Chapter 3

Numerical method for order parameter dynamics

In this chapter, we describe several numerical methodologies for the integration of the tensorial nematic equation of motion obtained in the previous chapter. We believe that these methods are applied to the study of the dynamic behaviour of nematic liquid crystals for the first time.

We first illustrate the non-dimensionalization of the different parameters arising in the Landau-de Gennes theory, thus setting basic length and time scales for the problems addressed in this thesis. We describe a numerical method, known as the method of lines, for integrating the deterministic part of the equation without approximation, illustrating this method using a parabolic initial value problem. The numerical implementation makes it possible to address several long-standing problems without resort to approximations: the structure of the isotropic-nematic interface, the structure of nematic droplets in an isotropic environment and dynamical scaling in coarsening kinetics. These form the topics of later chapters.

We then present a highly accurate technique for discretizing the derivative operator appearing in the GLdG equation. This spectral collocation technique is implemented for the first time, to our knowledge, for the TDGL equations for nematics. We illustrate the underlying idea using the example of a scalar equation. The accuracy of the technique makes it possible to understand several subtle properties of the isotropic-nematic interface. It can be applied to several associated problems which have not yet been addressed due to lack of efficient numerical methods [57].

We calculate the two-point correlations in real and Fourier space for the general nematic problem, working with the full non-linear theory. We then describe how these techniques can be extended to high performance computation. We conclude this chapter with an illustration of a new visualization technique for scalar and tensor fields relevant to our problem.

3.1 Introduction

The equations for the relaxational dynamics of the orientational order parameter presented in the previous chapter are a set of five coupled and parabolic differential equations. These
Chapter 3. Numerical method for order parameter dynamics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniaxial order</td>
<td>( S = 3(1 + \sqrt{1 - 24AC/B^2})/4 )</td>
</tr>
<tr>
<td>Uniaxial order at coexistence</td>
<td>( S_c = -2B/9C )</td>
</tr>
<tr>
<td>Free energy at coexistence</td>
<td>( \mathcal{F}_c = 9CS_c^4/16 )</td>
</tr>
<tr>
<td>Correlation length</td>
<td>( l_c = \sqrt{54C(L_1 + 2L_2/3)/B^2} )</td>
</tr>
<tr>
<td>Relaxation time</td>
<td>( \tau = (\Gamma\mathcal{F}_c)^{-1} )</td>
</tr>
</tbody>
</table>

Table 3.1: The dimensional parameters used for non-dimensionalization the order parameters, free energy with length and time scale.

...are highly non-linear, contain anisotropic gradient terms and are coupled strongly to each other.

The problems we study are categorized into two classes. One ignores the effect of fluctuations and focuses on the solution of the five partial differential equations (PDE). The other class of problems incorporates the fluctuating noise term, thus making these equations stochastic differential equations. No analytical solutions are available for these equations in general in either of these cases. Efficient and accurate numerical methods must therefore be sought. Conventionally, the methods used to solve the deterministic equations are based on the standard finite difference technique [77]. The elegance in our method can be well justified in terms of computational efficiency, when extended to the stochastic problems, as discussed in the subsequent sections.

3.2 Non-dimensionalization of the parameters

We scale the order parameter as \( Q_{\alpha\beta}^* = Q_{\alpha\beta}/S_c \) and \( S^* = S/S_c \), while the free energy is scaled as \( \mathcal{F}^* = \mathcal{F}/\mathcal{F}_c \). Here \( \mathcal{F}_c \) is proportional to the free energy barrier between the isotropic and nematic minima at coexistence. We measure lengths in units of \( l_c \). Finally, time is non-dimensionalised by the characteristic relaxation time \( \tau \).

The dimensional quantities which we use to scale the parameters are defined in terms of the Landau parameters in the Table 3.1. We choose the discretization length \( \Delta x = \Delta y = \Delta z = 1 \) and the integration time step \( \Delta t = 1 \). These define simulation units of length and time. To ensure that discretization errors and artifacts are kept to a minimum, the discretization length must be much smaller than the characteristic length \( \Delta x = \Delta y = \Delta z \ll l_c \). The integration time step must be much smaller than the characteristic time scale \( \Delta t \ll \tau \). The dimensionless discretization scales must then satisfy \( \Delta x \to \Delta x/l_c \ll 1, \Delta y \to \Delta y/l_c \ll 1, \Delta z \to \Delta z/l_c \ll 1 \) and \( \Delta t \to \Delta t/\tau \ll 1 \).

Our simulations are performed maintaining the above conditions, on grid sizes ranging from \( 16^2 \) to \( 1024^2 \) in two dimensions and \( 32^3 \) to \( 256^3 \) in three dimensions.
3.3 Method of lines

In this section we present a method, based on the strategy of semi-discretization, whereby an initial value deterministic PDE is discretized only in the spatial variables to yield a set of coupled ordinary differential equations (ODE). These ODEs can be solved using powerful general purpose solvers. In the literature, this semi-discretization strategy goes by the name of the ‘method of lines’ (MOL) [44, 43], since the solutions are obtained along fixed lines in the space-time plane.

