})}$.

Using theorem 1 for the field terms in H and writing the $\delta \ln Q = \dfrac{1}{Q}\delta Q$ term in equation (3.1) for the Hamiltonian, we have:

$$\begin{aligned}
&\delta H_{\delta\phi} \\
&\quad = -\frac{1}{V}\int_\Gamma \Big(- w_+(\boldsymbol{r}) + \frac{w_-(\boldsymbol{r})^2}{\chi_{AB}}\Big)\delta\phi(\boldsymbol{r})\, d\Gamma \\
&\quad - \frac{1}{Q}\delta Q_{\delta\phi} - \frac{\delta V_{\delta\phi}}{V^2}\int_\Omega \Big(- w_+(\boldsymbol{r}) + \frac{w_-^2(\boldsymbol{r})}{\chi_{AB}}\Big)d\boldsymbol{r} - \frac{-\delta V_{\delta\phi}}{V^2}Q,
\end{aligned} \tag{3.5}$$

where the fields $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$ are kept constant. In the case of a polymeric material with a fixed volume and fixed amount of material this expression simplifies to:

$$\delta H_{\delta\phi} = -\frac{1}{V}\int_\Gamma \Big(- w_+(\boldsymbol{r}) + \frac{w_-(\boldsymbol{r})^2}{\chi_{AB}}\Big)\delta\phi(\boldsymbol{r})\, d\Gamma - \frac{1}{Q}\delta Q_{\delta\phi}. \tag{3.6}$$

The term $\delta Q_{\delta\phi}$ requires one to take into account the change in the solution to the Fokker-Planck equation and the associated boundary conditions, for which we use the method introduced by Cea [12] (see Appendix C). In the following derivations we are going to treat the term $\delta Q_{\delta\phi}$ only which can be reintroduced in (3.6), and we note that all the other terms do not have a dependence on $q(\boldsymbol{r}, s)$.

3.3.4 Case of a Robin Boundary Condition

We define the Lagrangian:

$$\begin{aligned} L &= \int_{\Omega\times[0,1]} q(s,\boldsymbol{r})\delta(s=1)\; ds\, d\Omega \\ &+ \int_{\Omega\times[0,1]} \lambda(s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) - \big[\Delta q(s,\boldsymbol{r}) - q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\big]\Big)\; ds\, d\Omega, \end{aligned}$$

where $\lambda(s,\boldsymbol{r})$ is a Lagrange multiplier to enforce the Fokker-Planck equation (3.2) for q in Ω. After integrating by part the Lagrangian can be rewritten as:

$$\begin{aligned} L &= \int_{\Omega\times[0,1]} q(s,\boldsymbol{r})\delta(s=1)\; ds\, d\Omega \\ &+ \int_{\Omega\times[0,1]} \lambda(s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\Big) + \nabla q(s,\boldsymbol{r})\cdot\nabla\lambda(s,\boldsymbol{r})\; ds\, d\Omega \\ &- \int_{\Gamma\times[0,1]} \lambda(s,\boldsymbol{r})\frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\; d\Gamma\, ds, \end{aligned}$$

and after inserting the Robin boundary conditions of q on Γ:

$$\begin{aligned} L &= \int_{\Omega\times[0,1]} q(s,\boldsymbol{r})\delta(s=1)\; ds\, d\Omega \\ &+ \int_{\Omega\times[0,1]} \lambda(s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\Big) + \nabla q(s,\boldsymbol{r})\cdot\nabla\lambda\; ds\, d\Omega \\ &+ \int_{\Gamma\times[0,1]} \lambda(s,\boldsymbol{r})\alpha(s) q(s,\boldsymbol{r})\; d\Gamma\, ds, \end{aligned}$$

- The change of L in response to a change in δq is:

$$\begin{aligned}\delta L_{\delta q} = &\int_{\Omega\times[0,1]} \delta q(s,\boldsymbol{r})\delta(s=1)\; ds\, d\Omega \\ &+ \int_{\Omega\times[0,1]} \lambda(s,\boldsymbol{r})\Big(\delta q_s(s,\boldsymbol{r}) + \delta q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\Big) + \nabla\delta q(s,\boldsymbol{r})\cdot\nabla\lambda(s,\boldsymbol{r})\; ds\, d\Omega \\ &+ \int_{\Gamma\times[0,1]} \lambda(s,\boldsymbol{r})\alpha(s)\delta q(s,\boldsymbol{r})\; d\Gamma\, ds,\end{aligned} \tag{3.7}$$

Integrating by parts in space, we obtain:

$$\int_\Omega \lambda\Delta\delta q\; d\Omega = \int_\Omega \Delta\lambda\delta q\; d\Omega + \int_\Gamma \lambda\frac{\partial\delta q}{\partial\boldsymbol{n}}\; d\Gamma - \int_\Gamma \delta q\frac{\partial\lambda}{\partial\boldsymbol{n}}\; d\Gamma \tag{3.8}$$

Likewise, integrating by parts in the chain contour variable gives:

$$\int_0^1 \lambda\delta q_s\; ds = -\int_0^1 \lambda_s\delta q\; ds + [\lambda\delta q]_0^1,$$

with

$$[\lambda\delta q]_{s=0}^{s=1} = \lambda(s=1)\delta q(s=1) - \lambda(s=0)\delta q(s=0),$$

where the second term drops out since $\delta q(s=0)=0$. Finally, with the change of variable $s = 1-t$, we have:

$$\int_0^1 \lambda\delta q_s ds = \int_0^1 \lambda_t\delta q\; dt + \lambda(s=1)\delta q(s=1)$$

$$\text{and} \qquad w(t) = w(1-s) = w^\dagger(s) \text{ and} \qquad \alpha(t) = \alpha(1-s) = \alpha^\dagger(s). \tag{3.9}$$

Combining equations (3.7)-(3.9), we have:

$$\begin{aligned}\delta L_{\delta q} = &\int_{\Omega\times[0,1]} \Big(1+\lambda(t,\boldsymbol{r})\Big)\delta(t=0)\delta q \; d\Omega\, dt \\ &+ \int_{\Omega\times[0,1]} \Big(\lambda_t(t,\boldsymbol{r}) - \Delta\lambda(t,\boldsymbol{r}) + \lambda(t,\boldsymbol{r})w^\dagger\Big)\delta q(s,\boldsymbol{r}) \; d\Omega\, dt \\ &+ \int_{\Gamma\times[0,1]} \lambda(t,\boldsymbol{r})\alpha^\dagger(s)\delta q(s,\boldsymbol{r}) + \delta q(s,\boldsymbol{r})\frac{\partial\lambda(t,\boldsymbol{r})}{\partial \boldsymbol{n}} \; d\Gamma\, dt,\end{aligned}$$

and following Cea, we require the change in q to be zero to obtain the following system of equations for λ:

$$\begin{cases} \lambda_t(t,\boldsymbol{r}) = \Delta\lambda(t,\boldsymbol{r}) - \lambda(t,\boldsymbol{r})\, w^\dagger(t,\boldsymbol{r}) \text{ in } \Omega. \\ \\ \lambda(t,\boldsymbol{r})\alpha^\dagger(t) + \dfrac{\partial\lambda(t,\boldsymbol{r})}{\partial\boldsymbol{n}} = 0 \text{ on } \Gamma. \\ \\ \lambda(t=0,\boldsymbol{r}) = -1 \text{ in } \Omega. \end{cases} \tag{3.10}$$

Also comparing with equation (3.2), we have $\lambda = -q^\dagger$.

- Using theorems 1 and 2, the change of L in response to a change in $\delta\phi$ is:

$$\begin{aligned}\delta L_{\delta\phi} = &-\int_{\Gamma\times T} q(s,\boldsymbol{r})\delta(s=1)\delta\phi \; ds\, d\Gamma \\ &- \int_{\Gamma\times T} \lambda(1-s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r}) \; ds\, d\Gamma \\ &- \int_{\Gamma\times T} \nabla q(s,\boldsymbol{r})\cdot\nabla\lambda(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r}) \; ds\, d\Gamma \\ &- \int_{\Gamma\times T} \kappa(\boldsymbol{r})\lambda(1-s,\boldsymbol{r})\alpha(s)q(s,\boldsymbol{r})\delta\phi(\boldsymbol{r}) \; d\Gamma\, ds \\ &- \int_{\Gamma\times T} \alpha(s)\partial_{\boldsymbol{n}}(\lambda(1-s,\boldsymbol{r})q(s,\boldsymbol{r}))\delta\phi(\boldsymbol{r}) \; d\Gamma\, ds\end{aligned}$$

and inserting $-q^\dagger$ instead of λ:

$$\begin{aligned}\delta L_{\delta\phi} = &- \int_{\Gamma\times T} q(s,\boldsymbol{r})\delta(s=1)\delta\phi \; ds\, d\Gamma \\ &+ \int_{\Gamma\times T} q^\dagger(1-s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r}) \; ds\, d\Gamma \\ &+ \int_{\Gamma\times T} \nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r}) \; ds\, d\Gamma \\ &+ \int_{\Gamma\times T} \alpha(s)(\kappa(\boldsymbol{r}) + \partial_{\boldsymbol{n}})(q^\dagger(1-s,\boldsymbol{r})q(s,\boldsymbol{r}))\delta\phi(\boldsymbol{r}) \; d\Gamma\, ds\end{aligned} \tag{3.11}$$

One can see that in the case of a Neumann boundary condition the last two terms drop out and that we recover the expression given by equation (A.6) in the appendix. Likewise, in the case of a Dirichlet boundary condition the two last terms drop out and using theorem 3 in equation (3.11) along with the p.d.e for q, one recovers equation (A.5) of the appendix.

3.4 Examples of Shape Optimization using the Level-Set Derivatives

In the case of free surface polymers, the self-assembly process is described by the saddle point of the Hamiltonian, H, with respect to the fields $w_-(\boldsymbol{r})$ and $w_+(\boldsymbol{r})$ as well as with respect to the shape, here encoded by the level-set function $\phi(\boldsymbol{r})$. In this section, we provide a few examples to illustrate potential applications of the shape derivatives derived in this manuscript. In particular, we are focusing on illustrating that the energy of the systems considered decreases in the case where the domain's boundary is taken

as one of the free parameters of the optimization process. These examples show how a polymeric material with a free surface will change its shape in order to decrease its energy and transit to a locally stable state with a lower energy. The examples in two spatial dimensions could be useful in the context of directed self-assembly and the positioning of channels in integrated circuits or in the study of self-assembly in free surface droplets; the example in three spatial dimensions could be relevant to the study of free surface polymeric melts such as cubosomes [84, 83, 85, 32, 46, 70, 63].

Algorithm 5 describes the numerical steps we use to minimize $H[w_-(\boldsymbol{r}), w_+(\boldsymbol{r}), \phi(\boldsymbol{r})]$. The optimization with respect to the fields is obtained by solving the standard SCFT equations of section 3.2 (see Algorithm 6). In our work, we use the framework introduced in Ouaknin *et al.* [77], which imposes a Neumann boundary condition for q, as first proposed in [21], using the method of Papac *et al.* [79, 80] and the parallel framework of Mirzadeh *et al.* [68]. The optimization with respect to the shape drives the boundary Γ to a local saddle point. In our work, the boundary motion is described by the level-set equation:

$$\frac{\partial \phi(\boldsymbol{r})}{\partial \tau} + v_n(\boldsymbol{r})|\nabla \phi(\boldsymbol{r})| = 0, \tag{3.12}$$

in pseudo-time τ, where $v_n(\boldsymbol{r})$ is the normal velocity at Γ. We use the functional level-set derivatives derived in section 3.3 and choose $\delta\phi(\boldsymbol{r}) = -\dfrac{\delta H}{\delta \phi(\boldsymbol{r})}$ so that the total variation of the free energy is decreased: $\delta H = -\displaystyle\int_\Gamma \left|\frac{\delta H}{\delta \phi(\boldsymbol{r})}\right|^2 d\Gamma < 0$. We thus define $v_n(\boldsymbol{r})$ to be proportional to $-\delta\phi(\boldsymbol{r})$. Specifically, the velocity is computed inside Ω and is

extrapolated in a narrow band outside the domain by solving:

$$\frac{\partial v_n(\boldsymbol{r})}{\partial t} + \mathcal{H}\Big(\phi(\boldsymbol{r})\Big)\nabla v_n(\boldsymbol{r}) \cdot \nabla\phi(\boldsymbol{r}) = 0, \tag{3.13}$$

in order to have a well-defined normal velocity at the vicinity of the zero level set of ϕ. A similar equation is also used to extrapolate the statistical fields w_+ and w_-, which are needed to compute the Fokker-Planck equation in the advected domain $\phi^{\tau+1}(\boldsymbol{r}) < 0$. Finally the level-set function is transformed into a signed distance function [71, 86, 67]:

$$\frac{\partial\phi(\boldsymbol{r})}{\partial\tau} + \text{signum}\Big(\phi(\boldsymbol{r})\Big)\Big(|\nabla\phi(\boldsymbol{r})| - 1\Big) = 0, \tag{3.14}$$

which is desired for the numerical algorithms used in this paper. To regularize the shape, we add to the normal velocity a surface tension term $\gamma\,\kappa$, where $\gamma > 0$ is a chosen parameter, and we discretize this term as in [73]. In all of the examples presented below, the volume and the amount of material are kept constant by subtracting the average velocity from the velocity obtained by computing the shape derivative.

Algorithm 5 Procedure to find the saddle point of H with optimal shape

I. Perform algorithm 6

II.

while $||\delta H/\delta\phi||_2 > \epsilon_\phi$ **do** // ϵ_ϕ `is a chosen tolerance.`

1. Perform 1-4 steps of algorithm 6

2. Find the velocity for $\phi(\boldsymbol{r})$ and advect $\phi(\boldsymbol{r})$ every $O(100)$ mean-field steps

3. $t = t + 1$

III. Perform algorithm 6

Algorithm 6 Procedure to find the saddle point of H with respect to the fields w_- and w_+ (SCFT).

while $||\delta H/\delta\omega||_2 > \epsilon_w$ **do** // ϵ_w `is a chosen tolerance.`

1. Solve two Fokker Planck equations $q(s, \boldsymbol{r}), q^\dagger(s, \boldsymbol{r})$
2. compute the densities $\rho_A(\boldsymbol{r}), \rho_B(\boldsymbol{r})$
3. compute the force for $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$
4. advance the potentials $w_-(\boldsymbol{r}), w_+(\boldsymbol{r})$
5. $t = t + 1$

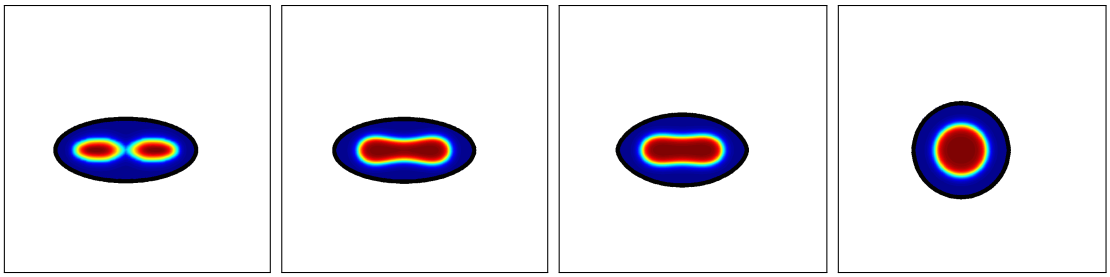
In all of the examples, we first only evolve the fields for $O(1000)$ mean-field steps until we reach a local saddle point. Then we evolve the fields for another O(10,000) mean-field steps, but this time in tandem with the optimization with respect to ϕ, which we evolve every $O(100)$ mean-field steps. Finally, we fix the boundary and relax the fields for another $O(1000)$ mean-field steps until we reach a new saddle point. This procedure ensures that, by changing the shape, we decrease the energy. Figures 3.3, 3.4, 3.5 and 3.6 depicts the decrease in energy during each of the three steps, confirming that the shape derivatives we developed can be used to optimize the coupled fields and shape in both two and three spatial dimensions. We define $\Delta H_{\Delta\phi}$ as the difference of energy of the polymer between its initial shape and its optimized shape, which corresponds to the end of phase I and the end of phase III of algorithm 5, respectively.

As it is always the case in optimization problems that may have multiple local optimal points, the initial data (here the seed in the fields, which in turn define the polymer

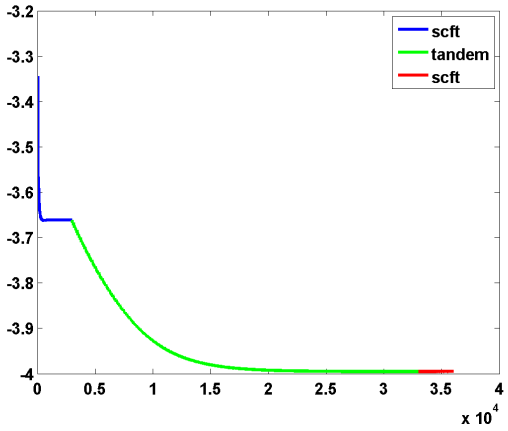
densities) influences which local saddle point (or metastable equilibrium) is reached. Figures 3.3 and 3.4 present the results of the shape optimization process obtained with two different seeds, which lead to significant differences in both the topology of the densities as well as in the shape of the free boundary.

Figure 3.5 provides an example where the seed consists of an random distribution in the statistical fields inside a circular free boundary. The optimization process leads to a symmetric density morphology. Interestingly, the minority region wets the free surface and locally changes the curvature in order to reduce the AB interfacial region.

Figure 3.6 provides an numerical example in three spatial dimensions that is motivated by the study of cubosomes, which are cubic bicontinuous network morphologies that have gained a lot of attention recently particularly because of their potential in food sciences and as drug delivery carriers [46, 70, 63]. Starting with a gyroid seed for the minority density enclosed into a cubic free surface, the self-assembly process drives the morphology of both the minority density and the free surface to a nontrivial geometry. In this case, the energy drop is $\Delta H_{\Delta\phi} = -0.0115$, illustrating that complex three dimensional configurations can be treated by our formalism.

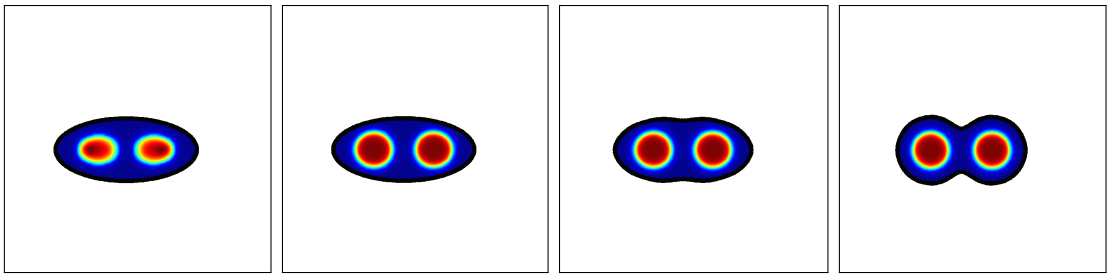


(a) t=0 (b) t=3,000 (c) t=6,000 (d) t=36,000

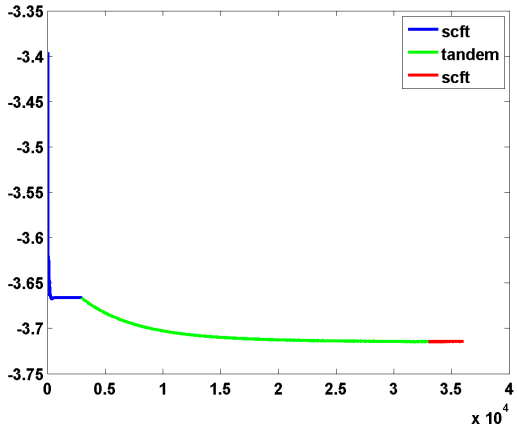


(e) Energy

Figure 3.3: Evolution of the densities and the shape: starting from a free surface with elliptical shape and a minority region enclosed into two ellipses, both the boundary and the minority region transit to a disk shape. One can also observe a change in topology in the minority region. The change in shape leads to an energy drop of $\Delta H_{\Delta\phi} = -0.334$. The parameters for this simulation are $(f_A, \chi_{AB}) = (0.3, 36)$, $L_x = 12.8R_g$, $\gamma = 0$ and $\Delta s = 0.01$.

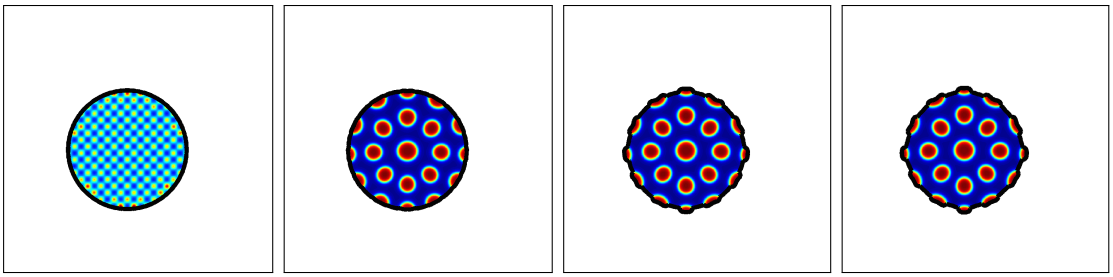


(a) t=0 (b) t=3,000 (c) t=6,000 (d) t=36,000

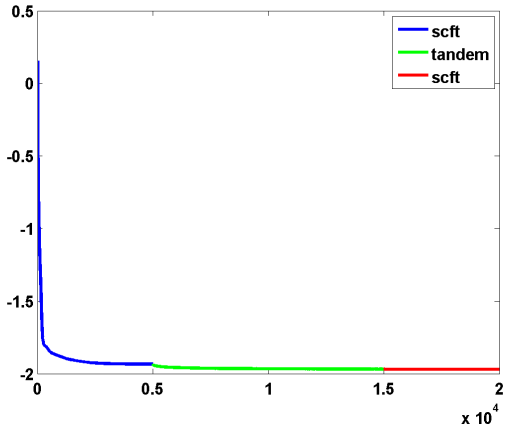


(e) Energy

Figure 3.4: Same example as in figure 3.3, but this time the seeded minority components are located farther apart. Here, the energy drop is $\Delta H_{\Delta\phi} = -0.0487$. The parameters for this simulation are $(f_A, \chi_{AB}) = (0.3, 36)$, $L_x = 12.8R_g$, $\gamma = 0$ and $\Delta s = 0.01$.