To illustrate the idea of MOL, consider a parabolic initial-value problem in a single spatial variable \( x \), the diffusion equation for a scalar field \( \psi(x,t) \) given as,

\[
\frac{\partial \psi(x,t)}{\partial t} = D \nabla^2 \psi(x,t). \tag{3.1}
\]

The semi-discretization technique proceeds by restricting the field \( \psi(x,t) \) to a set of \( N \) discrete ‘collocation’ points, \( x_n = n\Delta x, \, n = 1, 2, \ldots, N \), spaced \( \Delta x \) apart, and then uses a discrete approximation for the Laplacian based on these points. The simplest of these approximations is based on local polynomial interpolation, resulting in finite difference (FD) schemes [43]. However, the MOL is not restricted to finite differences. When high accuracy is needed, global interpolation based on trigonometric or Chebyshev polynomials can be used to generate spectral approximations of derivatives [70]. When conservation laws need to be respected, finite volume approximations to the derivatives can be used [42]. In complicated geometries, a finite-element discretization may be the most appropriate.

In this example, the simplest approximation is based on nearest-neighbour finite differences, for which

\[
\nabla^2 \psi(x_n) = \frac{1}{(\Delta x)^2}[\psi(x_{n+1}) - 2\psi(x_n) + \psi(x_{n-1})] + O[(\Delta x)^2]. \tag{3.2}
\]

Inserting this into the diffusion equation, we obtain a set of coupled ordinary differential equations

\[
\frac{\partial \psi(x_n,t)}{\partial t} = \frac{1}{(\Delta x)^2}[\psi(x_{n+1}) - 2\psi(x_n) + \psi(x_{n-1})]. \tag{3.3}
\]

This coupled set of ODEs, together with initial and boundary conditions, can now be integrated with a suitable numerical integration scheme which ensures accuracy, efficiency, and stability. It is at this stage that the flexibility of the MOL is most apparent, since any number of numerical integration schemes can be implemented without affecting the spatial discretization. Depending on the nature of the system of ODEs, the optimal choice may be either an explicit scheme, an implicit scheme, or one which is designed to handle stiffness. In the above example, for instance, it is well known that an explicit Euler integration scheme leads to an instability unless the time step \( \Delta t \) is constrained by the Courant-Friedrichs-Lewy condition \( \Delta t < (\Delta x)^2/2D \) [43]. In contrast, an implicit integration scheme based on the trapezoidal rule gives the stable Crank-Nicolson update [43]. In general, semi-discretization followed by numerical quadrature provides an elegant way of deriving many of the well-known schemes for parabolic PDEs.
The practical implementation of PDE solvers using the MOL discretization is simple since the spatially discretized ODE system can be passed directly to a general purpose ODE solver. The complexity, both algorithmic and computational, in the numerical integration can thereby be transferred directly to the ODE solver library.

We have followed the MOL discretization to construct a solver for PDEs describing the deterministic dynamics of the orientational order parameter presented in the previous chapter. We have used standard nearest-neighbour second-order accurate finite difference formula for first and second derivatives in the spatial discretization [1]. More accurate difference approximations to derivatives can be obtained using Fornberg’s general formula [29]. For the temporal integration, we have used the ODE solver routines in the GNU Scientific Library [27]. For the problems we study in this thesis, we find that an explicit multi-stage integrator gives a good compromise between accuracy, stability and computation expense.

Finally, it should be mentioned that the MOL discretization is not restricted to parabolic initial value problems, but also applies to hyperbolic and mixed parabolic - hyperbolic PDEs [44]. This allows the method to be extended to situations where effects of advection from hydrodynamic flow need to be accounted for. On an Intel(R) Core(TM)2 Duo CPU with speed 2.66GHz, the MOL code takes 0.33 seconds for one time step once a lattice size of $256^2$.

### 3.4 Spectral collocation technique

The simplest way of deriving the FD form of the derivative presented in Eq. (3.2) is to construct a local polynomial interpolant $P(x_n)$ of degree $\leq 2$ with $P(x_{n+1}) = \psi_{n+1}, P(x_n) = \psi_n$ and $P(x_{n-1}) = \psi_{n-1}$ and differentiating $P(x_n)$ and evaluating at $x = x_n$. The degree of the polynomial can be increased to yield a dense differentiation matrix that converges as $O[(\Delta x)^{\text{degree}}]$. In the spectral collocation method (SCM), the spatial part is discretized through an expansion in an orthogonal basis of Chebyshev polynomials in a bounded interval. The spatial discretization is done on the Chebyshev points $x_j = \cos(j\pi/N), j = 0, 1, \cdots, N$ which are the projections of equispaced points on the upper half periphery of a unit circle on the x-axis bounded by the interval $[-1, 1]$. This points which cluster mostly at the boundaries, constructed from Chebyshev polynomials, are used to construct the differentiation matrix. This is carried out by construction of a global polynomial interpolant $\leq N$ and differentiation and evaluation of the interpolant at the required space point. Differentiation operators constructed from this Chebyshev interpolant are spectrally accurate, in the sense that the error vanishes exponentially in the number of retained polynomials.