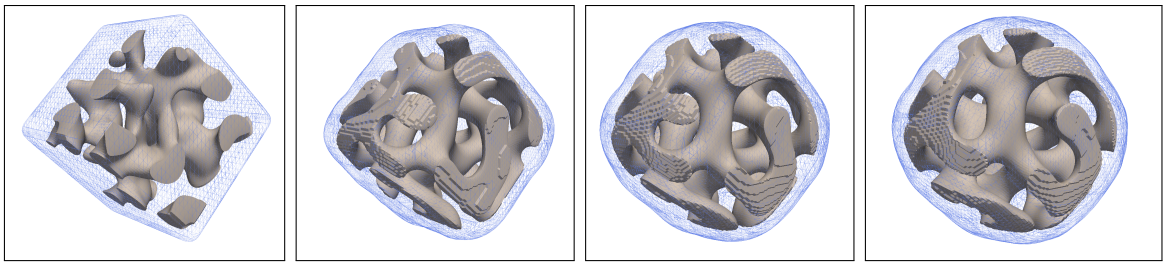


(a) t=0 (b) t=5000 (c) t=10,000 (d) t=20,000

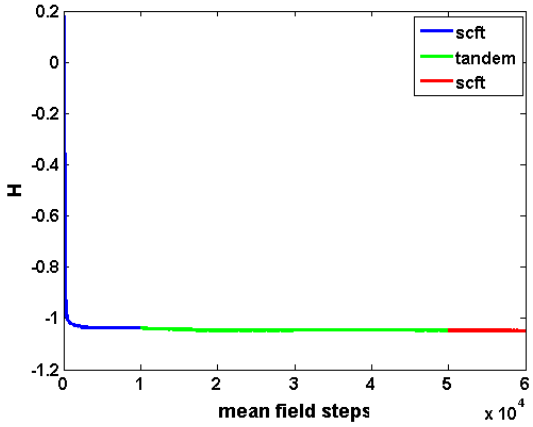


(e) Energy

Figure 3.5: Evolution of the densities and the free boundary: starting from a random density distribution enclosed by a circular free surface, the minority region self-assembles into a circular morphology. The energy decreases by $\Delta H_{\Delta\phi} = -0.0357$. The parameters for this simulation are $(f_A, \chi_{AB}) = (0.3, 25)$, $\gamma = 0.5/R_g$ and $\Delta s = 0.005$.



(a) t=0 (b) t=20,000 (c) t=40,000 (d) t=60,000



(e) Energy

Figure 3.6: Evolution of the density and the shape: starting with a cube for the shape and a gyroid seed for the A component. The energy drop is $\Delta H_{\Delta\phi} = -0.0115$. The parameters are $(f_A, \chi_{AB}) = (0.36, 20)$, $\gamma = 1.00/R_g$, $L_x = 18R_g$, and $\Delta s = 0.005$.

3.5 Conclusion

We derived functional level-set derivatives in strong and weak forms for in the cases of homogeneous Dirichlet, Neumann and Robin boundary conditions and in the context of polymer self-consistent field theory. Using these shape derivatives in the Neumann

case, we provided numerical examples of free boundary SCFT simulations for confined diblock copolymer melts in two and three spatial dimensions. The minimization process takes into account the shape of the free boundary in addition to the statistical fields, which can enable studies of their coupling. In all the cases studied, the minimization leads to locally stable states with decreased energy. This work could be used for the exploration of the free surface assembly of a wide range of polymeric materials.

Appendix A

Derivation in weak form for Dirichlet and Neumann

For convenience we derive here in detail for Dirichlet and Neumann as special cases.

A.1 Case of a Dirichlet Boundary Condition

We define the Lagrangian:

$$\begin{aligned} L &= \int_{\Omega\times[0,1]} q(s,\boldsymbol{r})\delta(s=1)\; ds\, d\Omega \\ &+ \int_{\Omega\times[0,1]} \lambda(s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) - \big[\Delta q(s,\boldsymbol{r}) - q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\big]\Big)\; ds\, d\Omega \\ &+ \int_{\Gamma\times[0,1]} \mu(s,\boldsymbol{r}) q(s,\boldsymbol{r})\; d\Gamma\, ds, \end{aligned}$$

where $\lambda(s, \boldsymbol{r})$ is a Lagrange multiplier to enforce the Fokker-Planck equation (3.2) for q in Ω and $\mu(s, \boldsymbol{r})$ is a Lagrange multiplier to enforce the Dirichlet boundary condition $q = 0$ on the boundary Γ.

- the change of L in the direction of δq:

$$\begin{aligned}\delta L_{\delta q} = &\int_{\Omega\times[0,1]} \delta q(s,\boldsymbol{r})\delta(s=1)\; ds\, d\Omega \\ &+ \int_{\Omega\times[0,1]} \lambda(s,\boldsymbol{r})\Big(\delta q_s(s,\boldsymbol{r}) - \big[\Delta\delta q(s,\boldsymbol{r}) - \delta q(s,\boldsymbol{r})\, w(s,\boldsymbol{r})\big]\Big)\; ds\, d\Omega \\ &+ \int_{\Gamma\times[0,1]} \mu(s,\boldsymbol{r})\delta q(s,\boldsymbol{r})\; d\Gamma\, ds\end{aligned} \tag{A.1}$$

  Integrating by parts in space, we obtain:

$$\int_\Omega \lambda(s,\boldsymbol{r})\Delta\delta q\; d\Omega = \int_\Omega \Delta\lambda\delta q\; d\Omega + \int_\Gamma \lambda(s,\boldsymbol{r})\frac{\partial\delta q}{\partial \boldsymbol{n}}\; d\Gamma - \int_\Gamma \delta q\frac{\partial\lambda}{\partial \boldsymbol{n}}\; d\Gamma \tag{A.2}$$

  Likewise, integrating by parts in time gives:

$$\int_0^1 \lambda(s,\boldsymbol{r})\delta q_s\; ds = -\int_0^1 \lambda_s\delta q\; ds + [\lambda\delta q]_0^1,$$

  with

$$[\lambda q]_{s=0}^{s=1} = \lambda(s=1)\delta q(s=1) - \lambda(s=0)\delta q(s=0),$$

  where the second term drops out since $\delta q(s=0) = 0$. Finally, with the change of variable $s = 1 - t$, we have:

$$\int_0^1 \lambda\delta q_s ds = \int_0^1 \lambda_t\delta q\; dt + \lambda(s=1)\delta q(s=1) \qquad \text{and} \qquad w(t) = w(1-s) = w^\dagger(s). \tag{A.3}$$

Combining equations (A.1)-(A.3), we have:

$$\delta L_{\delta q} = \int_{\Omega\times[0,1]} \Big(1+\lambda\Big)\delta(t=0)\delta q + \Big(\lambda_t - \Delta\lambda + \lambda w^\dagger\Big)\delta q(s,\boldsymbol{r})\; d\Omega\, dt$$
$$+ \int_{\Gamma\times[0,1]} \mu\delta q(s,\boldsymbol{r}) + \lambda\frac{\partial\delta q}{\partial \boldsymbol{n}} - \delta q\frac{\partial\lambda}{\partial\boldsymbol{n}}\; d\Gamma\, dt,$$

and thus we obtain the following system of equations for λ:

$$\begin{cases} \lambda_t(t,\boldsymbol{r}) = \Delta\lambda(t,\boldsymbol{r}) - \lambda(t,\boldsymbol{r})\, w^\dagger(t,\boldsymbol{r}) \text{ in } \Omega. \\ \\ \lambda(t,\boldsymbol{r}) = 0 \text{ on } \Gamma. \\ \\ \lambda(t=0,\boldsymbol{r}) = -1 \text{ in } \Omega. \end{cases} \tag{A.4}$$

as well as the relationship between λ and μ on Γ: $\mu\big|_\Gamma = -\dfrac{\partial\lambda}{\partial\boldsymbol{n}}\Big|_\Gamma$. Also comparing with equation (3.2), we have $\lambda = -q^\dagger$.

- the change of L in the direction of $\delta\phi$:

$$\delta L_{\delta\phi} = -\int_{\Gamma\times T} q(s,\boldsymbol{r})\delta(s=1)\delta\phi\; ds\, d\Gamma$$
$$-\int_{\Gamma\times T} \lambda(1-s,\boldsymbol{r})\underbrace{\Big(q_s(s,\boldsymbol{r}) - \big[\Delta q(s,\boldsymbol{r}) - q(s,\boldsymbol{r})\, w(\boldsymbol{r})\big]\Big)}_{=0 \text{ by (3.2)}}\delta\phi\; ds\, d\Gamma$$
$$-\int_{\Gamma\times T} \frac{\partial\mu q(s,\boldsymbol{r})}{\partial\boldsymbol{n}}\delta\phi\; d\Gamma\, ds$$
$$-\int_{\Gamma\times T} \kappa(\boldsymbol{r})\mu \underbrace{q(s,\boldsymbol{r})}_{=0 \text{ (Dirichlet)}} \delta\phi\; d\Gamma\, ds$$

Using $\mu\big|_\Gamma = -\dfrac{\partial\lambda}{\partial \boldsymbol{n}}\Big|_\Gamma$ and $\lambda = -q^\dagger$, we obtain:

$$\begin{aligned}\int_{\Gamma\times T} \frac{\partial\mu(s,\boldsymbol{r})q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\,\delta\phi & \\ &= \int_{\Gamma\times T}\Big(\mu\frac{\partial q}{\partial \boldsymbol{n}} + \underbrace{q(s,\boldsymbol{r})}_{=0\text{ (Dirichlet)}}\frac{\partial\mu}{\partial \boldsymbol{n}}q\Big)\,\delta\phi\;d\Gamma\,ds \\ &= \int_{\Gamma\times T}\frac{\partial q^\dagger(1-s,\boldsymbol{r})}{\partial \boldsymbol{n}}\frac{\partial q}{\partial \boldsymbol{n}}\,\delta\phi\;d\Gamma\,ds\end{aligned}$$

and thus:

$$\delta L\cdot\delta\phi = -\int_{\Gamma\times T}\Big(q(s,\boldsymbol{r})\delta(s=1) + \frac{\partial q^\dagger(1-s,\boldsymbol{r})}{\partial \boldsymbol{n}}\frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\Big)\delta\phi(\boldsymbol{r})\;d\Gamma\,ds. \quad \text{(A.5)}$$

A.2 Case of a Neumann Boundary Condition

As in the Dirichlet case, we introduce the Lagrangian:

$$L = \int_{\Omega\times T} q(s,\boldsymbol{r})\delta(s=1)ds d\Omega + \int_{\Omega\times T}\lambda(s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) - \big[\Delta q(s,\boldsymbol{r}) - q(s,\boldsymbol{r})\,w(s,\boldsymbol{r})\big]\Big)\;ds\,d\Omega.$$

After integration by parts and taking into account the homogeneous Neumann boundary condition on q, we get:

$$\begin{aligned}L & \\ &= \int_{\Omega\times T} q\delta(s=1)ds d\Omega \\ &+ \int_{\Omega\times T}\lambda(s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})\,w(s,\boldsymbol{r})\Big) + \nabla q(s,\boldsymbol{r})\cdot\nabla\lambda(s,\boldsymbol{r})\;ds\,d\Omega.\end{aligned}$$

- the change of L in the direction of δq:

$$\begin{aligned}
&\delta L_{\delta q} \\
&\quad = \int_{\Omega\times T} \delta q(s,\boldsymbol{r})\delta(s=1)ds d\Omega \\
&\quad + \int_{\Omega\times T} \lambda(s,\boldsymbol{r})\Big(\delta q_s(s,\boldsymbol{r}) + \delta q(s,\boldsymbol{r})\, w(\boldsymbol{r})\Big) + \nabla\delta q(s,\boldsymbol{r})\cdot\nabla\lambda(s,\boldsymbol{r})\; ds\, d\Omega.
\end{aligned}$$

Integrating by parts in space, we have:

$$\int_\Omega \nabla\delta q(s,\boldsymbol{r})\cdot\nabla\lambda(s,\boldsymbol{r})d\Omega = -\int_\Omega \Delta\lambda(s,\boldsymbol{r})\delta q(s,\boldsymbol{r})d\Omega + \int_\Gamma \delta q(s,\boldsymbol{r})\frac{\partial\lambda(s,\boldsymbol{r})}{\partial\boldsymbol{n}}d\Gamma.$$

Integrate by parts in time and combining the integrals on Ω and Γ as in the Dirichlet, we obtain the following systems of equation for λ:

$$\begin{cases}
\lambda_t(t,\boldsymbol{r}) = \Delta\lambda(t,\boldsymbol{r}) - \lambda(t,\boldsymbol{r})\, w^\dagger(t,\boldsymbol{r}) \text{ in } \Omega. \\
\dfrac{\partial\lambda(t,\boldsymbol{r})}{\partial\boldsymbol{n}} = 0 \text{ on } \Gamma. \\
\lambda(t=0) = -1.
\end{cases}$$

- the change of L in the direction of $\delta\phi$: substituting $\lambda(\boldsymbol{r},s) = -q^\dagger(\boldsymbol{r},s)$ in the Lagrangian and using theorem 1, we obtain:

$$\begin{aligned}
\delta L_{\delta\phi} = &-\int_{\Gamma\times T} q\delta(s=1)\delta\phi(\boldsymbol{r})\; ds\, d\Gamma \\
&+ \int_{\Gamma\times T} q^\dagger(1-s,\boldsymbol{r})\Big(q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})w(\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r})\; ds\, d\Gamma \qquad \text{(A.6)} \\
&+ \int_{\Gamma\times T} \nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})\; ds\, d\Gamma,
\end{aligned}$$

which is the change of L with respect to the level-set.

Remarks:

1. In the case of homogeneous Neumann boundary conditions, we have $\nabla q \cdot \nabla q^\dagger = \dfrac{\partial q}{\partial \boldsymbol{\tau}} \dfrac{\partial q^\dagger}{\partial \boldsymbol{\tau}}$, where $\boldsymbol{\tau}$ is the tangential vector to the boundary Γ.

2. We can use the factorization property of the partition function to write a more transparent expression for the shape derivative: $Q = \dfrac{1}{V} \displaystyle\int_\Omega q(\boldsymbol{r}, s) q^\dagger(\boldsymbol{r}, 1-s)\, d\boldsymbol{r}$ for any s. Thus choosing $s = 1$ gives $Q = \dfrac{1}{V} \displaystyle\int_\Omega q(\boldsymbol{r}, s=1) \underbrace{q^\dagger(\boldsymbol{r}, s=0)}_{=1}\, d\boldsymbol{r}$ and choosing $s = 0$ gives $Q = \dfrac{1}{V} \displaystyle\int_\Omega \underbrace{q(\boldsymbol{r}, s=0)}_{=1} q^\dagger(\boldsymbol{r}, s=1)\, d\boldsymbol{r}$. Thus one can write $Q = \dfrac{1}{2V} \displaystyle\int_\Omega q(\boldsymbol{r}, s=1) + q^\dagger(\boldsymbol{r}, s=1)\, d\boldsymbol{r}$, and by interchanging q and $q^\dagger$ in equation (A.6), one can write the following:

$$
\begin{aligned}
\delta L_{\delta\phi} = &- \int_{\Gamma\times T} q\delta(s=1)\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&+ \int_{\Gamma\times T} q^\dagger(1-s,\boldsymbol{r})\big[q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})w(\boldsymbol{r})\big]\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&+ \int_{\Gamma\times T} \nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&= \int_{\Gamma\times T} \Big(- q(s,\boldsymbol{r})\delta(s=1) + q^\dagger(1-s,\boldsymbol{r})\Delta q(s,\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&+ \int_{\Gamma\times T} \nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&= \int_{\Gamma\times T} -\frac{1}{2}\Big[q(s,\boldsymbol{r})\delta(s=1) + q^\dagger(1-s,\boldsymbol{r})\delta(s=0)\Big]\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&+ \int_{\Gamma\times T} \Big(\frac{1}{2}q^\dagger(1-s,\boldsymbol{r})\Delta q(s,\boldsymbol{r}) + \frac{1}{2}q(s,\boldsymbol{r})\Delta q^\dagger(1-s,\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&+ \int_{\Gamma\times T} \nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&= \frac{1}{2}\int_{\Gamma\times T} -\Big[q(s,\boldsymbol{r})\delta(s=1) - q^\dagger(1-s,\boldsymbol{r})\delta(s=0)\Big]\delta\phi(\boldsymbol{r})\ ds\, d\Gamma \\
&+ \frac{1}{2}\int_{\Gamma\times T} \Delta\Big[q(s,\boldsymbol{r})q^\dagger(1-s,\boldsymbol{r})\Big]\delta\phi(\boldsymbol{r})\ ds\, d\Gamma.
\end{aligned}
\tag{A.7}
$$

Using equation (3.4) we obtain the following symmetric expression:

$$
\delta L_{\delta\phi} = -\frac{1}{2}\int_{\Gamma}\Big(q(1,\boldsymbol{r}) + q^\dagger(1,\boldsymbol{r}) - Q\Delta f_+(\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r})\, d\Gamma. \tag{A.8}
$$

Appendix B

Derivation in the Strong Form

We provide here the derivation in the strong form[1] for completeness.

Using theorem 1 for Q:

$$\delta Q_{\delta\phi} = -\frac{1}{V}\int_{\Gamma} q(s=1,\boldsymbol{r})\delta\phi(\boldsymbol{r}) + \frac{1}{V}\int_{\Omega}\delta q(s=1,\boldsymbol{r})\,d\boldsymbol{r}. \tag{B.1}$$

We now define $\delta Q_{stress} = \frac{1}{V}\int_{\Omega}\delta q(s=1,\boldsymbol{r})\,d\boldsymbol{r}$ and deriving δQ_{stress} with respect to $\delta\phi(\boldsymbol{y})$, we obtain:

$$\frac{\delta Q_{stress}}{\delta\phi(\boldsymbol{y})} = \frac{1}{V}\int_{\Omega}\frac{\delta q(s=1,\boldsymbol{r})}{\delta\phi(\boldsymbol{y})}\,d\boldsymbol{r}. \tag{B.2}$$

Define $g(s,\boldsymbol{r},\boldsymbol{y}) = \frac{\delta q(s,\boldsymbol{r})}{\delta\phi(\boldsymbol{y})}$ with $g(s=0)=0$ as initial condition. We write a PDE for g when w is kept constant but we move the domain:

$$\frac{\delta q_s(s,\boldsymbol{r},\boldsymbol{y})}{\delta\phi(\boldsymbol{y})} = \frac{\delta\Delta q(s,\boldsymbol{r},\boldsymbol{y})}{\delta\phi(\boldsymbol{y})} - \frac{\delta q(s,\boldsymbol{r},\boldsymbol{y})}{\delta\phi(\boldsymbol{y})}w(s,\boldsymbol{r}) - \frac{\delta w(s,\boldsymbol{r})}{\delta\phi(\boldsymbol{y})}q(s,\boldsymbol{r},\boldsymbol{y}) \tag{B.3}$$

[1] Not to be confused with the material derivative [88].

Interchanging δ with the s derivative and the laplacian and using that w is kept constant the following PDE is obtained for g:

$$g_s(s,\boldsymbol{r},\boldsymbol{y}) = \Delta g(s,\boldsymbol{r},\boldsymbol{y}) - g(s,\boldsymbol{r},\boldsymbol{y})w(s,\boldsymbol{r}) \tag{B.4}$$

$$\begin{aligned}
\frac{\delta Q_{stress}(s)}{\delta\phi(\boldsymbol{y})} = & \\
& \frac{1}{V}\int_{\Omega} g(s,\boldsymbol{r},\boldsymbol{y})d\boldsymbol{r} = \\
& \frac{1}{V}\int_{\Omega} \underbrace{q^{\dagger}(s=0,\boldsymbol{r})}_{=1} g(s,\boldsymbol{r},\boldsymbol{y}) - \underbrace{g(s=0,\boldsymbol{r},\boldsymbol{y})}_{=0} q^{\dagger}(s,\boldsymbol{r})d\boldsymbol{r} = \\
& \frac{1}{V}\int_{\Omega\times T} (q^{\dagger}(s-t,\boldsymbol{r})g(t,\boldsymbol{r},\boldsymbol{y}))_t dtd\boldsymbol{r} = \\
& \frac{1}{V}\int_{\Omega\times T} g_t(t,\boldsymbol{r},\boldsymbol{y})q^{\dagger}(s-t,\boldsymbol{r}) - g(t,\boldsymbol{r},\boldsymbol{y})q^{\dagger}_t(s-t,\boldsymbol{r})dtd\boldsymbol{r} = \\
& \frac{1}{V}\int_{\Omega\times T} (\Delta g(t,\boldsymbol{r},\boldsymbol{y}) - g(t,\boldsymbol{r},\boldsymbol{y})\, w(\boldsymbol{r}))q^{\dagger}(s-t,\boldsymbol{r})dtd\boldsymbol{r} \\
& -\frac{1}{V}\int_{\Omega\times T} g(s,\boldsymbol{r},\boldsymbol{y})(\Delta q^{\dagger}(s-t,\boldsymbol{r}) - q^{\dagger}(s-t,\boldsymbol{r})\, w(\boldsymbol{r}))dtd\boldsymbol{r} = \\
& \frac{1}{V}\int_{\Omega\times T} \Delta g(t,\boldsymbol{r},\boldsymbol{y})q^{\dagger}(s-t,\boldsymbol{r}) - g(t,\boldsymbol{r},\boldsymbol{y})\Delta q^{\dagger}(s-t,\boldsymbol{r})d\sigma d\boldsymbol{r} = \\
& \frac{1}{V}\int_{\Gamma\times T} \underbrace{q^{\dagger}(s-t,\boldsymbol{r})}_{=0 \text{ for Dirichlet}} \nabla g(t,\boldsymbol{r},\boldsymbol{y})\cdot\nabla\phi(\boldsymbol{r})dtd\Gamma \\
& -\frac{1}{V}\int_{\Gamma\times T} g(t,\boldsymbol{r},\boldsymbol{y}) \underbrace{\nabla q^{\dagger}(s-t,\boldsymbol{r})\cdot\nabla\phi(\boldsymbol{r})}_{=0 \text{ for Neumann}} dtd\Gamma,
\end{aligned} \tag{B.5}$$

where $\boldsymbol{n} = \frac{\nabla\phi(\boldsymbol{r})}{|\nabla\phi(\boldsymbol{r})|}$ and we assume $\phi(\boldsymbol{r})$ is a signed distance function, i.e $|\nabla\phi(\boldsymbol{r})| = 1$.

B.1 Case of a Dirichlet Boundary Condition

Introducing Dirichlet boundary conditions $q(s, \boldsymbol{r}_\Gamma(\phi), \phi) = 0$, where $\boldsymbol{r}_\Gamma$ describes the position of the interface Γ,and take the derivative with respect to $\delta\phi(\boldsymbol{y})$:

$$\frac{\delta q(s, \boldsymbol{r}_\Gamma(\phi), \phi)}{\delta\phi(\boldsymbol{y})} + \nabla q(s, \boldsymbol{r}_\Gamma(\phi), \phi) \cdot \frac{\delta \boldsymbol{r}_\Gamma}{\delta\phi(\boldsymbol{y})} = 0 \tag{B.6}$$

Inserting the Dirichlet boundary conditions in (B.5) , i.e $q^\dagger(\boldsymbol{r}, s-t) = 0$, and using $\frac{\delta \boldsymbol{r}_\Gamma(\boldsymbol{\phi})}{\delta\phi(\boldsymbol{y})} = -\nabla\phi(\boldsymbol{r})\delta(\boldsymbol{r}_\Gamma - \boldsymbol{y})$ for a reinitialyzed $\phi(\boldsymbol{r})$ we obtain:

$$\delta Q_{stress}(s) = \frac{1}{V}\int_{\Gamma\times T} -\frac{\partial q(t,\boldsymbol{r})}{\partial \boldsymbol{n}}\frac{\partial q^\dagger(s-t,\boldsymbol{r})}{\partial \boldsymbol{n}}\delta\phi(\boldsymbol{r})dtd\boldsymbol{r} \tag{B.7}$$

We note the equivalence between equation (A.5) and equation (B.7).

B.2 Case of a Homogeneous Neumann Boundary Condition

We write the Neumann boundary conditions as $f(\boldsymbol{r}_\Gamma(\phi), \phi) = \nabla q(\boldsymbol{r}_\Gamma) \cdot \nabla\phi(\boldsymbol{r}) = 0$:

$$\frac{\delta f(\boldsymbol{r}_\Gamma(\phi), \phi)}{\delta\phi(\boldsymbol{y})} = \frac{\delta\nabla q(\boldsymbol{r}_\Gamma)}{\delta\phi(\boldsymbol{y})} \cdot \nabla\phi(\boldsymbol{r}_\Gamma)\delta\phi(\boldsymbol{r}_\Gamma) + \nabla q(\boldsymbol{r}_\Gamma)\frac{\delta\nabla\phi(\boldsymbol{r}_\Gamma)}{\delta\phi(\boldsymbol{y})} + \nabla(\nabla q \cdot \nabla\phi(\boldsymbol{r}_\Gamma))\frac{\delta \boldsymbol{r}_\Gamma}{\delta\phi(\boldsymbol{y})} \tag{B.8}$$

Inserting the boundary conditions as in the Dirichlet case into (B.5) we get:

$$\delta Q_{stress}(s) = \frac{1}{V}\int_{\Gamma\times T} q^{\dagger}(s-t,\boldsymbol{r})\left(\frac{\partial^2 q(t,\boldsymbol{r})}{\partial \boldsymbol{n}^2}\delta\phi(\boldsymbol{r}) - \nabla q(t,\boldsymbol{r}\right)\cdot\nabla\delta\phi(\boldsymbol{r})\,dtd\boldsymbol{r} \tag{B.9}$$

Now, using theorems 3 and 4 (theorem 2.67 of Sokolowski and Zolesio in [88]) it can be decomposed into terms of $\delta\phi$ and terms of $\nabla\delta\phi$ by taking $f = q^{\dagger}\delta\phi$ and $g = q$:

$$\begin{aligned}
&\int_{\Gamma\times T} q^{\dagger}(1-s,\boldsymbol{r})\left(\frac{\partial^2 q(s,\boldsymbol{r})}{\partial \boldsymbol{n}^2}\delta\phi(\boldsymbol{r}) - \nabla q(s,\boldsymbol{r})\cdot\nabla\delta\phi(\boldsymbol{r})\right)\,d\Gamma ds \\
&\qquad = \int_{\Gamma\times T} q^{\dagger}(1-s,\boldsymbol{r})\Delta q(\boldsymbol{r},s)\delta\phi(\boldsymbol{r}) + \nabla q(s,\boldsymbol{r})\cdot\nabla q^{\dagger}(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r}) \\
&\qquad - \frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\frac{\partial q^{\dagger}(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})}{\partial \boldsymbol{n}} - \kappa q^{\dagger}(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})\frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\;d\Gamma\,ds \\
&\qquad = \int_{\Gamma\times T} q^{\dagger}(1-s,\boldsymbol{r})\Delta q(\boldsymbol{r},s)\delta\phi(\boldsymbol{r}) + \nabla q(s,\boldsymbol{r})\cdot\nabla q^{\dagger}(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})\;d\Gamma\,ds \\
&\qquad = \int_{\Gamma\times T}\Big(q^{\dagger}(1-s,\boldsymbol{r})\big[q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})w(\boldsymbol{r})\big]\Big)\delta\phi(\boldsymbol{r})\;d\Gamma\,ds, \\
&\qquad + \int_{\Gamma\times T}\Big(\nabla q(s,\boldsymbol{r})\cdot\nabla q^{\dagger}(1-s,\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r})\;d\Gamma\,ds,
\end{aligned} \tag{B.10}$$

where in the first equality we use integration by parts on the surface, in the second equality we use the fact that $\dfrac{\partial q}{\partial \boldsymbol{n}} = 0$ and $\dfrac{\partial(q^{\dagger}\delta\phi)}{\partial \boldsymbol{n}} = 0$ and in the third we use the PDE. Again we note the equivalence between equation (A.6) and equation (B.9).

B.3 Case of a Homogeneous Robin Boundary Condition

We write the Robin boundary conditions as $f(\boldsymbol{r}_\Gamma(\phi),\phi) = \nabla q(\boldsymbol{r}_\Gamma)\cdot\nabla\phi(\boldsymbol{r})+\alpha(s)q(\boldsymbol{r}_\Gamma) = 0$:

$$\frac{\delta f(\boldsymbol{r}_\Gamma(\phi),\phi)}{\delta\phi(\boldsymbol{y})} = \frac{\delta\nabla q(\boldsymbol{r}_\Gamma)}{\delta\phi(\boldsymbol{y})}\cdot\nabla\phi(\boldsymbol{r}_\Gamma)\delta\phi(\boldsymbol{r}_\Gamma) + \nabla q(\boldsymbol{r}_\Gamma)\frac{\delta\nabla\phi(\boldsymbol{r}_\Gamma)}{\delta\phi(\boldsymbol{y})} + \nabla(\nabla q\cdot\nabla\phi(\boldsymbol{r}_\Gamma))\frac{\delta\boldsymbol{r}_\Gamma}{\delta\phi(\boldsymbol{y})} + \alpha(s)\left(\frac{\delta q(s,\boldsymbol{r}_\Gamma(\phi),\phi)}{\delta\phi(\boldsymbol{y})} + \nabla q(s,\boldsymbol{r}_\Gamma(\phi),\phi)\cdot\frac{\delta\boldsymbol{r}_\Gamma}{\delta\phi(\boldsymbol{y})}\right) \tag{B.11}$$

Inserting the boundary conditions as in the Dirichlet and Neumann cases into (B.5) and transforming $\dfrac{\partial q^\dagger(s-t,\boldsymbol{r})}{\partial\vec{n}}$ to $-\alpha^\dagger(s-t)q^\dagger(s-t,\boldsymbol{r})$ we get:

$$\delta Q_{stress}(s) = \frac{1}{V}\int_{\Gamma\times T} q^\dagger(s-t,\boldsymbol{r})\left(\frac{\partial^2 q(t,\boldsymbol{r})}{\partial\boldsymbol{n}^2} + \alpha(t)\frac{\partial q(t,\boldsymbol{r})}{\partial\boldsymbol{n}}\right)\delta\phi(\boldsymbol{r})\,dtd\boldsymbol{r} + \frac{1}{V}\int_{\Gamma\times T} -\nabla q(t,\boldsymbol{r})\cdot\nabla\delta\phi(\boldsymbol{r})\,dtd\boldsymbol{r} \tag{B.12}$$

Now, using theorems 3 and 4 (theorem 2.67 of Sokolowski and Zolesio in [88]) it can be decomposed into terms of $\delta\phi$ and terms of $\nabla\delta\phi$ by taking $f = q^\dagger\delta\phi$ and $g = q$:

$$\begin{aligned}
&\int_{\Gamma\times T} q^\dagger(1-s,\boldsymbol{r})\left(\frac{\partial^2 q(s,\boldsymbol{r})}{\partial \boldsymbol{n}^2} + \alpha(s)\frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\right)\delta\phi(\boldsymbol{r}) - \nabla q(s,\boldsymbol{r})\cdot\nabla\delta\phi(\boldsymbol{r}))\, d\Gamma ds \\
&\qquad = \int_{\Gamma\times T} q^\dagger(1-s,\boldsymbol{r})\Delta q(\boldsymbol{r},s)\delta\phi(\boldsymbol{r}) + \nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r}) \\
&\qquad - \frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\frac{\partial q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})}{\partial \boldsymbol{n}} - \kappa q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r})\frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\; d\Gamma\, ds \\
&\qquad + \int_{\Gamma\times T} \alpha(s) q^\dagger(1-s,\boldsymbol{r})\frac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\delta\phi(\boldsymbol{r})\; d\Gamma\, ds \\
&\qquad = \int_{\Gamma\times T} q^\dagger(1-s,\boldsymbol{r})\Delta q(\boldsymbol{r},s)\delta\phi(\boldsymbol{r}) + \nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\delta\phi(\boldsymbol{r}) \\
&\qquad + \alpha(s)(\kappa(\boldsymbol{r}) + \partial_{\boldsymbol{n}})(q(s,\boldsymbol{r})q^\dagger(1-s,\boldsymbol{r}))\delta\phi(\boldsymbol{r})\; d\Gamma\, ds \\
&\qquad = \int_{\Gamma\times T} \Big(q^\dagger(1-s,\boldsymbol{r})\big[q_s(s,\boldsymbol{r}) + q(s,\boldsymbol{r})w(\boldsymbol{r})\big]\Big)\delta\phi(\boldsymbol{r})\; d\Gamma\, ds, \\
&\qquad + \int_{\Gamma\times T} \Big(\nabla q(s,\boldsymbol{r})\cdot\nabla q^\dagger(1-s,\boldsymbol{r})\Big)\delta\phi(\boldsymbol{r})\; d\Gamma\, ds, \\
&\qquad + \int_{\Gamma\times T} \alpha(s)(\kappa(\boldsymbol{r}) + \partial_{\boldsymbol{n}})(q(s,\boldsymbol{r})q^\dagger(1-s,\boldsymbol{r}))\delta\phi(\boldsymbol{r})\; d\Gamma\, ds,
\end{aligned}$$

where in the first equality we use integration by parts on the surface, in the second equality we use the fact that $\dfrac{\partial q}{\partial \boldsymbol{n}} = -\alpha(s)q$ and $\dfrac{\partial \delta\phi}{\partial \boldsymbol{n}} = 0$ and in the third we use the PDE. We can obtain a symmetric expression as in Neumann for the first terms, and by inserting the densities ρ_A, ρ_B instead of $q, q^\dagger$ in the second integral and make use of the boundary conditions to obtain:

$$\delta Q_{stress} = \frac{1}{2}\int_\Gamma \Big(Q\Delta f_+(\boldsymbol{r}) - (\kappa(\boldsymbol{r}) + \partial_{\boldsymbol{n}})(\alpha_A\rho_A(\boldsymbol{r}) + \alpha_B\rho_B(\boldsymbol{r}))\Big)\delta\phi(\boldsymbol{r})\, d\Gamma. \tag{B.13}$$

Again we note the equivalence between equation (3.11) and equation (B.13). In addition if we use that $\alpha_A\rho_A + \alpha_B\rho_B = -\partial_{\boldsymbol{n}}(\rho_A + \rho_B)$ the last term can be written as $(\kappa(\boldsymbol{r})\partial_{\boldsymbol{n}} + \partial_{\boldsymbol{n}n})(-Qf_+(\boldsymbol{r}))$, and get the following symmetric expression:

$$\delta Q_{stress} = \frac{1}{2}\int_\Gamma Q(\Delta + \kappa(\boldsymbol{r})\partial_{\boldsymbol{n}} + \partial_{\boldsymbol{n}n})f_+(\boldsymbol{r})\delta\phi(\boldsymbol{r})\, d\Gamma. \tag{B.14}$$

Appendix C

Weak Form .vs. Strong Form

In equations (A.5) and (B.7) we get the same functional level-set derivative in the case of Dirichlet boundary conditions for both the weak form derivation and the non-weak form derivation. In equations (B.9), (A.6) at a first sight we do not get the same functional level-set derivatives using the weak and non-weak form but we showed in equation (B.10) using integration by parts on the surface that the integral on these two expressions is equal. However we note that in the weak form we do not get an explicit partial differential equation for $\delta q(s, \boldsymbol{r})$. In the case of the strong form, we obtain for $\delta q(s, \boldsymbol{r}) = \int g(s, \boldsymbol{r}, \boldsymbol{y}) \delta\phi(\boldsymbol{y}) d\boldsymbol{y}$ the following PDE for a specific and known in advance

$\delta\phi(\boldsymbol{r})$ change of the shape, which can be used to gain insight into the behavior of δq:

$$\begin{cases} \delta q(s,\boldsymbol{r}) = \Delta\delta q(s,\boldsymbol{r}) - w(s,\boldsymbol{r})\delta q(s,\boldsymbol{r}) \text{ in } \Omega \\ \\ \text{Dirichlet } \delta q(s,\boldsymbol{r}) = \dfrac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{n}}\delta\phi(\boldsymbol{r}) \text{ on } \Gamma, \\ \\ \text{Neumann } \dfrac{\partial \delta q(s,\boldsymbol{r})}{\partial \boldsymbol{n}} = \dfrac{\partial^2 q(s,\boldsymbol{r})}{\partial \boldsymbol{n}^2}\delta\phi(\boldsymbol{r}) - \dfrac{\partial q(s,\boldsymbol{r})}{\partial \boldsymbol{\tau}}\dfrac{\partial \nabla\delta\phi(\boldsymbol{r})}{\partial \boldsymbol{\tau}} \text{ on } \Gamma, \\ \\ \delta q(s=0,\boldsymbol{r}) = 0. \end{cases} \tag{C.1}$$

The weak form is convenient to implement a shape optimization procedure since we get a level-set sensitivity linear in $\delta\phi$ in contrast with the expression obtained in (B.9) which can not be used as is since it has $\nabla\delta\phi$ terms.

Appendix D

Cea's Method

This appendix outlines the method, introduced by Cea [12], that we use to derive the level-set derivative in the weak form in section 3.3. Consider a domain $\Omega = \{\boldsymbol{r} : \phi(\boldsymbol{r}) < 0\}$ in $\mathbb{R}^n$, where ϕ is a level-set function, and a functional J defined as:

$$J[u, \phi] = \int_{\Omega} u(\boldsymbol{r}) d\Omega.$$

Let $u(\boldsymbol{r})$ be constrained to satisfy a partial differential equation

$$E(\phi(\boldsymbol{r}), u(\boldsymbol{r})) = 0.$$

To find the change of J with respect to ϕ one introduces the Lagrangian $L[u, \phi] = J[u, \phi] + \int_{\Omega} \lambda(\boldsymbol{r}) E(\phi, u) d\Omega$. We highlight the steps of the derivation used by [12]:

1. Since $\delta L_{\delta\lambda} = \int_{\Omega} \delta\lambda E(\phi, u) d\Omega$ and $E(\phi, u) = 0$, u is the solution of $\delta L_{\delta\lambda} = 0$.

2. Since we choose λ to be the solution of $\delta L_{\delta u} = 0$, λ satisfies $\delta J_{\delta u} + \int_\Omega \lambda(\boldsymbol{r}) \delta E_{\delta u}(\phi, u) d\Omega = 0$,

3. When the constraint $E(u, \phi) = 0$ is satisfied we can write $j[u, \phi] = L(u, \phi, \lambda)$ so that the change of $j[u]$ with respect to the shape is given by: $\delta j_{\delta\phi} = \delta L_{\delta\phi} + \delta L_{\delta u} \delta u_{\delta\phi} + \delta L_{\delta\lambda} \delta\lambda_{\delta\phi}$,

4. Finally Hadamard's theorem states that the change of a functional $j[u]$ with respect to the shape depends on the values on Γ only [41], thus:

$$\delta j_{\delta\phi} = -\int_\Gamma \Big(u(\boldsymbol{r}) + \lambda(\boldsymbol{r}) E(\phi(\boldsymbol{r}), u(\boldsymbol{r})) \Big) \delta\phi(\boldsymbol{r}) d\Gamma$$

.

Chapter 5

Level Set Strategy for Inverse DSA-SCFT problem

We introduce a level-set strategy to find the geometry of confinement that will guide the self-assembly of block copolymers to a given target design in the context of lithography. The methodology is based on a shape optimization algorithm, where the level-set normal velocity is defined as the pressure field computed through a self-consistent field theory simulation. We present numerical simulations that demonstrate that this methodology is capable of finding guiding templates for a variety of target arrangements of cylinders and thus is an effective approach to the inverse directed self-assembly problem.

5.1 Introduction

Directed self-assembly (DSA) of block copolymers is now recognized as a complementary patterning technique to accompany the continuous miniaturization necessary for next generation lithography [27, 37]. Similar to conventional patterning techniques such as photolithography, DSA relies on a prepatterned mask to transfer and print a target pattern on a wafer. DSA is a bottom-up approach however, and the presence of the guiding mask is not only necessary to the formation of targeted arrangements, but is critical for the precise control over long-range order and the elimination of undesirable defects [5]. In contrast to self-assembly of block copolymers in bulk and thin films, using chemically prepatterned surfaces [45, 22] or topographical confinement [84] for example, it was demonstrated in a large body of experimental and numerical studies how DSA can affect the orientation of the microdomains during phase separation and produce target features with high reproducibility and minimal defectivity [93, 23].

Of the many components of an integrated circuit, vertical interconnect access (VIA) structures, also called contact holes, are prime candidates for a simple and cost-effective implementation of a DSA-assisted patterning technique [17, 82]. In VIA lithography, the main goal is to produce high resolution cylinders with a reduced critical dimension relative to a larger guiding template. While the confining template can be generated using conventional optical lithography, the cylinders within are the result of the self-assembly of cylinder-forming block copolymers and a subsequent selective etch process to remove the minority block. Beyond hole shrinking, cylinder placement accuracy is

also essential to the successful implementation of a DSA scheme for VIA lithography [95, 52]. Because VIAs are the components of an integrated circuit that connect various conducting layers, it is imperative that the formed cylinders be accurately located to allow current to pass through the layers [17].

The forward problem, consisting of predicting the resulting self-assembled features of block copolymers at equilibrium within a given and fixed confining shape, has received most of the attention in recent computational studies of DSA for VIA lithography [43, 49, 44]. Using field-based simulations [24], the hole shrinking and placement errors were extensively characterized in both single and arrays of VIAs in various guiding templates. Mean-field approachs, such as self-consistent field theory (SCFT), or including thermal fluctuations, as in complex Langevin (CL) simulations [24], have enabled an exhaustive exploration of the vast parameter space in VIA DSA. In addition to optimal conditions, which favor the emergence of defect-free structures, field-based simulations were used to estimate the formation energies of defective structures and the kinetic barriers for the melting of such defects into perfect morphologies [51, 52, 40].

Contrary to the forward problem, the inverse problem, which consists of constructing an adequate confining mask *shape* that leads to a given self-assembled morphology of the block copolymers, has received little attention in computational investigations. With the increasing complexity of design rules of circuit layouts and the decreasing dimensions of integrated components, there is high demand for the cost-efficient design and fabrication of mask templates that accurately produce a given arrangement of morphologies on a

wafer. While technologies based on double and quadruple patterning steps must resort to multiple concurrent masks to achieve target layouts, a solution to the DSA inverse problem can potentially offer an inexpensive alternative to produce similar target layouts while keeping the number of masking steps at a minimum. In the context of VIA lithography, the emphasis is therefore on mask shapes that can direct the self-assembly of the block copolymer into various cylindrical structures with a prescribed diameter and location. Furthermore, the resulting mask must be as smooth as possible to integrate fabrication constraints.

Recently, Latypov *et al.* introduced a methodology for shape optimization in DSA lithography [54, 53]. They proposed an inverse design algorithm where the shape is unknown and the target density is fixed. While successfully predictive in the proof-of-concept examples therein, the method relies on a linearization of the mean field equations along with a primitive parameterization of the unknown shape. The extension of the algorithm to the full non-linearized equations and complex shapes remains to be investigated.

The level-set formalism [73] has been successfully used to compute optimal shapes in a variety of problems, ranging from inverse imaging to structural engineering and fluid mechanics, e.g. [74, 1, 94, 86, 72] and the references therein. An advantage of the level-set representation is that no parametrization is required to characterize a free boundary and its dynamics, making it well-suited to represent arbitrary shapes. Another advantage is that the level-set function can be used to locate *exactly* the position of the

interface in order to impose *sharp* boundary conditions. In our case, this will prove useful to impose effective boundary conditions.

In this paper, we introduce a level-set approach for the search of optimal masks in DSA lithography. We demonstrate how the advantages of the level-set method and the full nonlinear self-consistent mean field theory (SCFT) can be leveraged in tandem to compute optimal guiding templates for a variety of target arrangements of cylinders in the context of VIA lithography. Of course, the shape optimization algorithm we present could be used with a linearization of the mean field equations, as in [54, 53], benefiting from the advantages of a parametrization-free representation. However, although it would be straightforward and may provide advantages in terms of computational efficiency, this approach would be at the expense of physical accuracy. We thus use SCFT simulations and in particular, we consider Neumann boundary conditions. The choice of Neumann (or effective) boundary conditions for molten polymers at air or solid interfaces is justified (see De Gennes [21]) and offers the advantage that values at the mask's boundary, where the level-set velocity is computed, remain smooth [79, 77]. In turn, the level-set evolution is robust to variations in the numerical parameters, a property that would otherwise be difficult to guarantee. The shape optimization procedure we introduce involves two stages. In the first stage, the level-set function representing the mask's shape is evolved to a steady state using the pressure field obtained from SCFT simulations. The resulting shape is then fixed and used in the second stage where both the exchange and pressure fields from SCFT are converged to their saddle point values.

The algorithm is successful if the target density is obtained at the end of the second stage. Otherwise, a correction stage is added and the process repeats.

The paper is organized as follows. In section 3.2, we describe both the level-set equations and the SCFT framework in the case of free moving boundaries. The proposed shape optimization algorithm, a level-set-based circle detection algorithm that we use in the correction step, and the numerical strategies will be detailed in section 5.5. Finally, section 5.6 presents numerical results on several relevant examples of target density structures.

Diblock copolymer chains are comprised of two connected blocks A and B, and can be parametrized by the A-block fraction parameter f_A along with an interaction parameter χ_{AB}. The first block contains polymerized A monomers and the second block consists of B monomers. f_A is the fraction of A in the chain and χ_{AB} quantifies the net repulsive interaction between monomers of A and B. An important goal in the study of assemblies of molten diblock copolymers is to understand the pattern selection of self-assembled, spatially periodic mesophases at equilibrium as a function of f_A and χ_{AB} [24, 7].

Self consistent field theory (SCFT) is a successful mean-field theory and a mature computational tool for describing inhomogeneous phases of block copolymers in the parameter space (f_A, χ_{AB}) [24, 35, 36]. It has enabled researchers to predict and understand block copolymer self assembly in a wide range of applications [24, 60, 61, 59]. In the following section, we describe the required equations for the current computational study in the context of SCFT [24, 35, 36], i.e. a standard model of molten incompress-

ible diblock copolymer. In 5.3, we present a corresponding model for a compressible diblock melt confined to an arbitrary domain by a mask. Detailed and rigorous derivations for the incompressible model can be found in [24] and the references therein and in [8, 39, 15] for the mask model. A more physically-motivated presentation can be found in the excellent review of Matsen [57]. For the shape optimization simulations, we use only the incompressible AB diblock model with Neumann boundary conditions. For the simulations with an arbitrary initial configuration and a fixed shape (see section 5.6.2), we use a compressible AB diblock copolymer melt with a mask model (5.3).

5.2 Incompressible AB Diblock Copolymer in the Melt

The physical system under consideration is an incompressible melt composed of AB diblock copolymers. The field-based theory of polymer thermodynamics uses a Hubbard-Stratonovitch transform to convert the partition function from a particles representation to a field representation (see [24, 35, 36] and references therein). One therefore searches for the statistical fields w_- and w_+ that optimize the Hamiltonian[1]:

$$H[w_+, w_-] = \int \left(-w_+(\boldsymbol{r}) + \frac{w_-^2(\boldsymbol{r})}{\chi_{AB}}\right) d\boldsymbol{r} - V \ln Q[w_+, w_-], \tag{5.1}$$

[1]This is actually H/n, where n is the number of molecular chains; χ_{AB} is actually $\chi_{AB}N$ where we absorb N, the number of segments in a polymeric chain.

where the normalized partition function Q can be formulated as:

$$Q[w_+, w_-] = \frac{1}{V} \int q(\boldsymbol{r}, s = 1; [w_+, w_-])\, d\boldsymbol{r}. \tag{5.2}$$

In these equations, $w_+(\boldsymbol{r})$ acts as a pressure potential that enforces the incompressibility constraint and $w_-(\boldsymbol{r})$ is the exchange potential, which is conjugate to the density difference between A and B. The functionals q and $q^\dagger$ are the solution of a Fokker-Planck equation, which is solved from $s = 0$ to $s = 1$, with initial conditions $q(s = 0) = 1$. Here, $s \in [0, 1]$ is a contour variable that describes the location along the backbone of the chain. The functionals q and $q^\dagger$, i.e. the chain propagators, represent the statistical weight of the chain at location $\boldsymbol{r}$ and contour length s. The boundary conditions are homogeneous Neumann on Γ, $\dfrac{\partial q}{\partial n} = 0$ (see figure 5.1), for the forward and backward Fokker-Planck equations:

$$\begin{cases} \partial_s q(s, \boldsymbol{r}) &= \bigtriangledown^2 q(s, \boldsymbol{r}) - q(s, \boldsymbol{r}) \times w(\boldsymbol{r}) \qquad \text{(forward)}, \\ \\ \partial_s q^\dagger(s, \boldsymbol{r}) &= \bigtriangledown^2 q^\dagger(s, \boldsymbol{r}) - q^\dagger(s, \boldsymbol{r}) \times w^\dagger(\boldsymbol{r}) \quad \text{(backward)}. \end{cases} \tag{5.3}$$

The forward potential $w(\boldsymbol{r})$ is calculated as follows:

$$w(\boldsymbol{r}) = \begin{cases} w_A(\boldsymbol{r}) = w_+(\boldsymbol{r}) - w_-(\boldsymbol{r}) \qquad 0 < s < f_A, \\ \\ w_B(\boldsymbol{r}) = w_+(\boldsymbol{r}) + w_-(\boldsymbol{r}) \qquad f_A < s < 1, \end{cases} \tag{5.4}$$

and the backward potential $w^{\dagger}(\boldsymbol{r})$:

$$w^{\dagger}(\boldsymbol{r}) = \begin{cases} w_B(\boldsymbol{r}) = w_+(\boldsymbol{r}) + w_-(\boldsymbol{r}) & 0 < s < 1 - f_A, \\ \\ w_A(\boldsymbol{r}) = w_+(\boldsymbol{r}) - w_-(\boldsymbol{r}) & 1 - f_A < s < 1. \end{cases} \tag{5.5}$$

The normalized local densities (volume fraction) of the components A and B are defined as:

$$\rho_{A,B}(\boldsymbol{r}; [w_+, w_-]) = -\frac{V}{Q[w_+, w_-]} \frac{\delta Q[w_+, w_-]}{\delta w_{A,B}(\boldsymbol{r})}. \tag{5.6}$$

Elaboration of these functional derivatives leads to important equations relating the densities to the forward and backward propagators:

$$\begin{aligned} \rho_A(\boldsymbol{r}; [w_+, w_-]) = & \frac{1}{Q[w_+, w_-]} \int_0^{f_A} q^{\dagger}(1 - s, \boldsymbol{r}, [w_+, w_-]) q(s, \boldsymbol{r}, [w_+, w_-])\, ds \\ \\ \rho_B(\boldsymbol{r}; [w_+, w_-]) = & \frac{1}{Q[w_+, w_-]} \int_{f_A}^{1} q^{\dagger}(1 - s, \boldsymbol{r}, [w_+, w_-]) q(s, \boldsymbol{r}.[w_+, w_-])\, ds \end{aligned} \tag{5.7}$$

In the mean field approximation, where fluctuations can be neglected, it is sufficient to find the saddle point for the Hamiltonian, i.e. $\dfrac{\delta H}{\delta w(\boldsymbol{r})} = 0$. The free energy of the melt is then given by the Hamiltonian at the saddle point. The functional derivatives with respect to the statistical fields, i.e. the driving forces of the optimization procedure[2] are computed using the normalized densities as follows:

$$f_+(\boldsymbol{r}) = \frac{\delta H}{\delta w_+(\boldsymbol{r})} = \rho_A(\boldsymbol{r}) + \rho_B(\boldsymbol{r}) - 1, \tag{5.8}$$

$$f_-(\boldsymbol{r}) = \frac{\delta H}{\delta w_-(\boldsymbol{r})} = \frac{2w_-(\boldsymbol{r})}{\chi_{AB}} + \rho_B(\boldsymbol{r}) - \rho_A(\boldsymbol{r}). \tag{5.9}$$

[2]These are related to the physical, thermodynamic forces that drive self-assembly.

We look for the saddle of $H[w_+, w_-]$ in terms of the statistical fields w_- (the exchange field) and w_+ (the pressure field). This is a highly nonlinear functional optimization problem that requires an iterative solver where w_+ and w_- are evolved according to:

$$\begin{cases} w_+^{t+1}(\boldsymbol{r}) = & w_+^t(\boldsymbol{r}) + \lambda f_+(\boldsymbol{r}), \\ \\ w_-^{t+1}(\boldsymbol{r}) = & w_-^t(\boldsymbol{r}) - \lambda f_-(\boldsymbol{r}). \end{cases} \tag{5.10}$$

The pressure field is evolved in an ascent direction to enforce incompressibility and the exchange field is evolved in a descent direction to reduce the energy. The convergence is established when the L_2 norms $F_- = (\frac{1}{V}\int_V f_-^2(\boldsymbol{r})dV)^{1/2}$ and $F_+ = (\frac{1}{V}\int_V f_+(\boldsymbol{r})^2 dV)^{1/2}$ are less than a given tolerance. In this work, we use the method of Ouaknin *et al.* [77] for solving the SCFT equations with Neumann boundary conditions on arbitrary domains.

5.3 The Mask Model

The physical system is a compressible melt composed of AB diblock copolymers. The Hamiltonian for a confined compressible polymer with a mask [8, 39, 15, 43, 49, 44]

is given by[3]:

$$
\begin{aligned}
H[w_+, w_-, \phi_w] = & \\
& \int \Big(\frac{-0.5\zeta^{-1}}{0.5\chi_{AB}\zeta^{-1}+1} w_+^2(\boldsymbol{r}) + \frac{(\chi_{w_+}\zeta^{-1}+1)\phi_w(\boldsymbol{r})-1}{0.5\chi_{AB}\zeta^{-1}+1} w_+(\boldsymbol{r}) \, d\boldsymbol{r} \\
& + \int \frac{2\chi_{w_-}}{\chi_{AB}} \phi_w(\boldsymbol{r}) w_-(\boldsymbol{r}) + \frac{w_-^2(\boldsymbol{r})}{\chi_{AB}} \, d\boldsymbol{r} \\
& - \bar{\Phi}_w V \ln Q[w_+, w_-],
\end{aligned}
$$

$\phi_w(\boldsymbol{r})$ is defined as the wall volume fraction which augments the pressure field to repel the polymeric material and localize it inside the confined domain. That is the wall volume fraction defines the mask and takes a value of 1 outside the domain, 0 inside, 0.5 at the interface and varies smoothly at the interface. $\chi_{w_+} = \chi_{w_a} + \chi_{w_b}, \chi_{w_-} = \chi_{w_a} - \chi_{w_b}$ where χ_{w_a}, χ_{w_b} are the chemical interaction between the polymeric components A and B with the wall. ζ^{-1} is the compressibility tolerance and tends to zero as the material becomes incompressible. The boundary conditions are periodic in all directions $q(\boldsymbol{r}) = q(\boldsymbol{r} + \boldsymbol{L})$. The net advantage of this model is the possibility to use the `FFT` algorithm [18] with the `FFTW` library [25, 26]. We use a level-set function $\phi(\boldsymbol{r})$ to define the wall volume fraction:

$$
\phi_w(\boldsymbol{r}) = \frac{1}{2}\Big(\frac{e^{\phi(\boldsymbol{r}/\delta_w)} - e^{-\phi(\boldsymbol{r}/\delta_w)}}{e^{\phi(\boldsymbol{r}/\delta_w)} + e^{-\phi(\boldsymbol{r}/\delta_w)}} + 1 \Big), \tag{5.11}
$$

where the level-set is the signed distance function such that is negative inside the mask and positive outside. $\bar{\Phi}_w$ is the effective domain fraction.

$$
\bar{\Phi}_w = \frac{1}{V} \int 1 - \phi_w(\boldsymbol{r}) \, d\boldsymbol{r}.
$$

[3]This is actually H/n, where n is the number of molecular chains, χ_{AB}, ζ^{-1}, χ_{w_-}, χ_{w_+} are actually $\chi_{AB}N$, $(\zeta N)^{-1}$, $\chi_{w_-}N$, $\chi_{w_+}N$ where we absorbed N, the number of segments in a polymeric chain.

$Q[w_A, w_B], w_A(\boldsymbol{r}), w_B(\boldsymbol{r}), w(\boldsymbol{r}), w^\dagger(\boldsymbol{r})$ are defined in the same way than in the incompressible case and $q(s, \boldsymbol{r}), q^\dagger(s, \boldsymbol{r})$ solve the same Fokker-Planck equation than in the incompressible case.

The normalized local densities (volume fraction) of the components A and B ρ_A and ρ_B are computed as:

$$\begin{aligned} \rho_A(\boldsymbol{r}; [w_A, w_B]) = & \frac{\bar{\Phi}_w}{Q[w_A, w_B]} \int_0^f q^\dagger(1-s, \boldsymbol{r}, [w_A, w_B]) q(s, \boldsymbol{r}, [w_A, w_B])\, ds. \\ \rho_B(\boldsymbol{r}; [w_A, w_B]) = & \frac{\bar{\Phi}_w}{Q[w_A, w_B]} \int_f^1 q^\dagger(1-s, \boldsymbol{r}, [w_A, w_B]) q(s, \boldsymbol{r}, [w_A, w_B])\, ds. \end{aligned} \tag{5.12}$$

and the functional derivatives with respect of the statistical fields are given by:

$$f_+(\boldsymbol{r}) = \frac{\delta H}{\delta w_+(\boldsymbol{r})} = \rho_A(\boldsymbol{r}) + \rho_B(\boldsymbol{r}) + \frac{-\zeta^{-1}}{0.5\chi_{AB}\zeta^{-1} + 1} w_+(\boldsymbol{r}) + \frac{(\chi_{w_+}\zeta^{-1} + 1)\phi_w(\boldsymbol{r}) - 1}{0.5\chi_{AB}\zeta^{-1} + 1}. \tag{5.13}$$

$$f_-(\boldsymbol{r}) = \frac{\delta H}{\delta w_-(\boldsymbol{r})} = \frac{2w_-(\boldsymbol{r})}{\chi_{AB}} + \rho_B(\boldsymbol{r}) - \rho_A(\boldsymbol{r}) + 2\frac{\chi_{w_-}}{\chi_{AB}}\phi_w(\boldsymbol{r}). \tag{5.14}$$

5.4 Level-Set Equations

In our approach, the shape of the confining template (or mask) is described through a level-set function $\phi(\boldsymbol{r})$, which describes the boundary of the mask in an implicit way [73]. Specifically, the boundary Γ is defined as $\phi(\boldsymbol{r}) = 0$, Ω_- as $\phi(\boldsymbol{r}) < 0$ and Ω_+ as $\phi(\boldsymbol{r}) > 0$ (see figure 5.1). The Hamiltonian introduced in section 5.2 is thus now a functional of $\phi(\boldsymbol{r})$ in addition of the pressure potential $w_+(\boldsymbol{r})$ and of the exchange potential $w_-(\boldsymbol{r})$, i.e $H = H[\phi(\boldsymbol{r}), w_+(\boldsymbol{r}), w_-(\boldsymbol{r})]$. In this work, we use the level-set method on adaptive grids

introduced in Min and Gibou [67]. In addition we make use of the length and surface geometric integral procedures introduced in Min and Gibou [64, 65], required to build the different discretizations and integrate quantities over an arbitrary domain.

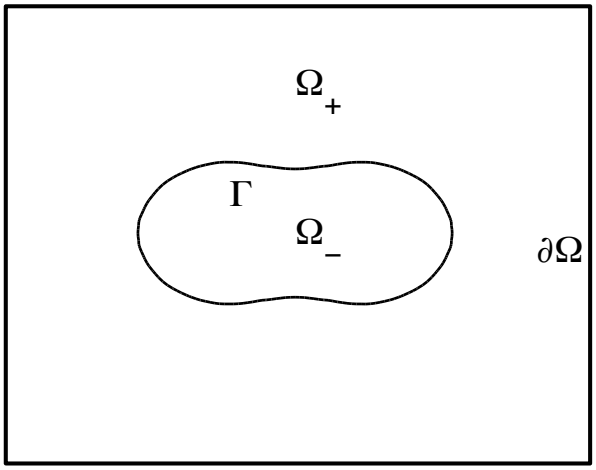


Figure 5.1: The Fokker-Planck equation is solved in Ω_- with Neumann boundary conditions on Γ.

There are three main equations in the level-set framework that we will use: the level-set advection equation (5.15), the level-set reinitialization equation (5.16), and the equation to extrapolate a scalar quantities across the shape's boundary (5.17).

The coupling between the SCFT simulations and the level-set framework is through the definition of the level-set velocity, $\boldsymbol{v}(\boldsymbol{r})$, which evolves the geometry of the template it represents via the advection equation:

$$\frac{\partial\phi(\boldsymbol{r})}{\partial t} + \boldsymbol{v}(\boldsymbol{r}) \cdot \nabla\phi(\boldsymbol{r}) = 0. \tag{5.15}$$

As we will explain in more detail later, in the present work we choose a velocity that is normal to the local boundary and proportional to the pressure field: $\boldsymbol{v}(\boldsymbol{r}) = w_+(\boldsymbol{r})\boldsymbol{n}$.

The reinitialization equation receives as input a level-set function and transforms it into a signed distance function with the same 0-contour. The reinitialization equation does not couple to the SCFT model; it is a post-processing step required to enforce the level-set function as a signed distance function, which is desired to compute robustly spatial derivatives of the level-set function and that can be used in other procedures [86, 72]. The reinitialization equation is:

$$\frac{\partial \phi(\boldsymbol{r})}{\partial \xi} + \texttt{sign}(\phi(\boldsymbol{r})(|\nabla \phi(\boldsymbol{r})| - 1) = 0, \tag{5.16}$$

where ξ is a pseudo time and `sign` refers to the signum function.

The extrapolation equation does not change the values of the level-set but uses it to extrapolate scalar quantities from the inside ($\phi(\boldsymbol{r}) < 0$) to the outside ($\phi(\boldsymbol{r}) > 0$) of an irregular domain. The extrapolation procedure is used in the present context to extrapolate the values of the pressure field w_+ to a narrow band of width d of the interface ($\phi(\boldsymbol{r}) > 0 \cap \phi(\boldsymbol{r}) < d$) without changing the values of $w_+(\boldsymbol{r})$ where $\phi(\boldsymbol{r}) < 0$. This procedure thus specifies the velocity $\boldsymbol{v}(\boldsymbol{r})$ within a narrow band of the level-set, which is desirable to obtain robust numerical results. The extrapolation equation (5.17) for the pressure field is based on solving a partial differential equation as in [2]:

$$\frac{\partial w_+(\boldsymbol{r})}{\partial \xi} + \mathcal{H}(\phi(\boldsymbol{r}))\nabla w_+(\boldsymbol{r}) \cdot \mathbf{n} = 0, \tag{5.17}$$

where $\mathcal{H}$ is the Heaviside function. We also use equation (5.17) to extrapolate the exchange field $w_-(\boldsymbol{r})$ in the domain that is swept by the interface during the computational times t and $t+1$, i.e. the region defined by ($\phi^t(\boldsymbol{r}) > 0 \cap \phi^{t+1}(\boldsymbol{r}) < 0$), similarly to what

is done in [31]. The extrapolated values of w_+ and w_- in $(\phi^t(\boldsymbol{r}) > 0 \cap \phi^{t+1}(\boldsymbol{r}) < 0)$ are used for solving the Fokker-Planck equations and for computing the Hamiltonian at the next mean field step in the advected domain $(\phi^{t+1}(\boldsymbol{r}) < 0)$.

5.5 Level-Set-Based Inverse Design Algorithm with Constraints

5.5.1 Shape Optimization Algorithm

The problem of interest is to find the shape of the confinement template that will direct the self-assembly of polymer to target density. The algorithm we introduce is given in Algorithm 7, which consists of two stages: an shape optimization phase and a standard SCFT optimization stage.

In the shape optimization stage, which seeks to attract the polymer to our desired design target, we seed $w_-(\boldsymbol{r})$ by setting $f_-(\boldsymbol{r}) = 0$ (see equation (5.9)) and deducing $w_-(\boldsymbol{r})$ after enforcing $\rho_A(\boldsymbol{r})$ and $\rho_B(\boldsymbol{r})$ to match the target design:

$$w_-^{seed}(\boldsymbol{r}) = \frac{\chi_{AB}}{2}(\rho_A^{target}(\boldsymbol{r}) - \rho_B^{target}(\boldsymbol{r})).$$

In the first stage of the algorithm, the exchange field is frozen and the shape and the pressure potential are evolved. We initialize the level-set function to contain the target (see figure 5.2) and the pressure field to be zero everywhere. In the second stage, the shape is frozen while both the pressure and exchange fields are evolved. We

Algorithm 7 Procedure to find an optimal shape for a prescribed pattern

Stage 1 (shape optimization): freeze $w_-(\boldsymbol{r})$ and evolve $w_+(\boldsymbol{r})$ and $\phi(\boldsymbol{r})$.

while $||\boldsymbol{v}||_2 > \epsilon_v$ **do**

1. Solve two Fokker-Planck equations for $q(s, \boldsymbol{r})$ and $q^\dagger(s, \boldsymbol{r})$ using (5.3);
2. Compute the densities $\rho_A(\boldsymbol{r})$ and $\rho_B(\boldsymbol{r})$ via (5.6);
3. Compute the force for $w_+(\boldsymbol{r})$ with equation (5.8) and the velocity $\boldsymbol{v}(\boldsymbol{r})$ for $\phi(\boldsymbol{r})$;
4. Advance the potential $w_+(\boldsymbol{r})$ and $\phi(\boldsymbol{r})$ with a constraint on the curvature.

Stage 2 (standard SCFT optimization): freeze $\phi(\boldsymbol{r})$ evolve $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$.

while $||\delta H/\delta\omega||_2 > \epsilon_w$ **do**

1. Solve two Fokker-Planck equations for $q(s, \boldsymbol{r})$ and $q^\dagger(s, \boldsymbol{r})$ using (5.3);
2. Compute the densities $\rho_A(\boldsymbol{r})$ and $\rho_B(\boldsymbol{r})$ via (5.6);
3. Compute the forces for $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$ with equations (5.8) and (5.9);
4. Advance the potentials $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$ via (5.10).

note that the second stage is the standard SCFT algorithm where we optimize for the statistical fields w_+ and w_-. As previously mentioned, we take the level-set velocity to be $v(\boldsymbol{r}) = w_+(\boldsymbol{r})\boldsymbol{n}$. The physical intuition for this choice is as follows: $w_+(\boldsymbol{r})$ is positive (resp. negative) in the case where the pressure potential seeks to expel (resp. attract) the polymeric material because it violates the incompressibility constraint. In turn, this means the confined domain should be forced to evolve in that direction.

5.5.2 Level-Set-Based Circle Detection Algorithm

In order to automatically assess if the targeted design is obtained, and correct the template's shape in the event it is not, we developed an algorithm that automatically detects the number of circles, their centers and their radii. Generic circle detection algorithms can be applied to a wide variety of configurations but may be computationally expensive and require considerable amount of memory [38, 3, 96, 20]. In the DSA SCFT context, prior information can be leveraged to develop an $O(M)$ circle detection algorithm, where M is the number of grid points. The first question that we ask is how to ensure that we have the desired number of circles and radius size $n_{\text{desired}}, R_{\text{desired}}$.

In this context, we compute the length, L_ψ, and the surface, S_ψ, of the minority region A. For simplicity, we describe the minority region by another level-set function, $\psi(\boldsymbol{r})$: $\psi(\boldsymbol{r})$ takes negative values in A-rich regions and positive values in B-rich regions and is zero at the A-B interface. We assume that the circles have the same size so we can write $S_\psi = \pi(nR^2)_{\text{detected}}$ and $L_\psi = 2\pi(nR)_{\text{detected}}$, where n_{detected} and R_{detected}

are the number of circles detected and their corresponding radius, respectively. This implies that $R_{\text{detected}} = \dfrac{2S_\psi}{L_\psi}$ and $n_{\text{detected}} = \dfrac{L_\psi^2}{4\pi S_\psi}$. In the case where $|n_{\text{detected}} - n_{\text{desired}}|$ or $|R_{\text{detected}} - R_{\text{desired}}|$ is larger than a given threshold, we conclude that Algorithm 7 has not reached the target design. The quantity $|n_{\text{detected}} - n_{\text{desired}}|$ is a measure of the deviation from a perfect circle shape or from the target number of circles, while $|R_{\text{detected}} - R_{\text{desired}}|$ quantifies the deviation with respect to the circles' radius.

The procedure is given in Algorithm 8, where $\boldsymbol{r}_i, \boldsymbol{r}_j$ are the centers location detected and r_{min} is the set of the centers location. To find the length L_ψ and surface S_ψ of the minority region we use a level-set approach [64, 65] by segmenting the A region. By using equation (5.9) at equilibrium with $\rho_A = \rho_B = 0.5$ we see that at the interface between A and B, the exchange field $w_-(\boldsymbol{r})$ is zero and changes sign. Therefore, we segment the regions by using $w_-(\boldsymbol{r})$ as initial condition for the level-set $\psi(\boldsymbol{r})$ and reinitialize $\psi(\boldsymbol{r})$ with equation (5.16). Finally, we leverage the fact that if we have only circles of the same size, the minimums of the level-set values are located at the center of the circles: to find the centers' location, we find the $n_{detected}$ points with the lowest level-set values and which have a distance higher than $R_{detected}$ between themselves.

Algorithm 8 Procedure to detect circles

1. Take $\dfrac{w_{-}(\boldsymbol{r})}{\chi_{AB}}$ as initial condition for a level-set distance function ψ;
2. Reinitialize $\psi(\boldsymbol{r})$ using the reinitialization equation (5.16);
3. Compute the surface integral $S_{\psi} = \int_{\psi<0} 1 d\boldsymbol{r}$ as in [64, 65];
4. Compute the length integral $L_{\psi} = \int_{\psi=0} 1 d\boldsymbol{r}$ as in [64, 65];
5. Determine the number of circles $(n_{detected})$ and radii $R_{detected}$ from $S_{\psi} = (n\pi R^2)_{detected}$ and $L_{\psi} = 2\pi(nR)_{detected}$;
6. If $n_{detected} = n_{desired}$, find the set of the first $n_{detected}$ minima r_{min} such that $|\boldsymbol{r}_i - \boldsymbol{r}_j| < R_{detected}$ for each $\boldsymbol{r}_i, \boldsymbol{r}_j \in r_{min}$.

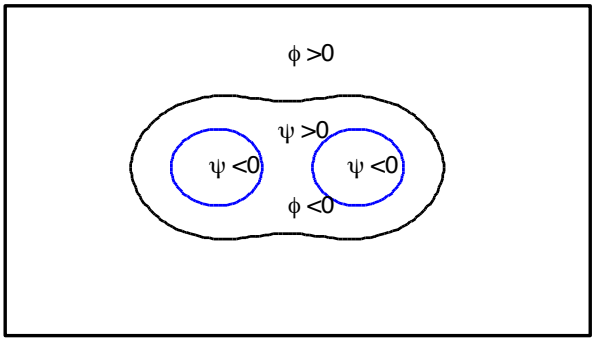


Figure 5.2: $\phi(\boldsymbol{r})$ is the level-set function which describes the shape of the confined domain and $\psi(\boldsymbol{r})$ is the level-set which describes the density of the A-component of the polymeric melt, i.e. the target.

In reality, since the $w_{-}(\boldsymbol{r})$ field will drift in the second stage of Algorithm 7, the density fields will drift off the target but the mismatch is typically small. We correct the target *centers' location* by an iterative procedure that repositions the circles using

equation (5.18):

$$\boldsymbol{r}_{target}^{t+1} = \boldsymbol{r}_{target}^{t} + \lambda_{\boldsymbol{r}}(\boldsymbol{r}_{target}^{0} - \boldsymbol{r}_{detected}^{t}) \tag{5.18}$$

where $\boldsymbol{r}_{target}^{t}$ and $\boldsymbol{r}_{detected}^{t}$ are the centers' locations of the targets and of the circles detected, respectively at iteration t and $\lambda_{\boldsymbol{r}}$ is the amplitude of the correction. In this procedure, we neglect the shape or size drifts.

5.5.3 Constrained Curvature

As described in the introduction, in order to integrate fabrication constraints, it is necessary to constrain the curvature of the optimized shape to a critical curvature κ_{cr}. We simply set $v_n(\boldsymbol{r}) = 0$ in the case where $\kappa(\boldsymbol{r}) > \kappa_{cr}$. Doing so enables the shape to react to the constraint not only locally where the shape would otherwise have a curvature larger than what is permitted, but globally, as other parts of the shape can rearrange as a response to the constraint. Another approach could be to perform an unconstrained shape evolution and then smooth the shape using motion-by-mean-curvature [87], while keeping the morphology. We use the former approach.

5.5.4 Solving the Fokker-Planck Equation with Neumann Boundary Conditions

The Fokker-Planck equation with Neumann boundary conditions for SCFT is solved by the method introduced in Ouaknin *et al.* [77], which is based on a Strang splitting [91] with boundary conditions imposed with the method of Papac *et al.* on an adaptive

quadtree grid [79]. The quadtree (see figure 5.5) is parametrized by its min and max levels, which correspond to the coarser and finer cells. For example in a $[0,1]^2$ domain, a quadtree with levels 4-7 will have the coarsest cells with size $1/2^4 = 1/16$ and finest cells with size $1/2^7 = 1/128$. The parallelization on adaptive quadtree grids is done using the recent library `parCASL` developed by Mirzadeh *et al.* [68]. The mask model given in 5.3 is solved with pseudo-spectral algorithms for SCFT with a Strang splitting [81] using the `FFT` algorithm [18] encoded in the `FFTW` library [25, 26].

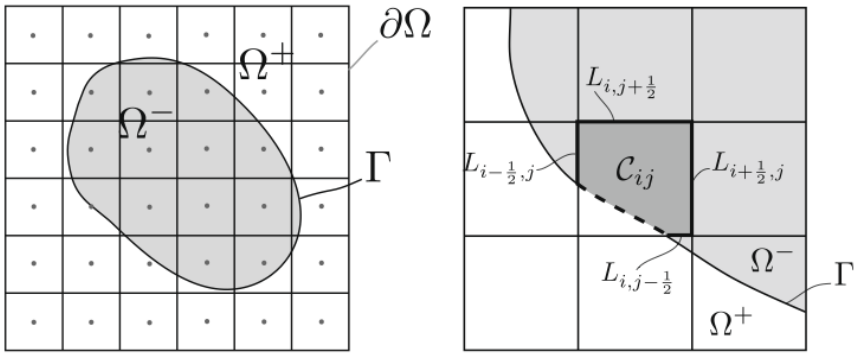
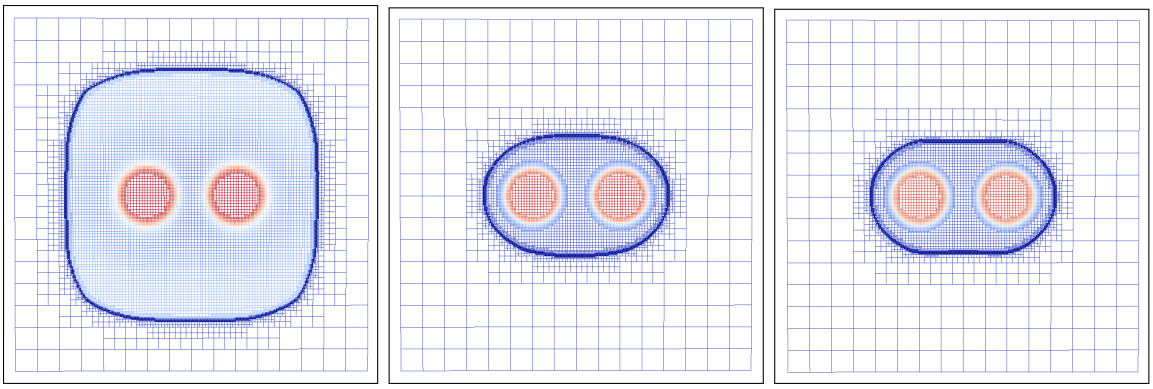


Figure 5.3: Cell-based discretization of the domain (left) and cell cut by the interface (right)



(a) initial stage of the grid's evolving shape (b) grid's evolving shape (c) final stage of the grid's evolving shape

Figure 5.4: Shape Optimization: the quadtree grid enables the refinement of regions of interest; here the domain's boundary and the interface between the minority region A and the majority region B.

For the sake of completeness, we will present here the principal results needed for the current study. We use the unconditionally stable Crank-Nicholson scheme to solve the equations (5.3) in s. For example, denoting q^s the solution vector at the contour position s, the discretization of the forward case without the external field is:

$$\frac{q^{s+1} - q^s}{\Delta s} = \frac{1}{2}\left(\widetilde{\nabla}^2 q^{s+1} + \widetilde{\nabla}^2 q^s\right), \tag{5.19}$$

where $\widetilde{\nabla}^2$ refers to the discretization of the ∇^2 operator. The nodes adjacent to the irregular domain (boundary nodes), see figure 5.3, are treated using a finite volume approach as in [79, 80], while the interior nodes are treated with finite differences on quadtree as introduced in [66].

The Laplacian can thus be written as:

$$\widetilde{\nabla}^2 q = Mq,$$

where M is a matrix that depends only on the quadtree data structure. The Crank-Nicholson scheme with Strang splitting for the Fokker-Planck equation can thus be written as:

$$q^{s+1} = e^{-w\frac{\Delta s}{2}I} M_2^{-1} M_1 e^{-w\frac{\Delta s}{2}I} q^s, \tag{5.20}$$

where $M_1 = \left(I + \frac{\Delta s}{2}M\right)$ and $M_2 = \left(I - \frac{\Delta s}{2}M\right)$. We note that M_2^{-1} has been shown to exist in [66].

Referring to figure 5.3 and considering the Crank-Nicholson time stepping, the discretization for the boundary nodes is based on a finite volume approach giving:

$$\int_{C_{i,j}\cap\Omega^-} \frac{q_{i,j}^{s+1} - q_{i,j}^s}{\Delta s} dA = \frac{1}{2}\left(\int_{\partial C_{i,j}\cap\Omega^-} \nabla q_{i,j}^{s+1} \cdot \boldsymbol{n}\, dL + \int_{\partial C_{i,j}\cap\Omega^-} \nabla q_c^s \cdot \boldsymbol{n}\, dL\right),$$

where we have used the homogeneous Neumann interface conditions to cancel the boundary term. The gradients are then approximated with central differences, giving:

$$\begin{aligned}\int_{\partial C_{i,j}\cap\Omega_-} \bigtriangledown q_{i,j} \cdot \boldsymbol{n} dL &= \frac{q_{i+\frac{1}{2},j} - q_{i,j}}{h} L_{i+\frac{1}{2},j} + \frac{q_{i,j+\frac{1}{2}} - q_{i,j}}{h} L_{i,j+\frac{1}{2}} \\ &+ \frac{q_{i-\frac{1}{2},j} - q_{i,j}}{h} L_{i-\frac{1}{2},j} + \frac{q_{i,j-\frac{1}{2}} - q_{i,j}}{h} L_{i,j-\frac{1}{2}},\end{aligned}$$

where $L_{i,j}$ are the lengths fractions of the cell edges inside the domain as depicted in figure 5.3. We approximate the surface and length integrals with the geometric procedures detailed in [64, 65]. The discretization produces a linear system of equations for q^{s+1}. We note that, for efficient implementation, M_1 and M_2 need to be computed only once for a given mesh and can then be used throughout the solution process for q.

5.6 Results and Discussion

5.6.1 Typical Results of the Shape Optimization Algorithm

In this section, we describe in details typical results of the shape optimization algorithm. Figure 5.6 ((a)-(d)) depicts the evolution of the template's shape and the corresponding diblock copolymer's density. The black circles describe the target centers' location, while the blue + symbols describe the computed centers' location, demonstrating that the shape optimization procedure is capable of converging to a template's shape that can direct the self-assembly to a target design.

Figure 5.6 ((e)) illustrates the two stages of Algorithm 7: in the first stage, only the pressure force is relaxed while in the second stage both the pressure and exchange forces are relaxed. The pressure force is depicted in red for the first stage and pink for the second stage, while the exchange force is depicted in blue for the first stage and in green for the second stage. In figure 5.6 ((f)), the intensive energy (H/V) is shown to decrease during both stages (in blue for the first stage and green for the second stage) and figures 5.6 ((g), (h)) (in blue for the first stage [shape optimization] and green for the second stage) demonstrates that the average velocity is negative most of the shape optimization: the zero-level-set that initially surrounds the target shrinks wrap the desired template's shape. In the later stage of the shape optimization process, the average velocity becomes positive before decreasing to a pre-described threshold, a manifestation of the adjustment

of the zero-level-set in the later stage of the shape optimization process (see figure 5.6 ((k))).

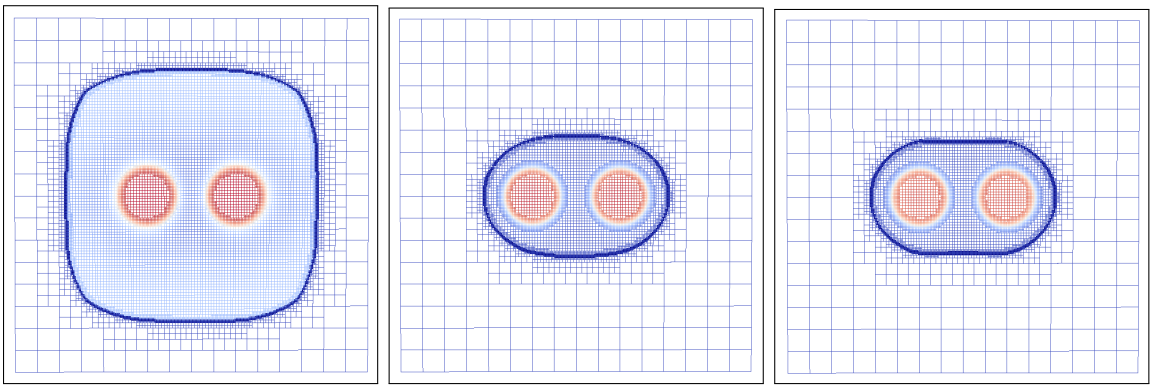
In the beginning of the shape relaxation process, both the absolute value of the average velocity and the standard deviation increase in magnitude. Indeed, even though we relax the shape, w_+ is set to zero and builds up in the first stage of the optimization process in tandem with the shape evolution. Ultimately, the shape evolution overcomes the pressure built-up and both the average velocity and the standard deviation of the velocity decrease in magnitude until they get lower than a predefined threshold.

Figure 5.7 illustrates the behavior of the correction algorithm. Figure 5.7 ((a)) gives the initial square template's shape and the corresponding diblock copolymer's density. Figure 5.7 ((b)) depicts the template's shape and the corresponding density along with both the target and computed centers' location obtained after the optimization process (both the pressure and exchange fields are updated). Figure 5.7 ((b)) illustrates that there is a slight mismatch between the location of the target's centers and the computed centers. To remedy this, we iteratively update the target's centers with equation (5.18) and repeat the optimization procedure until convergence. The result of this procedure is given in figure 5.7 ((c)) and the shape correction is depicted in 5.7 ((d)).

In the case where the target's features are close together, their boundary layers overlap and the exact $w_-(\boldsymbol{r})$ is important to describe their interaction. In our algorithm we fix in the first stage $w_-(\boldsymbol{r})$ and solve for $w_+(\boldsymbol{r})$ and $\phi(\boldsymbol{r})$ and then in the second stage, we fix $\phi(\boldsymbol{r})$ and solve for $w_-(\boldsymbol{r})$ and $w_+(\boldsymbol{r})$. That is, we guess $w_-(\boldsymbol{r})$ in the first

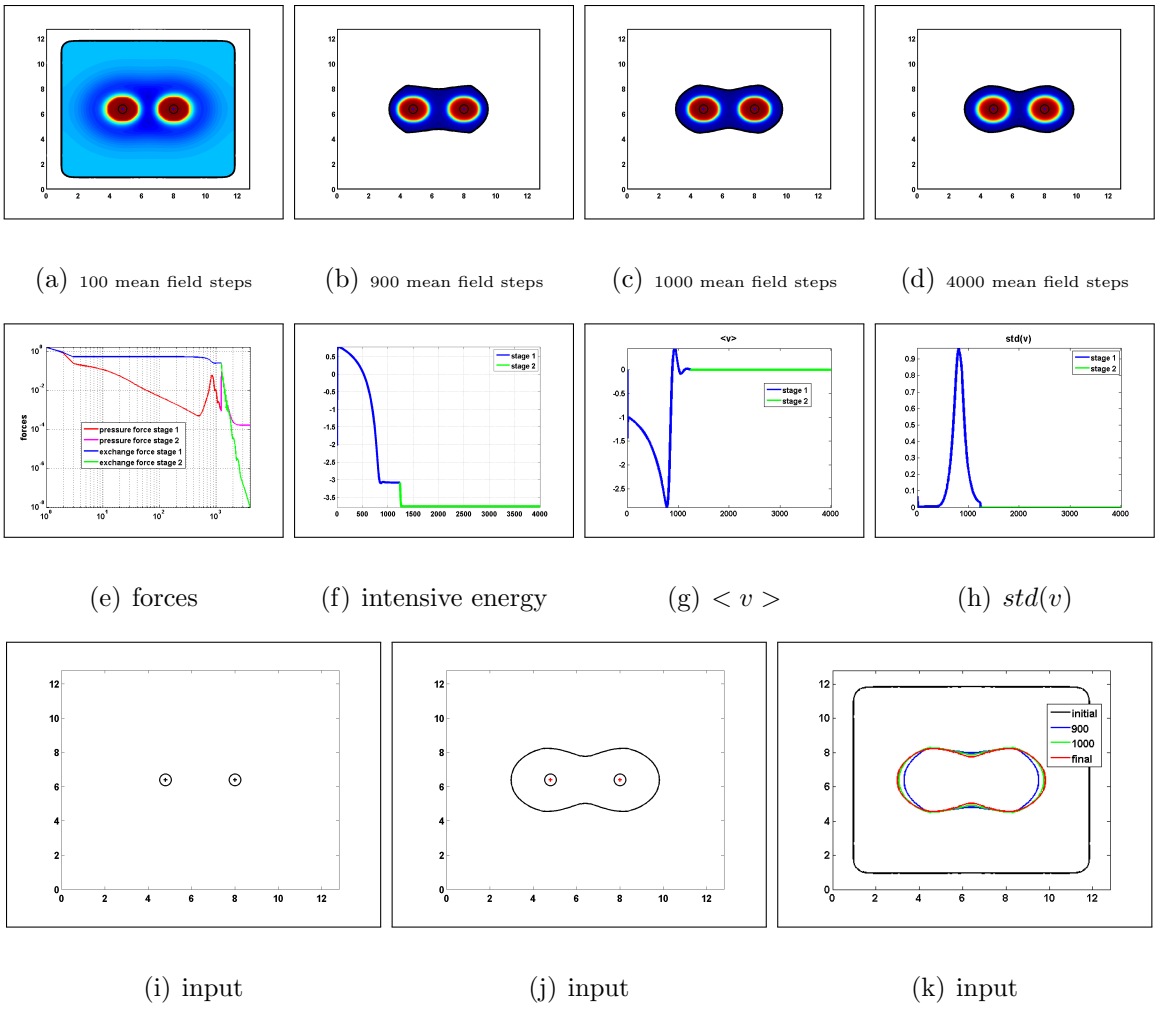
stage as either a binary value or a hyperbolic tangent profile but obviously $w_{-}(\boldsymbol{r})$ solved in the second stage may be different than the one guessed, which may result in a slight mismatch between the target centers and the computed centers. To remedy this issue we used a correction stage introduced previously in the outer loop by correcting the desired target up to the mismatch obtained as in equation (5.18). In figures 5.8((a)-(d)) we see that the optimization with correction stage based on the level-set based circle detection algorithm is successful for different topologies such as three aligned circles and three circles arranged in an isosceles triangle.

The examples of this section are instructive and typical of the inverse DSA problem. We first note that even for simple target shapes (here two cylinders), it is difficult find a template's shape that will guide the self-assembly to the target with the exact desired center-to-center's distance. This remark is also clearly communicated in [54, 53].



(a) initial stage of the grid's evolving shape (b) grid's evolving shape (c) final stage of the grid's evolving shape

Figure 5.5: Shape Optimization: the quadtree grid enables the refinement of regions of interest; here the domain's boundary and the interface between the minority region A and the majority region B.

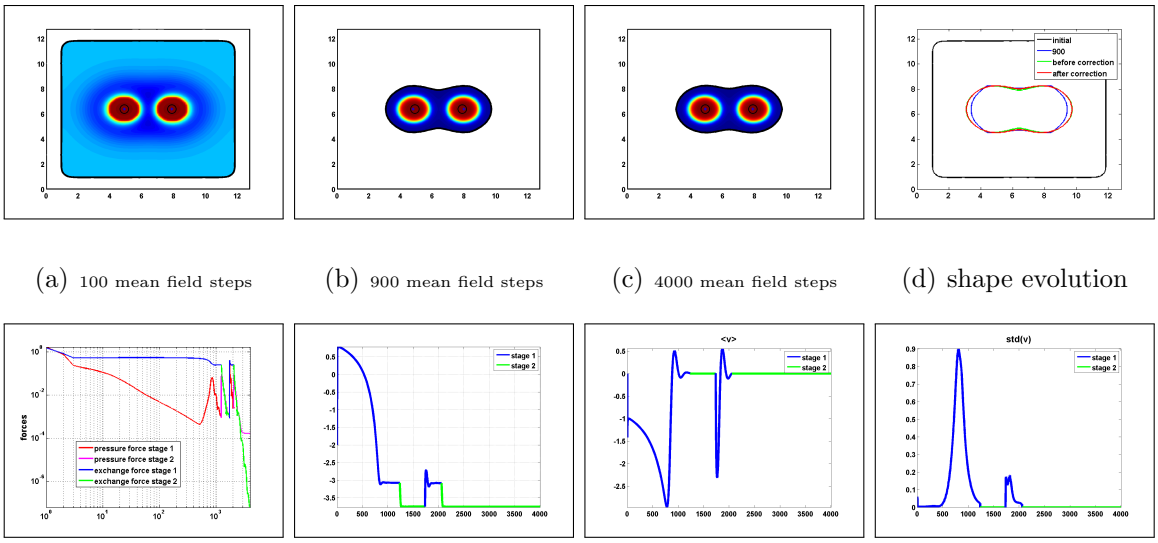


(a) 100 mean field steps (b) 900 mean field steps (c) 1000 mean field steps (d) 4000 mean field steps

(e) forces (f) intensive energy (g) $< v >$ (h) $std(v)$

(i) input (j) input (k) input

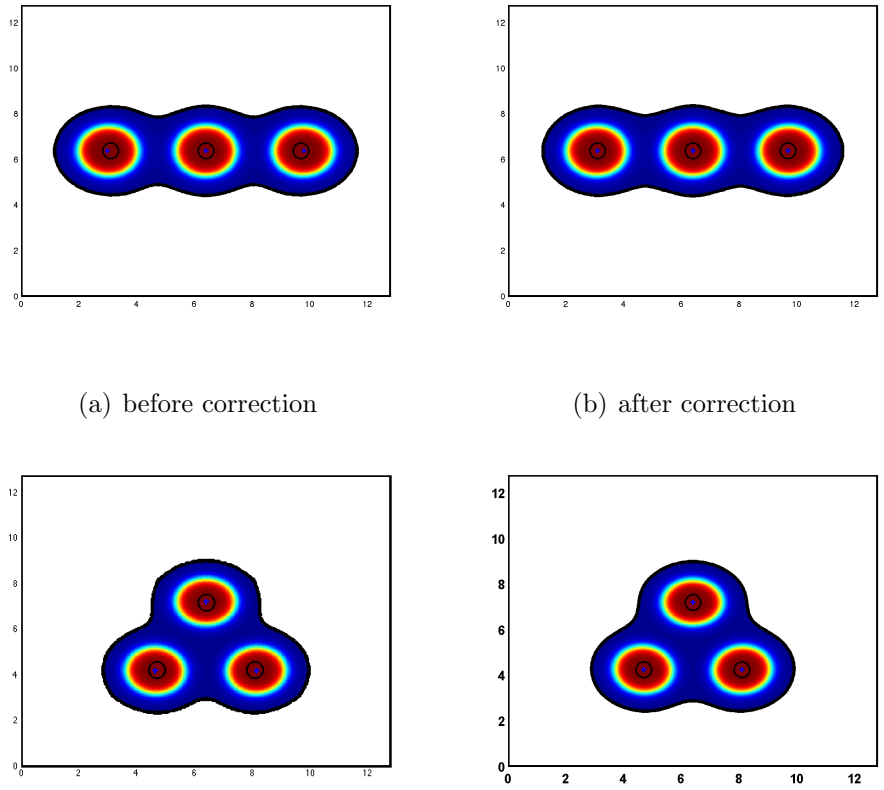
Figure 5.6: Results of the shape optimization algorithm without correction. First row: the shape and density evolution. Second row: the pressure and exchange forces and the intensive energy. Third row: the average velocity and the standard deviation of the velocity. Bottom row: input design and the optimal shape obtained with the computed centers location. The parameters for this simulation are $(f_A, \chi_{AB}) = (0.3, 36)$, $\Delta r = 3.2R_g$, $\lambda = 1$, $\Delta s = 0.01$, $\kappa_{cr} = 2/R_g$ and the grid levels are 4-7.



(a) 100 mean field steps (b) 900 mean field steps (c) 4000 mean field steps (d) shape evolution

(e) forces (f) intensive energy (g) $< v >$ (h) $std(v)$

Figure 5.7: Results of the shape optimization algorithm with correction. Top row: the shape and density evolution. Bottom row: the pressure and exchange forces, the intensive energy, the average velocity and the standard deviation of the velocity illustrate the two cycles of the optimization algorithm. The parameters for this simulation are $(f_A, \chi_{AB}) = (0.3, 36)$, $\Delta r = 3.0 R_g$, $\lambda = 1$, $\Delta s = 0.01$, $\kappa_{cr} = 2/R_g$ and the grid levels are 4-7.



(a) before correction

(b) after correction

(c) before correction

(d) after correction

Figure 5.8: Examples of the shape optimization with and without correction for three circles. First row: before and after correction for three aligned circles. Second row: before and after correction for three circles in an isosceles arrangement. The parameters for these simulations are $(f_A, \chi_{AB}) = (0.3, 36)$, $\Delta r = 3.3R_g$, $\Delta s = 0.01$, $\lambda = 1$, $\kappa_{cr} = 2/R_g$ with grid levels of 4-7 for the three aligned circles and $(f_A, \chi_{AB}) = (0.3, 36)$, $\Delta r = 3.4R_g$, $\Delta s = 0.01$, $\lambda = 1$, $\kappa_{cr} = 2/R_g$ with grid levels of 4-8 for the isosceles arrangement.

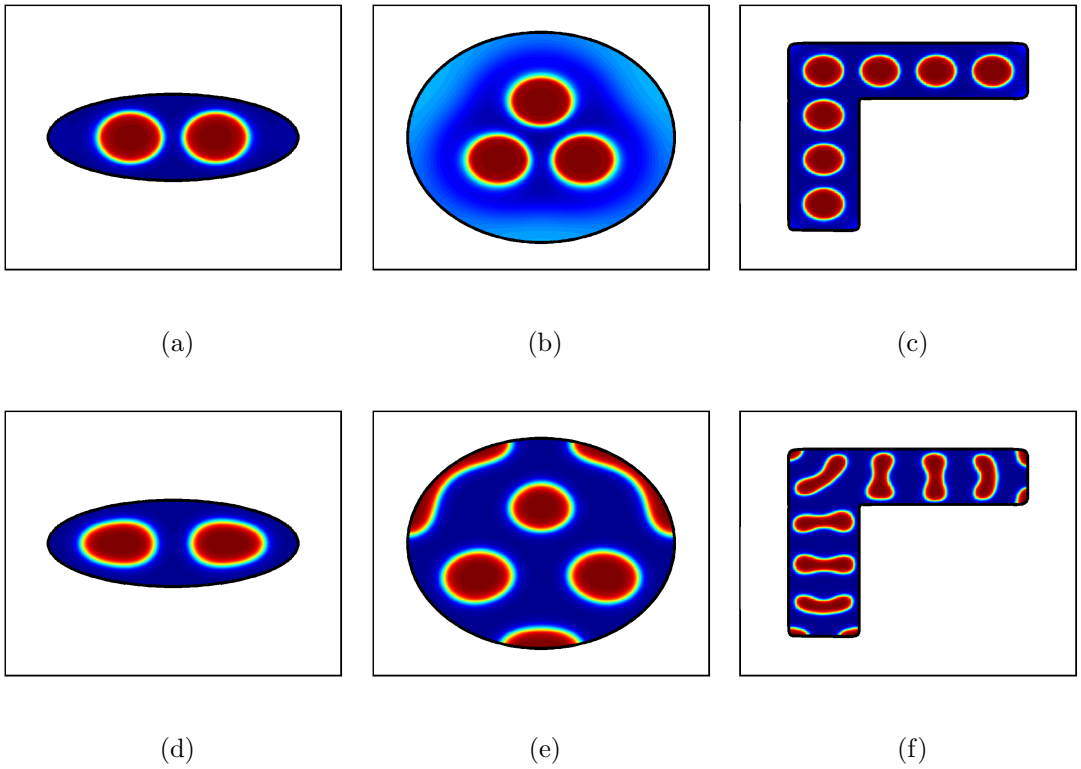
5.6.2 Stability

Recall that the goal of inverse DSA is to find the shape of a template that will direct the self-assembly to a target design and that our algorithm has two stages: the first stage freezes $w_-(\boldsymbol{r})$ and evolves $w_+(\boldsymbol{r})$ and $\phi(\boldsymbol{r})$, i.e. finds the template's shape; the second stage performs a standard SCFT simulation with the template's shape $\phi(\boldsymbol{r})$ obtained in the first stage. The second stage is therefore the simulation of self-assembly given a fixed template. Since the target can be expressed by the density of the minority component, one can in principle start a SCFT simulation of the self-assembly process by seeding the corresponding $w_-(\boldsymbol{r})$ to that of the target. However, in general the self-assembly will evolve the morphology to a different configuration due to the influence of the template's shape. We will say that a configuration is *locally stable* in the case where $w_-(\boldsymbol{r})$ does not change when evolving the SCFT algorithm after initializing $w_-(\boldsymbol{r})$ with its target value. Now, a template that produces local stability may not be enough to ensure that the self-assembly will converge to the target design from any initial seed. We will say that the configuration is *stable* in the case where the target design is obtained when evolving the SCFT algorithm from an arbitrary $w_-(\boldsymbol{r})$ seed. While we are in general interested in stability, a first step towards this goal is to find a shape that preserves local stability.

Local Stability

In figure 5.9, we present examples that illustrate that local stability is not straightforward in the case where the template's shape is not that obtained from our algorithm.

In those SCFT simulations, we first freeze the exchange potential and let only the pressure potential evolve before we evolve both the exchange potential and the pressure potential. This initialization thus gives the most favorable conditions for local stability. However, the initial states depicted in figures 5.9 ((a)-(c)) and their corresponding final states in figures 5.9 ((d)-(f)), demonstrate that local stability is not preserved. For example, figures 5.9 ((a), (d)) show that the two initial circles stretch asymmetrically in the x-direction. Figures 5.9 ((b), (e)) and ((c), (f)) illustrate the lack of local stability even more drastically: the initial circular shapes morph into non-circular shapes with centers of mass that are different from those of the initial shapes and the minority region now wets the walls of the confined domain. In these cases, the original morphology is completely lost.



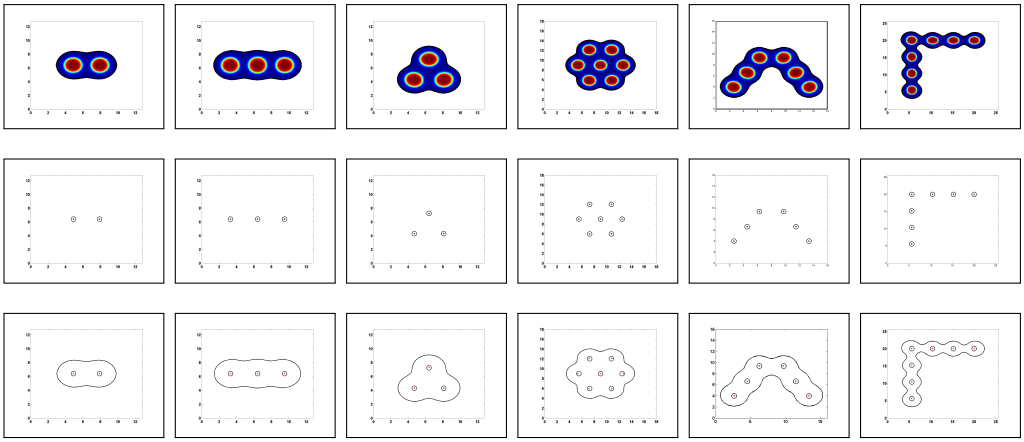
(a) (b) (c)

(d) (e) (f)

Figure 5.9: Examples showing local instability: a density enforced in the first stage of algorithm 7 (top row) is not preserved in the second stage (bottom row). ((a),(d)) $(\Delta r, L_x) = (2.8, 14)R_g$,((b),(e)) $(\Delta r, L_x) = (3.5, 14)R_g$, (((c),(f))) $(\Delta r,\ L_x) = (5.33, 32)R_g$. The parameters for these simulations are $f_A = 0.3$, $\chi_{AB} = 36$, $\Delta s = 0.01$, $\lambda = 1$ and the grid levels are 4-7.

In the previous examples, local stability is not attained due to the inappropriate templates' shapes. We now demonstrate that the shape optimization algorithm introduced in this paper produces templates' shapes that are locally stable. In figure 5.10, we demonstrate local stability by considering different design targets, finding the templates' shapes using Algorithm 7 and using a SCFT simulation, where both $w_-(\boldsymbol{r})$ and $w_+(\boldsymbol{r})$ are evolved. We used our circle detection algorithm to quantify for the mismatch

between the desired design and the one obtained by the optimization process. In figure 5.10 the second row depicts the target and the bottom row depicts the output of our shape optimization. These results show that Algorithm 7 produces templates' shapes that provide local stability, even in the case of non-trivial topologies.

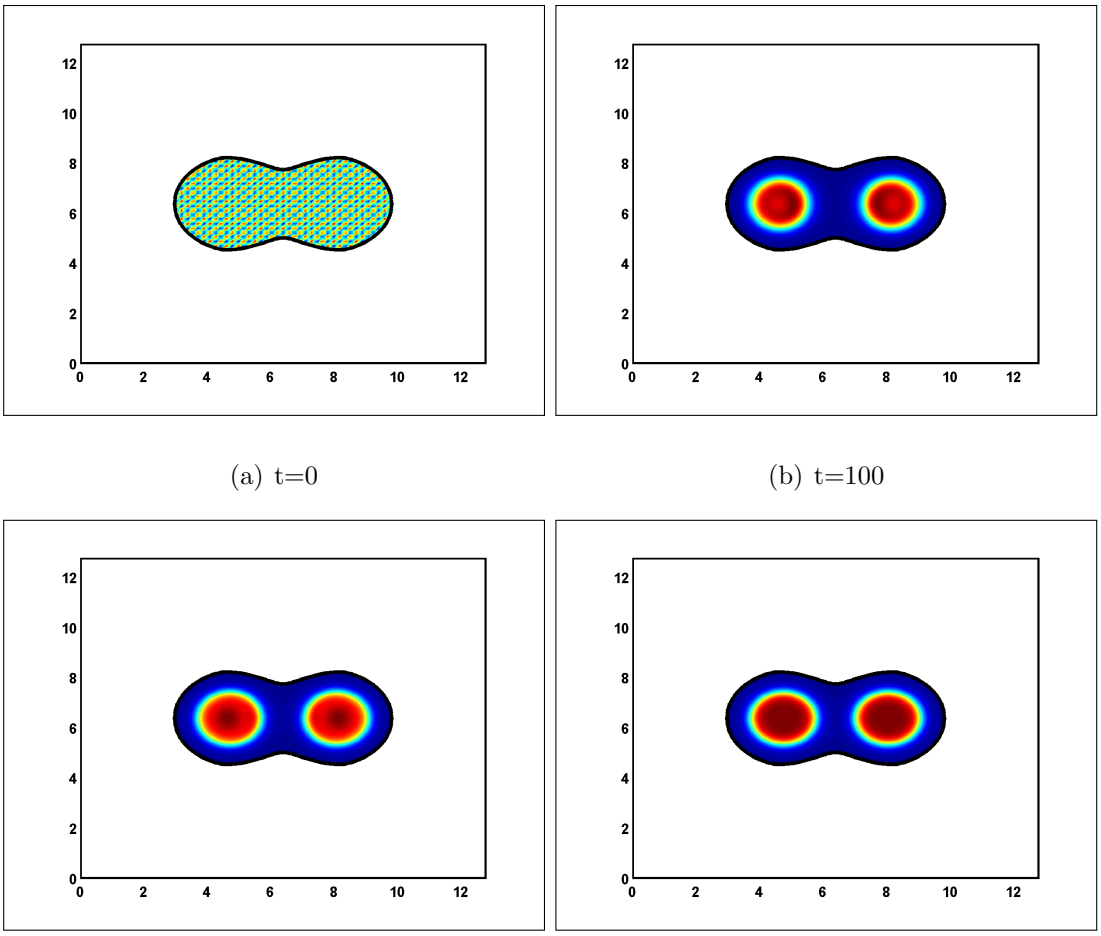


(m) (n) (o) (p) (q) (r)

Figure 5.10: Different topologies obtained with the shape optimization algorithm. Top row: densities inside the optimal shape, second row: input target, bottom row: output of the shape optimization algorithm where the circles depict the input target centers and the + depict the actual centers obtained with the optimal design. ((m)) two circles $(f_A, \chi_{AB}) = (0.3, 36)$ and $\Delta r = 3R_g$ ((n)) three aligned circles $(f_A, \chi_{AB}) = (0.3, 36)$ and $\Delta r = 3R_g$, ((o)) three circles in a isosceles $(f_A, \chi_{AB}) = (0.3, 36)$ and $\Delta r = 3.4R_g$, ((p)) star shape $(f_A, \chi_{AB}) = (0.28, 28)$ and $\Delta r = 3.5R_g$ ((q)) V-shape $(f_A, \chi_{AB}) = (0.28, 36)$ and $(\Delta r_{top}, \Delta r_{diagonal}) = (3.55, 3.2)$ ((r)) L-Shape $(f_A, \chi_{AB}) = (0.28, 36)$ and $\Delta r = 4.8R_g$. In all the examples, the parameters are $\Delta s = 0.01$, $\lambda = 1$, $\kappa_{cr} = 2/R_g$ and the grid levels are 4-7.

Stability

In the simulations shown in figures 5.8-5.10, we fixed the exchange field at the beginning of a simulation and we optimized the shape with stage 1 of Algorithm 7 such that when we start to evolve the exchange field in stage 2 of Algorithm 1 the morphology is kept. That is, we were looking for a shape to obtain *local stability* in the morphology density target. An important question is what happens when the optimal shape obtained from Algorithm 1, $\phi_{opt}(\boldsymbol{r})$, is used with an arbitrary seed $w_{-}^{arb}(\boldsymbol{r})$. In practice, this stage uses a standard SCFT optimization with the mask model (see Algorithm 9) where both the pressure and exchange fields are relaxed. That is, we want to check if the shape found by Algorithm 7 can be used to obtain stability.



(a) t=0 (b) t=100

(c) t=200 (d) t=2000

Figure 5.11: Example showing a SCFT optimization from an arbitrary configuration (top-left) using the mask model, with a template shape obtained with the method introduced in this manuscript. The parameters for this simulation are $(\chi_{wa}, \chi_{wb}) = \chi_{AB}(1, -1)$, $(f_A, \chi_{AB}) = (0.3, 36)$, $\Delta r = 3.2Rg$, $\Delta s = 0.01$, $\lambda = 1$, $\kappa_{cr} = 2/R_g$ with grid levels of 4-7.

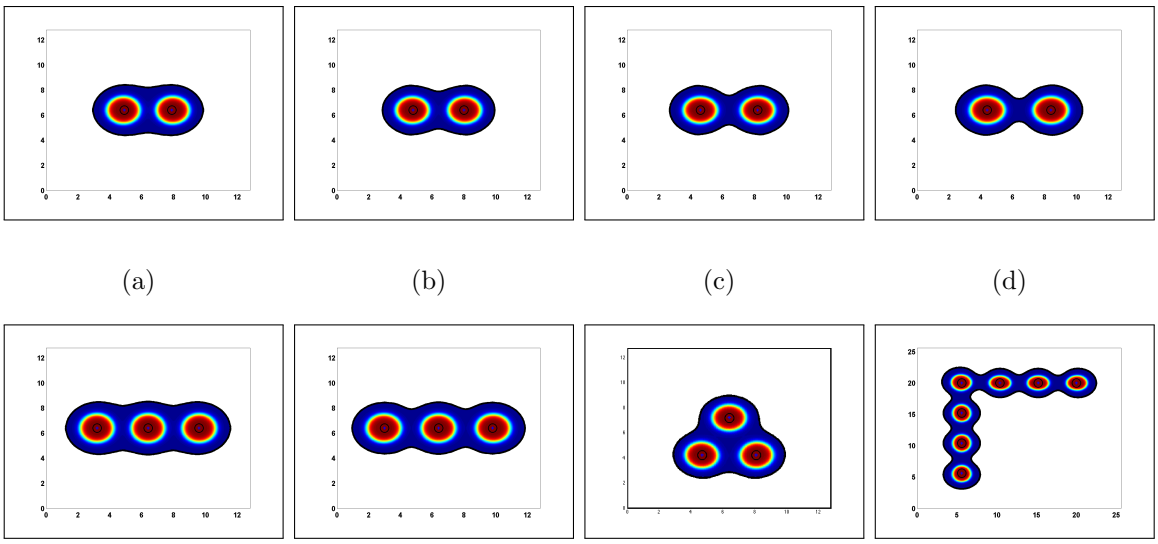
Although we are only considering a few cases and thus cannot theoretically guarantee the accuracy of Algorithm 7 for every target design, there is strong indication that the algorithm performs well for a wide range of targets. Figures 5.11 and 5.12 give examples

where we run Algorithm 9 with $\phi_{opt}(\boldsymbol{r})$ in equation (5.11) given by the shapes obtained with the inverse design Algorithm 7, and an arbitrary seed uncorrelated to the target. To obtain stability for the desired target, we vary the wall interaction with the polymer segments A and B to prevent A to wet the wall as in figure 5.9 ((e), (f)). This can be achieved by using either Robin boundary conditions [24] or with the mask model described in appendix A. We present the results with the mask model since it is the standard model used in this field but we note that we obtained the same results with Robin boundary conditions. We take $\frac{\chi_{w_a}}{\chi_{AB}} = O(1)$ and $\frac{\chi_{w_b}}{\chi_{AB}} = O(1)$ to obtain the results shown in this section. Figures 5.11 and 5.12 give examples of the results obtained from an arbitrary seed:

$$w_{-}^{arb}(\boldsymbol{r}) = \frac{\chi_{AB}}{2}\cos(36\frac{2\pi x}{L_x})\sin(36\frac{2\pi y}{L_y}) \quad \text{and} \quad w_{+}(t=0,\boldsymbol{r}) = 0.$$

We show in figure 5.11 a detailed example of the evolution of the density as a function of the mean field step. We find that the density converges to the target density, demonstrating stability.

We now turn our attention to different configurations, noting that we have checked that all the configuration in figures 5.10 lead to stability. It is important to recall that during self-assembly, a natural distance between cylinders is selected so that, if the distance between the target's cylinders is larger than the natural distance, the template's shape needs to prevent them from approaching one another. This is done by increasing the shape's concavity and thus limiting the region between the two cylinders to a thin gap. On the other hand, if the distance targeted is close to the natural distance there



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.12: Stability: Mean field optimization from an arbitrary seed using the mask model, with a template shape obtained with the method introduced in this manuscript: densities for different distances and topologies ((a)-(d)), two circles with $\Delta r = (3.0, 3.2, 3.6, 4)R_g$, ((e),(f)) three aligned circles with $\Delta r = (3.2, 3.4)R_g$, ((g)) three circles in an isosceles triangle with $\Delta r = 3.4R_g$ and grid levels of 4-8, ((h)), L-shape with $\Delta r = 4.8R_g$, $f_A = 0.28$. The parameters for these simulations are $(f_A, \chi_{AB}) = (0.3, 36)$, $(\chi_{wa}, \chi_{wb}) = \chi_{AB}(1, -1)$, $\Delta s = 0.01$, $\lambda = 1$, $\kappa_{cr} = 2/R_g$ with grid levels of 4-7.

Algorithm 9 Mask model SCFT with arbitrary seed and a fixed shape

Use $\phi_{opt}(\boldsymbol{r})$ from Algorithm 7 for (5.11), seed $w_-(t = 0, \boldsymbol{r}) = w_-^{arb}(\boldsymbol{r})$, evolve $w_-(\boldsymbol{r}), w_+(\boldsymbol{r})$ while $\phi_{opt}(\boldsymbol{r})$ is fixed

while $||\delta H/\delta w||_2 > \epsilon_w$ **do**

1. Solve two Fokker-Planck equations $q(s, \boldsymbol{r}), q^\dagger(s, \boldsymbol{r})$ using (5.3) ;
2. Compute the densities $\rho_A(\boldsymbol{r}), \rho_B(\boldsymbol{r})$ via (5.12);
3. Compute the forces for $w_+(\boldsymbol{r}), w_-(\boldsymbol{r})$ with equations (5.13), (5.14) ;
4. Advance the potentials $w_+(\boldsymbol{r}), w_-(\boldsymbol{r})$.

is no need for this constraint and the algorithm finds a slight concavity. In figures 5.12 ((a)-(d)), we see that by increasing the center-to-center distance, the template's shape presents a thinner gap between the circles to compensate for their natural propensity to be closer one another. Figure 5.12 (top row) indeed illustrates that the region between two cylinders is thin in the case where the distance between the location of target's cylinder is larger than the natural distance.

Finally to describe the effect of the curvature constraint in more details, we present in figure 5.13 a series of shapes obtained with algorithm 5 for two circles with $\Delta r = 4R_g$, and different curvature constraints. We can see that the constraint on the curvature prevents the concavity in the middle of the shape but compensates on the flanks and

push a bit on the edges. Eventually, constraining further will prevent the shape to rearrange and we would not have been able to preserve the design.

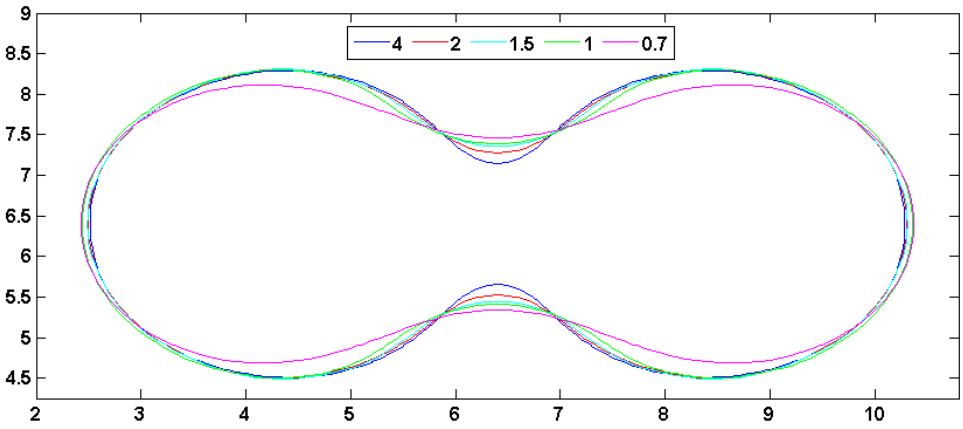
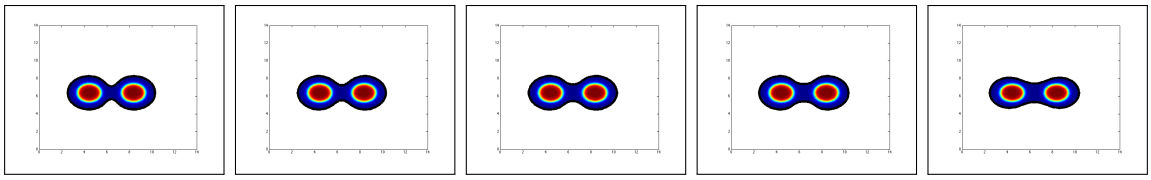


Figure 5.13: Effect of the curvature constrain κ_{cr}. For two aligned circles with a center-to-center distance $\Delta_r = 4R_g$, the optimal shape obtained is depicted for different curvature constraints. In the concavity zone located in the middle of the shape, we can see the effect of the curvature barrier. On the left and right flanks and the edges, we can see the shape rearrangement and the compensation effects. The parameters for these simulations are $(f_A, \chi_{AB}) = (0.3, 36)$, $R = 1.00R_g$, $\lambda = 1$, $\Delta s = 0.01$, $\Delta x = 0.1R_g$.



(a) $\kappa_{cr} = 4R_g^{-1}$ (b) $\kappa_{cr} = 2R_g^{-1}$ (c) $\kappa_{cr} = 1.5R_g^{-1}$ (d) $\kappa_{cr} = 1R_g^{-1}$ (e) $\kappa_{cr} = 0.7R_g^{-1}$

Figure 5.14: Effect of the curvature constrain κ_{cr}. For two aligned circles with a center-to-center distance $\Delta r = 4R_g$, the optimal shape obtained is depicted for different curvature constraints. In the concavity zone located in the middle of the shape, we can see the effect of the curvature barrier. On the left and right flanks and the edges, we can see the shape rearrangement and the compensation effects. The parameters for these simulations are $(f_A, \chi_{AB}) = (0.3, 36)$, $R = 1.00R_g$, $\lambda = 1$, $\Delta s = 0.01$, $\Delta x = 0.1R_g$.

5.7 Conclusion

We have presented a level-set approach to solve the inverse design problem for DSA applied to lithography using the shape as a free variable of the design. The algorithm evolves the shape with a normal velocity equal to w_+ while freezing the exchange potential until the shape reaches a steady-state. We demonstrated that we can achieve stability from an arbitrary seed for different designs. We also presented a level-set based circle detection algorithm that can be used in a correction stage that iteratively adjusts the distance between the centers of the desired morphology and perform the shape optimization until the mismatch between the optimal design and the target is negligible. Future research will explore optimizing the parameter space $(f_A, \chi_{AB}, \chi_{wa}, \chi_{wb})$ in tan-

dem with the template's shape to obtain global stability in an automated way for a given design.

Bibliography

[1] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set mehtod. *Journal of Computational Physics*, 2003.

[2] T Aslam. A partial differential equation approach to multidimensional extrapolation. *J. Comput. Phys.*, 193:349–355, 2004.

[3] T.J. Atherton and D.J. Kerbyson. Size invariant circle detection. *Image and Vision Computing*, 17(11):795 – 803, 1999.

[4] Satish Balay, Jed Brown, , Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. *PETSc Users Manual*. Argonne National Laboratory, 2012.

[5] J. Bang, U. Jeong, D. Y. Ryu, T. P. Russell, and C. J. Hawker. Block copolymer nanolithography: translation of molecular level control to nanoscale patterns. *Advanced Materials*, 21(47):4769–4792, 2009.

[6] Jean-Louis Barrat, Glenn H. Fredrickson, and Scott W. Sides. Introducing variable cell shape methods in field theory simulations of polymers. *The Journal of Physical Chemistry B*, 109(14):6694–6700, 2005. PMID: 16851752.

[7] Frank S. Bates, Marc A. Hillmyer, Timothy P. Lodge, Christopher M. Bates, Kris T. Delaney, and Glenn H. Fredrickson. Multiblock polymers: Panacea or pandora box? *Science*, 336(6080):434–440, 2012.

[8] A. W. Bosse, C. J. García-Cervera, and G. H. Fredrickson. Microdomain ordering in laterally confined block copolymer thin films. *Macromolecules*, 40(26):9570–9581, 2007.

[9] A. W. Bosse, S. W. Sides, K. Katsov, C. J. García-Cervera, and G. H. Fredrickson. Defects and their removal in block copolymer thin film simulations. *Journal of Polymer Science Part B: Polymer Physics*, 44(18):2495–2511, 2006.

[10] M. Burger and S. Osher. A survey on level set methods for inverse problems and optimal design. In *CAM report (04-02), (in press)*, 2004.

[11] Carsten Burstedde, Lucas C. Wilcox, and Omar Ghattas. `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.

[12] Jean Cea. Conception optimale ou identificatión de formes, calcul rapide de la dérivée directionnelle de la fonction coût. *ESAIM: Mathematical Modelling and*

Numerical Analysis - Modélisation Mathématique et Analyse Numérique, 20(3):371–402, 1986.

[13] Tony F. Chan and Xue-Cheng Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *Journal of Computational Physics*, 193(1):40 – 66, 2004.

[14] Jae-Byum Chang, Hong Kyoon Choi, Adam F. Hannon, Alfredo Alexander-Katz, Caroline A. Ross, and Karl K. Berggren. Design rules for self-assembled block copolymer patterns using tiled templates. *Nat Commun*, 5, 02 2014.

[15] Tanya L. Chantawansri, Su-Mi Hur, Carlos J. García-Cervera, Hector D. Ceniceros, and Glenn H. Fredrickson. Spectral collocation methods for polymer brushes. *The Journal of Chemical Physics*, 134(24):–, 2011.

[16] J. Y. Cheng, C. A. Ross, E. L. Thomas, H. I. Smith, and G. J. Vansco. Fabrication of nanostructures with long-range order using block copolymer lithography. *Appl. Phys. Lett.*, 81:3657–3659, 2002.

[17] J. Y. Cheng, D. P. Sanders, H. D. Truong, S. Harrer, A. Friz, S. Holmes, M. Colburn, and W. D. Hinsberg. Simple and versatile methods to integrate directed self-assembly with optical lithography using a polarity-switched photoresist. *ACS nano*, 4(8):4815–4823, 2010.

[18] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[19] Len Czuba. 2 - application of plastics in medical devices and equipment. In Kayvon ModjarradSina Ebnesajjad, editor, *Handbook of Polymer Applications in Medicine and Medical Devices*, Plastics Design Library, pages 9 – 19. William Andrew Publishing, Oxford, 2014.

[20] E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[21] P.-G De Gennes. A rule of sums for semidilute polymer chains near a wall. *C. R. Seances Acad. Sci, Ser. B*, 290(23):509–510, 1980.

[22] F. A. Detcheverry, D. Q. Pike, P. F. Nealey, M. Müller, and J. J. de Pablo. Monte carlo simulation of coarse grain polymeric systems. *Physical Review Letters*, 102(19):197801–197804, 2009.

[23] J. Finders, M. Dusa, B. Vleeming, B. Hepp, M. Maenhoudt, S. Cheng, and T. Vandeweyer. Double patterning lithography for 32nm: critical dimensions uniformity and overlay control considerations. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 8(1):011002–011002, 2009.

[24] G. H. Fredrickson. *The equilibrium theory of inhomogeneous polymers*, volume 134. Oxford University Press, USA, 2006.

[25] Matteo Frigo and Steven G Johnson. The fastest fourier transform in the west. Technical report, DTIC Document, 1997.

[26] Matteo Frigo and Steven G Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

[27] K. Galatsis, K. L. Wang, M. Ozkan, C. S. Ozkan, Y. Huang, J. P. Chang, H. G. Monbouquette, Y. Chen, P. F. Nealey, and Y. Botros. Patterning and templating for nanoelectronics. *Advanced Materials*, 22(6):769–778, 2010.

[28] Li Gan, Shun Han, Jinqiu Shen, Jiabi Zhu, Chunliu Zhu, Xinxin Zhang, and Yong Gan. Self-assembled liquid crystalline nanoparticles as a novel ophthalmic delivery system for dexamethasone: Improving preocular retention and ocular bioavailability. *International Journal of Pharmaceutics*, 396(1–2):179 – 187, 2010.

[29] Gopal Garg, Shailendra Saraf, and Swarnlata Saraf. Cubosomes: An overview. *Biological and Pharmaceutical Bulletin*, 30(2):350–353, 2007.

[30] F. Gibou and R. Fedkiw. Fast hybrid k-means level set algorithm for segmentation. Technical report, Stanford, 2002. Also in the proceeding of the 4^{th} International Conf. on Stat., Math. and Related Fields, Honolulu (2005).

[31] F. Gibou, R. Fedkiw, R. Caflisch, and S. Osher. A level set approach for the numerical simulation of dendritic growth. *J. Sci. Comput.*, 19:183–199, 2003.

[32] Sharon C. Glotzer and Michael J. Solomon. Anisotropy of building blocks and their assembly into complex structures. *Nat Mater*, 6(7):557–562, 08 2007.

[33] Adam F. Hannon, Yi Ding, Wubin Bai, Caroline A. Ross, and Alfredo Alexander-Katz. Optimizing topographical templates for directed self-assembly of block copolymers via inverse design simulations. *Nano Letters*, 14(1):318–325, 2014.

[34] Adam F. Hannon, Kevin W. Gotrik, Caroline A. Ross, and Alfredo Alexander-Katz. Inverse design of topographical templates for directed self-assembly of block copolymers. *ACS Macro Letters*, 2(3):251–255, 2013.

[35] Eugene Helfand. Block copolymer theory. iii. statistical mechanics of the microdomain structure. *Macromolecules*, 8(4):552–556, 1975.

[36] Eugene Helfand. Theory of inhomogeneous polymers: Fundamentals of the gaussian random walk model. *The Journal of Chemical Physics*, 62(3):999–1005, 1975.

[37] D. J. C. Herr. Directed block copolymer self-assembly for nanoelectronics fabrication. *Journal of Materials Research*, 26(02):122–139, 2011.

[38] Hough. P.v.c. method and means for recognizing complex patterns. U.S. Patent 3,069,654,, Dec. Dec. 18, 1962.

[39] S.-M. Hur, C. J. García-Cervera, E. J. Kramer, and G. H. Fredrickson. Scft simulations of thin film blends of block copolymer and homopolymer laterally confined in a square well. *Macromolecules*, 42(15):5861–5872, 2009.

[40] Tatsuhiro Iwama, Nabil Laachi, Bongkeun Kim, Michael Carilli, Kris T. Delaney, and Glenn H. Fredrickson. Computational study of directed self-assembly in neutral prepatterns for a graphoepitaxial pitch-multiplication application. *Macromolecules*, 48(4):1256–1261, 2015.

[41] Hadamard J. *Mémoires sur le Problème d'Analyse Relatif à l'Équilibre des Plaques Élastiques Encastrées*, volume 908 of *Oeuvre de J. Hadamard*. C.N.R.S., Paris, 1908.

[42] J.Strain. Tree methods for moving interfaces. *J. Comput. Phys*, 1999.

[43] B. Kim, N. Laachi, K. T. Delaney, and G. H. Fredrickson. Directed self-assembly of diblock copolymers in laterally confining channels: Effect of rough surfaces on line-edge roughness (ler) and defectivity. *Proc. of SPIE Advanced Lithography*, pages 90491D–90491D, 2014.

[44] Bongkeun Kim, Nabil Laachi, Kris T. Delaney, Michael Carilli, Edward J. Kramer, and Glenn H. Fredrickson. Thermodynamic and kinetic aspects of defectivity in directed self-assembly of cylinder-forming diblock copolymers in laterally confining thin channels. *Journal of Applied Polymer Science*, 131(24):n/a–n/a, 2014.

[45] S. O. Kim, H. H. Solak, M. P. Stoykovich, N.J. Ferrier, J. J. De Pablo, and P. F. Nealey. Epitaxial self-assembly of block copolymers on lithographically defined nanopatterned substrates. *Nature*, 424(6947):411–414, 2003.

[46] Yunju La, Chiyoung Park, Tae Joo Shin, Sang Hoon Joo, Sebyung Kang, and Kyoung Taek Kim. Colloidal inverse bicontinuous cubic membranes of block copolymers with tunable surface functional groups. *Nat Chem*, 6(6):534–541, 06 2014.

[47] N. Laachi, K. T. Delaney, B. Kim, S.-M. Hur, R. Bristol, D. Shykind, C. J. Weinheimer, and G. H. Fredrickson. The hole shrink problem: Theoretical studies of directed self-assembly in cylindrical confinement. *Proc. of SPIE Advanced Lithography*, page 868014, 2013.

[48] N. Laachi, K. T. Delaney, B. Kim, S.-M. Hur, R. Bristol, D. Shykind, C. J. Weinheimer, and G. H. Fredrickson. Self-consistent field theory investigation of directed self-assembly in cylindrical confinement. *J. Polym. Sci. Pol. Phys.*, 2014.

[49] N. Laachi, T. Iwama, K. T. Delaney, B. Kim, R. Bristol, D. Shykind, C. J. Weinheimer, and G. H. Fredrickson. Field-theoretic simulations of directed self-assembly in cylindrical confinement: Placement and rectification aspects. *Proc. of SPIE Advanced Lithography*, pages 9049–1M, 2014.

[50] N. Laachi, H. Takahashi, K. T. Delaney, S.-M. Hur, D. Shykind, C. J. Weinheimer, and G. H. Fredrickson. Self-consistent field theory of directed self-assembly in laterally confined lamellae-forming diblock copolymers. *Proc. of SPIE Advanced Lithography*, page 83230K, 2012.

[51] Nabil Laachi, Kris T. Delaney, Bongkeun Kim, Su-Mi Hur, Robert Bristol, David Shykind, Corey J. Weinheimer, and Glenn H. Fredrickson. Self-consistent field

theory investigation of directed self-assembly in cylindrical confinement. *Journal of Polymer Science Part B: Polymer Physics*, 53(2):142–153, 2015.

[52] Nabil Laachi, Tatsuhiro Iwama, Kris T. Delaney, David Shykind, Corey J. Weinheimer, and Glenn H. Fredrickson. Directed self-assembly of linear arrays of block copolymer cylinders. *Journal of Polymer Science Part B: Polymer Physics*, 53(5):317–326, 2015.

[53] Azat Latypov. Computational solution of inverse directed self-assembly problem, 2013.

[54] Azat Latypov, Grant Garner, Moshe Preil, Gerard Schmid, Wei-Long Wang, Ji Xu, and Yi Zou. Computational simulations and parametric studies for directed self-assembly process development and solution of the inverse directed self-assembly problem. *Japanese Journal of Applied Physics*, 53(6S):06JC01, 2014.

[55] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–136, 1982.

[56] Xingkun Man, David Andelman, and Henri Orland. Block copolymer films with free interfaces: Ordering by nanopatterned substrates. *Phys. Rev. E*, 86:010801, Jul 2012.

[57] M W Matsen. The standard gaussian model for block copolymer melts. *Journal of Physics: Condensed Matter*, 14(2):R21, 2002.

[58] M. W. Matsen and M. Schick. Stable and unstable phases of a diblock copolymer melt. *Phys. Rev. Lett.*, 72:2660–2663, Apr 1994.

[59] Mark W Matsen and FS Bates. Origins of complex self-assembly in block copolymers. *Macromolecules*, 29(23):7641–7644, 1996.

[60] MW Matsen. Phase behavior of block copolymer/homopolymer blends. *Macromolecules*, 28(17):5765–5773, 1995.

[61] MW Matsen. Stabilizing new morphologies by blending homopolymer with block copolymer. *Physical review letters*, 74(21):4225, 1995.

[62] Laurence W. McKeen. 3 - plastics used in medical devices. In Kayvon Modjarradsina Ebnesajjad, editor, *Handbook of Polymer Applications in Medicine and Medical Devices*, Plastics Design Library, pages 21 – 53. William Andrew Publishing, Oxford, 2014.

[63] Raffaele Mezzenga, Peter Schurtenberger, Adam Burbidge, and Martin Michel. Understanding foods as soft materials. *Nat Mater*, 4(10):729–740, 10 2005.

[64] C. Min and F. Gibou. Geometric integration over irregular domains with application to level set methods. *J. Comput. Phys.*, 226:1432–1443, 2007.

[65] C. Min and F. Gibou. Robust second-order accurate discretizations of the multidimensional Heaviside and Dirac delta functions. *J. Comput. Phys.*, 227:9686–9695, 2008.

[66] C. Min, F. Gibou, and H. Ceniceros. A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids. *J. Comput. Phys.*, 218:123–140, 2006.

[67] Chohong Min and Frederic Gibou. A second-order-accurate level set method on non-graded adaptive Cartesian grids. *J. Comput. Phys.*, 225:300–321, 2007.

[68] Mohammad Mirzadeh, Arthur Guittet, Carsten Burstedde, and Frederic Gibou. Parallel level-set methods on adaptive tree-based grids. *J. Comp. Phys. (submitted)*, 2015.

[69] K. Naito, H. Hieda, M. Sakurai, Y. Kamata, and K. Asakawa. 2.5-inch disk patterned media prepared by an artificially assisted self-assembly method. *IEEE Trans. Magn.*, 38:1949–1951, 2002.

[70] Tri-Hung Nguyen, Tracey Hanley, Christopher J.H. Porter, and Ben J. Boyd. Nanostructured liquid crystalline particles provide long duration sustained-release effect for a poorly water soluble drug after oral administration. *Journal of Controlled Release*, 153(2):180 – 186, 2011.

[71] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.

[72] S. Osher and R. P. Fedkiw. Level Set Methods: An Overview and Some Recent Results. *Journal of Computational Physics*, 169:463–502, 2001.

[73] S. Osher and J.A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[74] Stanley Osher and Fadil Santosa. Level set methods for optimization problems involving geometry and constraints: frequen- cies of a two-density inhomogeneous drum. *J. Comp. Phys.*, 171:272–288, 2001.

[75] G. Y. Ouaknin, N. Laachi, D. Bochkov, K. Delaney, G. H. Fredrickson, and F. Gibou. Functional level-set derivatives for free boundary problems with polymer self consistent field theory. submitted, 2016.

[76] G. Y. Ouaknin, N. Laachi, K. Delaney, G. H. Fredrickson, and F. Gibou. Level set strategy for inverse dsa-lithography. submitted, 2016.

[77] G. Y. Ouaknin, N. Laachi, K. Delaney, G. H. Fredrickson, and F. Gibou. Self consistent field theory simulations of polymers on arbitrary domains. submitted, 2016.

[78] G. Y. Ouaknin, N. Laachi, K. Delaney, G. H. Fredrickson, and F. Gibou. Shape optimization for dsa. *SPIE Proceedings*, Advanced Lithography, 2016.

[79] Joseph Papac, Frederic Gibou, and Christian Ratsch. Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions. *Journal of Computational Physics*, 229(3):875–889, February 2010.

[80] Joseph Papac, Asdis Helgadottir, Christian Ratsch, and Frederic Gibou. A level set approach for diffusion and stefan-type problems with robin boundary conditions on quadtree/octree adaptive cartesian grids. *Journal of Computational Physics*, 233:241–261, 2013.

[81] K. Ø. Rasmussen and G. Kalosakas. Improved numerical algorithm for exploring block copolymer mesophases. *Journal of Polymer Science Part B: Polymer Physics*, 40(16):1777–1783, 2002.

[82] B. Rathsack, M. Somervell, J. Hooge, M. Muramatsu, K. Tanouchi, T. Kitano, E. Nishimura, K. Yatsuda, S. Nagahara, I. Hiroyuki, et al. Pattern scaling with directed self assembly through lithography and etch process integration. *Proc. of SPIE Advanced Lithography*, pages 83230B–83230B, 2012.

[83] R. A. Segalman. Patterning with block copolymer thin films. *Materials Science and Engineering: R: Reports*, 48(6):191–226, 2005.

[84] R. A. Segalman, H. Yokoyama, and E. J. Kramer. Graphoepitaxy of spherical domain block copolymer films. *Advanced Materials*, 13(15):1152–1155, 2001.

[85] Rachel A. Segalman. Directing self-assembly toward perfection. *Science*, 321(5891):919–920, 2008.

[86] J. A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, 1999. Cambridge.

[87] P. Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. *J. Sci. Comput.*, 19:439 – 456, 2003.

[88] J. Sokolowski and J.P. Zolesio. *Introduction to shape optimization: shape sensitivity analysis*. Springer series in computational mathematics. Springer-Verlag, 1992.

[89] Patrick T. Spicer. Progress in liquid crystalline dispersions: Cubosomes. *Current Opinion in Colloid Interface Science*, 10(5–6):274 – 279, 2005.

[90] P. Stasiak, J. D. McGraw, K. Dalnoki-Veress, and M. W. Matsen. Step edges in thin films of lamellar-forming diblock copolymer. *Macromolecules*, 45(23):9531–9538, 2012.

[91] G. Strang. On the Construction and Comparison of Difference Schemes. *SIAM Journal on Numerical Analysis*, 5:506–517, September 1968.

[92] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.

[93] H. Takahashi, N. Laachi, K. T. Delaney, S.-M. Hur, C. J. Weinheimer, D. Shykind, and G. H. Fredrickson. Defectivity in laterally confined lamella-forming diblock copolymers: Thermodynamic and kinetic aspects. *Macromolecules*, 45(15):6253–6265, 2012.

[94] Maxime Theillard, Landry Fokoua Djodom, Jean-Léopold Vié, and Frédéric Gibou. A second-order sharp numerical method for solving the linear elasticity equations on

irregular domains and adaptive grids – application to shape optimization. *Journal of Computational Physics*, In press, 2012.

[95] H. Yi, X.-Y. Bao, J. Zhang, C. Bencher, L.-W. Chang, X. Chen, R. Tiberio, J. Conway, H. Dai, Y. Chen, S. Mitra, and H.-S. P. Wong. Flexible control of block copolymer directed self-assembly using small, topographical templates: Potential lithography solution for integrated circuit contact hole patterning. *Advanced Materials*, 24(23):3107–3114, 2012.

[96] HK Yuen, J Princen, J Illingworth, and J Kittler. Comparative study of hough transform methods for circle finding. *Image and Vision Computing*, 8(1):71 – 77, 1990.

Appendix A

SCFT for a diblock copolymer melt

This appendix retraces the main steps required to derive the equations used in this document. It is a very short description of SCFT, and we recommend a reader who wants to learn about the field to read the excellent monograph of Fredrickson. The approach used to investigate polymer physics in this work is statistical mechanics. The polymeric fluid is modeled as a closed system where particles can not enter or exit but can exchange energy with a big system of temperature T. The partition function is defined to be:

$$Z = \sum exp(-\beta U_i)$$

where the summation is over all the possible states of the physical system, and β is the boltzman factor $\frac{1}{k_B T}$. The probability to be in a particular state i at equilibrium is given by:

$$P_i = \frac{1}{Z} exp(-\beta U_i)$$

and the average value of a physical observable f is:

$$< f >= \frac{1}{Z} \sum_i exp(-\beta U_i) f_i$$

The polymer chain is modelled as a continuous gaussian chain and the deformation energy is given by:

$$U_0[\boldsymbol{r}] = \frac{3\beta}{2b^2} \int_0^N ds |\frac{d\boldsymbol{r}}{ds}|^2,$$

where s goes from 0 to N, N is the number of monomers on the backbone of the chain, and b is the average bond distance between these monomers. To compute the partition function, the chain is cut to N_s statistical segments of length $1/N_s$ and the degrees of freedom are the positions $\boldsymbol{r}(s)$ along the continuous chain. Then the different states are enumerated as $\prod_{i=0}^{N_s} \int d\boldsymbol{r}_i$ then we take N_s to ∞ and define the functional integral:

$$\int D\boldsymbol{r} = \lim_{N_s \to \infty} \prod_{i=0}^{N_s} \int d\boldsymbol{r}_i$$

and then the partition function becomes

$$Z_0 = \int D\boldsymbol{r} exp(-\beta U_0[\boldsymbol{r}])$$

In the case of a diblock copolymer the chain is comprised of two connected components A and B, where f_A is the fraction of A monomers and $1 - f_A$ is the fraction of B monomers in the chain.

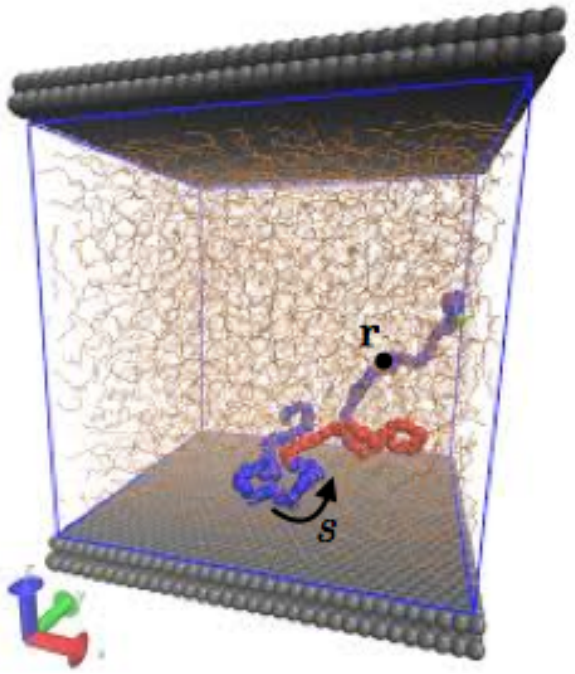


Figure A.1: Continuous diblock copolymer chain. s is the contour chain and $r(s)$ is the location of the chain for a given contour s. f_A is the fraction of A in the diblock copolymer chain. In our case we assume that the monomer bond distance b is equal for A and B.

We are interested in a physical system of n AB diblock copolymer chains at the molten state. In this case the energy comprises the deformation of the polymer chain for the short range interaction and for the energy interaction between the two different types we add a repulsion term s.t the equations who describe the system are given by:

$$U_0 = \frac{3\beta}{2b^2} \sum_{j=1}^{n} \int_0^N |\frac{d\boldsymbol{r}_j}{ds}|^2 ds$$

$$\beta U_1 = v_0 \chi_{AB} \int d\boldsymbol{r} \rho_A(\boldsymbol{r}) \rho_B(\boldsymbol{r})$$

where,

$$\rho_A(\boldsymbol{r}) = \sum_{j=1}^{n} \int_0^{f_A N} ds \delta(\boldsymbol{r} - \boldsymbol{r}_j(s)),$$

$$\rho_B(\boldsymbol{r}) = \sum_{j=1}^{n} \int_{f_A N}^{N} ds \delta(\boldsymbol{r} - \boldsymbol{r}_j(s)),$$

are the microscopic densities. In addition we have an incompressibility constrain $\rho_A(\boldsymbol{r}) + \rho_B(\boldsymbol{r}) = \rho_0(\boldsymbol{r})$. $v_0 = 1/\rho_0$ and $\boldsymbol{r}_j(s)$ is the space location of polymer chain j for a given contour chain s. The partition function is formulated by

$$Z = \frac{1}{n!(\lambda_T^3)^{nN}} \prod_{j=1}^{n} \int Dr_j e^{-\beta U_0 - \beta U_1} \times \delta[\rho_+(\boldsymbol{r}) - \rho_0(\boldsymbol{r})],$$

where $\lambda_T = h/(2\pi m\beta)^{1/2}$ is the thermal length with h the Planck constant, and m the mass of a statistical segment. We define $\rho_+(\boldsymbol{r}) = \rho_A(\boldsymbol{r}) + \rho_B(\boldsymbol{r})$ and $\rho_-(\boldsymbol{r}) = \rho_A(\boldsymbol{r}) - \rho_B(\boldsymbol{r})$, such that

$$e^{-\beta U_1} = e^{\frac{v_0 \chi_{AB}}{4} \int (\rho_+(\boldsymbol{r})^2 - \rho_-(\boldsymbol{r})^2) d\boldsymbol{r}}$$

We are going now to perform a Stratonovich transform: we are going to convert the partition function from a particles representation to a field representation. We define the delta functional:

$$\int D\rho\delta[\rho-\rho_0]F[\rho]=F[\rho_0]$$

for any functional $F[\rho]$. $\delta[\rho-\rho_0]$ can be viewed as an infinite dimensional version of the Dirac Delta function that vanishes unless the fields $\rho(\boldsymbol{r})$ and $\rho_0(\boldsymbol{r})$ are identical. A complex exponential representation of the δ functional using M_g grid points:

$$\delta[\rho-\rho_0]=\prod_r\delta(\rho(r)-\rho_0(r))=\frac{1}{(2\pi)^{M_g}}\prod^r\int_{-\infty}^{\infty}e^{iw(r)[\rho(r)-\rho_0(r)]}dw(r)=\int Dwe^{i\int d\boldsymbol{r}^n w(r)[\rho(r)-\rho_0(r)]}$$

where we used:

$$\delta(x)=\frac{1}{2\pi}\int_{\infty}^{\infty}dke^{ikx}$$

and the last equality comes from restoring the continuum and can be seen as changing the order of integration between r and w and defining functional integrals. So we can write:

$$\delta[\rho_0-\rho_+]=\int Dw_+e^{-i\int d\boldsymbol{r}(\rho_+-\rho_0)w_+},$$

Using functional Gaussian integration and Stratonovich transform it can be shown that:

$$\begin{aligned}e^{-\beta U_1}&=e^{-\frac{\chi_{AB}}{4}\int(\rho_+^2(\boldsymbol{r})-\rho_-^2(\boldsymbol{r}))d\boldsymbol{r}}\\&\propto\int Dw_-e^{\int d\boldsymbol{r}\left(\rho_-(\boldsymbol{r})w_--\frac{\rho_0}{\chi_{AB}}w_-^2\right)}\end{aligned}$$

and by decomposing U_0 to it's A and B part:

$$\beta U_0=\beta\int_0^{f_AN}|\frac{d\boldsymbol{r}(s)}{ds}|^2ds+\beta\int_{f_AN}^1|\frac{d\boldsymbol{r}(s)}{ds}|^2ds$$

factoring the different contributions of the Hamiltonian by the different densities we have $\rho_0(-iw_+ + \frac{w_-^2}{\chi_{AB}})$, $-\rho_- w_- = -\rho_A w_- + \rho_B w_-$ and $i\rho_+ w_+ = \rho_A i w_+ + \rho_B i w_+$ which gives for ρ_A and ρ_B: $\rho_A(iw_+ - w_-), \rho_B(iw_+ - w_-)$. Inserting these results into the Boltzmann factor of the partition function gives:

$$
\begin{aligned}
Z &= \frac{1}{n!(\lambda_T^3)^{nN}} \prod_{j=1}^{n} \int Dr_j e^{-\beta U_0 - \beta U_1} \times \delta[\rho_+ - \rho_0] \\
&\propto \int Dw_+ \int Dw_- \prod_{j=1}^{n} \int Dr_j e^{-\beta \int_0^{f_A N} ds |\frac{d\boldsymbol{r}}{ds}|^2} e^{-\beta \int d\boldsymbol{r} \rho_A (iw_+ - w_-)} \\
&\times e^{-\beta \int_{f_A N}^{N} ds \| \frac{d\boldsymbol{r}}{ds}|^2} e^{-\beta \int d\boldsymbol{r} \rho_B (iw_+ + w_-)} \times e^{-\int d\boldsymbol{r} \rho_0 (iw_+ - \frac{w_-^2}{\chi_{AB}})} \\
&= \int Dw_+ \int Dw_- \Big(\int Dr e^{-\beta \int ds \| \frac{d\boldsymbol{r}}{ds}|^2} e^{-\beta \int d\boldsymbol{r} \rho_A (iw_+ - w_-)} \Big)^n \\
&\times \Big(e^{-\beta \int_{f_A N}^{N} ds \| \frac{d\boldsymbol{r}}{ds}|^2} e^{-\beta \int d\boldsymbol{r} \rho_B (iw_+ + w_-)} \Big)^n \times e^{-\int d\boldsymbol{r} \rho_0 (iw_+ - \frac{w_-^2}{\chi_{AB}})} \\
&= \int Dw_+ \int Dw_- \Big(\int Dr e^{-\beta \int_0^{N} ds \| \frac{d\boldsymbol{r}}{ds}|^2} e^{-\beta \int d\boldsymbol{r} \rho(s) w(s)} \Big)^n \\
&\times e^{-\int d\boldsymbol{r} \rho_0 (iw_+ - \frac{w_-^2}{\chi_{AB}})} \\
&= \int Dw_+ \int Dw_- e^{n ln Q[w]} \times e^{-\int d\boldsymbol{r} \rho_0 (iw_+ - \frac{w_-^2}{\chi_{AB}})} \\
&= Z_0 \int Dw_+ \int Dw_- e^{-H[w_+, w_-]}
\end{aligned}
$$

where,

$$
H[w_+, w_-] = \rho_0 \int d\boldsymbol{r} \Big(- iw_+(\boldsymbol{r}) + \frac{w_-(\boldsymbol{r})^2}{\chi_{AB}} \Big) - n ln Q[w_+, w_-]
$$

and the potential $w(\boldsymbol{r}, s)$ is given by

$$
w(\boldsymbol{r}, s) = \begin{cases} w_A(r) = iw_+ - w_- & 0 < s < f_A N \\ w_B(r) = iw_+ + w_- & f_A N < s < f_A N \end{cases}
$$

and Q is defined as:

$$Q[w_+, w_-] = \int Dr e^{-\beta \int_0^N ds \| \frac{dr}{ds} |^2} e^{-\beta \int d\boldsymbol{r} \rho(s) w(s)}).$$

It remains to compute the functional $Q[w_+, w_-]$. We are going to show that in fact Q is the normalized partition function of a single diblock copolymer chain interacting with a potential $w(\boldsymbol{r}, s)$. We derive first the normalized partition function for an homopolymer continuous gaussian chain which interacts with an external potential $w(r)$ and then for a diblock copolymer chain. For an homopolymer chain interacting with a potential w the energy becomes:

$$U(\boldsymbol{r}^{N+1}) = U_0(\boldsymbol{r}^{N+1}) + U_w(\boldsymbol{r}^{N+1})$$

where,

$$U(\boldsymbol{r}^{N+1}) = U_0(\boldsymbol{r}^{N+1}) + U_w(\boldsymbol{r}^{N+1}) = \frac{3\beta}{2b^2} \int_0^N ds |\frac{d\boldsymbol{r}}{ds}|^2 + \beta \int d\boldsymbol{r} w(\boldsymbol{r}) \rho(\boldsymbol{r})$$

U_0 is the harmonic stretching energy of the gaussian chain and U_w accounts for the energy interaction of each bead with the potential field $\beta w(\boldsymbol{r})$. The potential energy associated with an imposed chemical potential field $w(\boldsymbol{r})$ is as well functional of the polymer shape $\boldsymbol{r}(s)$. The normalized partition function $Q[w]$ is expressed as a ratio of path integrals:

$$Q[w] = \frac{Z[w]}{Z_0} = \frac{\int Dr e^{-\beta U_0[r]} e^{-\beta U_1[r]}}{\int Dr e^{-\beta U_0[r]}},$$

The continuous gaussian chain is discretized to N bonds and $N+1$ beads

$$\int D\boldsymbol{r} \approx \prod_{i=0}^{N_s} \int d\boldsymbol{r}_i, \frac{dr}{ds} = \frac{\Delta r}{\Delta s},$$

such that

$$Z_0 = V(\int d\boldsymbol{b} exp(-\beta h(|\boldsymbol{b}|)))^N,$$

where $b_i = r_i - r_{i-1}$ and $h(x) = 3k_BTx^2/(2b^2)$. Defining a transition probability function:

$$p(\Delta \boldsymbol{r}) = \frac{exp(-\beta h(|\Delta \boldsymbol{r}|))}{\int d\boldsymbol{b} exp(-\beta h(|\boldsymbol{b}|))} = (\frac{3}{2\pi b^2 \Delta s})^{3/2} exp(\frac{-3|\Delta r|^2}{2b^2 \Delta s})$$

$$\begin{aligned} Q[w] &= \frac{\int d\boldsymbol{r}^{N+1} exp(-\beta U(\boldsymbol{r}^{N+1}))}{V(\int d\boldsymbol{b} exp(-\beta h(|\boldsymbol{b}|)))^N} \\ &= \frac{1}{V} \int e^{-w(\boldsymbol{r}_N)\Delta s} p(\boldsymbol{r}_N - \boldsymbol{r}_{N-1}) e^{-w(\boldsymbol{r}_{N-1})} p(\boldsymbol{r}_{N-1} - \boldsymbol{r}_{N-2}) ....^4 \\ &....e^{-w(\boldsymbol{r}_2)\Delta s} p(\boldsymbol{r}_2 - \boldsymbol{r}_2) e^{-w(\boldsymbol{r}_1)\Delta s} p(\boldsymbol{r}_1 - \boldsymbol{r}_0) e^{-w(\boldsymbol{r}_0)\Delta s} \end{aligned}$$

The partition function is defined in a recursive way by defining a functional q:

$$q(\boldsymbol{r}, s + \Delta s; [w]) = e^{(-\Delta s w(\boldsymbol{r}))} \int dr' p(r - r') q(r', s, [w])$$

$$q(\boldsymbol{r}, 0, [w]) = exp(-\Delta s w(\boldsymbol{r}))$$

$$Q[w] = \frac{1}{V} \int d\boldsymbol{r} q(\boldsymbol{r}, N; [w]))$$

Using Taylor series we get a partial differential equation for the functional q[w]:

$$\frac{\partial}{\partial s} q(\boldsymbol{r}, s; [w]) = \frac{b^2}{6} \bigtriangledown^2 q(\boldsymbol{r}, s; [w]) - wq$$

This PDE can be explained as the probability function to be at a certain location r for a contour s is governed by a random walk interacting with an external potential. And as initial condition $q(\boldsymbol{r}, 0, [w]) = 1$.

The continuous Gaussian chain in a potential $w(\boldsymbol{r})$ has a factorization property:

$$Q[w] = \frac{1}{V} \int d\boldsymbol{r} q(\boldsymbol{r}, N - s; [w]) q(\boldsymbol{r}, s, [w])$$

In the case of a diblock copolymer with total polymerization index of N:

$$type = \begin{cases} A & \text{if } 0 < s < f_A N \\ B & \text{if } f_A N < s < N \end{cases}$$

where f_A is the fraction of the copolymer that is type A. The stretching energy of such a diblock can be expressed as: $\beta U_0 = \int_0^N ds \frac{3}{2[b(s)]^2} |\frac{dr}{ds}|^2$, and the interaction with a potential $\beta U_1[w_A, w_B] = \int dr w_A(r)\rho_A(r) + w_B(r)\rho_B(r)$, where we abuse notation and the microscopic densities are now defined for a single chain only $\rho_A(\boldsymbol{r}) = \int_0^{f_A N} ds \delta(\boldsymbol{r} - \boldsymbol{r}_j(s))$, $\rho_B(\boldsymbol{r}) = \int_{f_A N}^N ds \delta(\boldsymbol{r} - \boldsymbol{r}_j(s))$.

In this this case we solve two Fokker-Planck equations: one for q when s starts from the A part and finishes in the B part and another for $q^\dagger$ when it starts from the B part and finishes in the A part.

$$\frac{\partial}{\partial s}q(\boldsymbol{r},s;[w_A,w_B]) = \frac{b^2}{6}\bigtriangledown^2 q(\boldsymbol{r},s;[w_A,w_B]) - wq$$

$$w(\boldsymbol{r},s) = \begin{cases} w_A(\boldsymbol{r}) & \text{if } 0 < s < f_A N \\ w_B(\boldsymbol{r}) & \text{if } f_A N < s < N \end{cases}$$

$$\frac{\partial}{\partial s}q^\dagger(\boldsymbol{r},s;[w_A,w_B]) = \frac{b^2}{6}\bigtriangledown^2 q^\dagger(\boldsymbol{r},s;[w_A,w_B]) - w^\dagger q^\dagger$$

$$w^\dagger(\boldsymbol{r},s) = \begin{cases} w_B(\boldsymbol{r}) & \text{if } 0 < s < 1 - f_A N \\ w_A(\boldsymbol{r}) & \text{if } 1 - f_A N < s < N \end{cases}$$

In this case the normalized partition function is written as:

$$\begin{aligned} Q[w_A,w_B] = \frac{z[w_A,w_B]}{z_0} &= \frac{\int e^{-\beta U_0[r]}e^{-\beta U_1[r]}}{\int Dr e^{-\beta U_0[r]}} \\ &= \frac{\int Dr e^{-\beta U_0[r]}e^{-\int dr w_A(r)\rho_A(r)+w_B(r)\rho_B(r)}}{\int Dr e^{-\beta U_0[r]}}, \end{aligned}$$

and it can be formulated as:

$$Q[w_A,w_B] = \frac{1}{V}\int dr q(r,N;[w_A,w_B]) = \frac{1}{V}\int dr q^\dagger(r,N;[w_A,w_B])$$

Now using that the density is given by:

$$< \rho(\boldsymbol{r}^{N+1}) >= \frac{\int dr^{N+1}\rho(\boldsymbol{r}^{N+1})exp(-\beta U(\boldsymbol{r}^{N+1}))}{\int dr^{N+1}exp(-\beta U(\boldsymbol{r}^{N+1}))}$$

we obtain:

$$\begin{cases} \rho_A(r;[w_A,w_B]) & = -\frac{1}{Q[w_A,w_B]}\frac{\delta Q[w_A,w_B]}{\delta w_A} \\ \rho_B(r;[w_A,w_B]) & = -\frac{1}{Q[w_A,w_B]}\frac{\delta Q[w_A,w_B]}{\delta w_B} \end{cases}$$

Finally taking the derivative from the physical expression:

$$\frac{\delta Q[w_A,w_B]}{\delta w_A(r)} = \frac{1}{V}\Delta s \times \sum_{j=0}^{f_A N}\int d\boldsymbol{r}^{N+1}[e^{-\Delta s w_B(\boldsymbol{r}_N)}\phi_B(r_N - r_{N-1})e^{-\Delta s w_B(\boldsymbol{r}_{N_1-1})}\phi_B(r_{N_1-1}-r_{N_1-2})$$

$$...e^{-\Delta s w_A(\boldsymbol{r}_{fN})}\phi_A(r_{fN}-r_{fN-1})$$

$$...e^{-w_A(r_j)\Delta s}(-1)\delta(r-r_j)....e^{-\Delta s w_A(\boldsymbol{r}_2)}\phi_A(r_2-r_1)e^{-\Delta s w_A(\boldsymbol{r}_1)}\phi(r_1-r_0)e^{-\Delta s w_A(\boldsymbol{r}_0)}]$$

$$= \frac{-1}{V}\int_0^{f_A N} q^\dagger(N-j,r,[w_A,w_B])q(j,r,[w_A,w_B])ds$$

Similarly for ρ_B

$$\frac{\delta Q[w_A,w_B]}{\delta w_B(r)} = \frac{-1}{V}\int_{f_A N}^{N} q^\dagger(N-j,r,[w_A,w_B])q(j,r,[w_A,w_B])ds$$

And for the polymer melt to compute the density we just multiply the obtained expressions for ρ_A and ρ_B by n the number of diblock copolymer chains.

Mean field approximation The partition function obtained is not easy to compute as it involves functional integrals over $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$ and it is practically impossible to perform these integrals numerically. At the mean field approximation we assume that a single field configuration $w_+^*(\boldsymbol{r}), w_-^*(\boldsymbol{r})$ dominates the functional integral. This field configuration is obtained by demanding that $H[w_+,w_-]$ be stationary with respect to variations in $w_+(\boldsymbol{r})$ and $w_-(\boldsymbol{r})$,

$$\frac{\delta H[w_+, w_-]}{\delta w_+(\boldsymbol{r})} = \frac{\delta H[w_+, w_-]}{\delta w_-(\boldsymbol{r})} = 0$$

The mean field approximation is pretty accurate for high mass and highly coordinated polymer chains, because the field fluctuations are small. In contrast for light polymer chains, since the chains can easily change position and so change the potential experienced on a particular statistical segment, the field fluctuations can't be neglected. In other words the mean field approximation can be used on physical systems where the field fluctuations can be neglected. Before finishing we remind again that this is a very short description of SCFT, and we recommend a reader who wants to learn about the field to read the excellent monograph of Fredrickson. Finally we note that the equations developed here have physical dimensions, while the equations used in the chapters of this thesis have been scaled and are dimensionless.