The implementation of boundary condition in SCM is fairly straightforward for the problems related either to a bounded interval or an open interval. For problems with periodic domain, trigonometric polynomial interpolant is constructed on equispaced grid, whereas, for problems with non-periodic domain, algebraic polynomial interpolant is constructed on unevenly spaced grid, one of the examples are the Chebyshev interpolant. Imposing either
3.4. Spectral collocation technique

Figure 3.1: Variation of the scalar magnetization $\psi$ across the planar interface. Symbols represent spectral data, solid curves are the analytic solution of Eq.(3.4). The inset shows the maximum error obtained doing a spectral calculation in the least square approximation. The numerical parameters are $\epsilon = 0.01$ and $N = 16, 32, 48$ and 64.

Dirichlet, Neumann or Robin boundary condition\(^1\) is easy to implement in SCM, which can either be implemented through restricting the interpolants to satisfy the boundary condition or by adding additional equations to enforce the boundary condition without restricting the interpolant.

3.4.1 Benchmark of Allen-Cahn equation

To illustrate the idea of SCM, consider a parabolic initial-value problem in a single spatial variable $z$: the one dimensional Allen-Cahn bistable equation for a scalar field $\psi(z, t)$ representing a nonlinear reaction-diffusion system given as

$$\partial_t \psi(z, t) = \psi(z, t) - \psi^3(z, t) + \epsilon \nabla^2 \psi(z, t), \quad (3.4)$$

whose solution has a profile $\psi(z, t) = \tanh(z/\xi)$ bounded between $\pm 1$ and $\xi = \sqrt{2\epsilon}$ denote the correlation length. The interpolant is constructed so as to satisfy Dirichlet boundary conditions. In Fig.(3.1), we plot the equilibrium profile of $\psi(z)$ with increasing number of Chebyshev polynomial $N$ from 16 to 64. In the inset, we show how the maximum error committed using the SCM goes down with $N$. The DMSUITE library is used for the numerical implementation of SCM \[74\].

\(^1\)The Dirichlet condition specifies field values at the boundary, the Neumann condition specifies the derivative of the field at the boundary while the Robin condition specifies both Dirichlet and Neumann conditions simultaneously at the boundary.
3.5 Stochastic method of lines

The fluctuating nematodynamic equations contained in Eq. (1.34) are five non-linear stochastic partial differential equations which can only be studied numerically. Here we combine the method of lines for solving initial-value PDEs presented in the previous section, with a stochastic version of the Runge-Kutta integrator for systems of stochastic ordinary differential equations. This enables us to construct an accurate and efficient solver for the equations of fluctuating nematodynamics. The results we present in this thesis were first described in [10], where a method of lines approach was used to solve the deterministic time-dependent Ginzburg-Landau equations numerically. The methodology here can thus be thought of as a generalization of the method of lines to stochastic PDEs. To apply the MOL to stochastic PDEs, we must account for the fact that integrators for ODEs do not automatically provide efficient and accurate solutions of stochastic differential equations (SDE). Qualitatively, the noise term in a SDE is a rapidly varying function and hence must be integrated with some care. At a more technical level, the noise is a Wiener process and the theory of stochastic integration must be used to evaluate it correctly [31].

Common stochastic integrators include those due to Maryuama [46] and Milstein [50]. In our work, we use an integrator proposed recently by Wilkie [75], based on a multi-step Runge-Kutta strategy. The integrator is accurate and easy to implement by making small changes to a deterministic Runge-Kutta integrator. Further, since it is an explicit integrator, no matrix inversions are involved. This makes it attractive when the MOL semi-discretization produces a large system of ODEs, as in our case.

3.5.1 Benchmark of Ornstein-Uhlenbeck process

To test Wilkie’s algorithm for a stochastic Runge-Kutta integrator (SRK4), and to check the validity of the fluctuation-dissipation relation, we perform a simple benchmark test on the Ornstein-Uhlenbeck process, which is the velocity of a Brownian particle represented as the Ito differential equation,

\[ dv(t) = -Γv(t)dt + dW(t), \]  

where \( dW(t) = \sqrt{2k_BTdT}N(0,1) \) is the increment of the stochastic variable in the interval \( dt \), and \( N(0,1) \) is a zero-mean unit-variance normal deviate. By construction, the increments of this stochastic variable are independent and normally distributed with mean \( \langle dW(t) \rangle = 0 \). The particular choice of the variance ensures that the equilibrium distribution of \( v \) is a Gaussian with variance \( k_BT \). The stationary two-point autocorrelation of the velocity from Eq. (3.5) is,

\[ \langle v(t)v(t+τ) \rangle = k_BT \exp(-Γτ). \]  

Fig. (3.2) shows the autocorrelation as a function of time and the histogram of equal-time fluctuations of \( v \). It is well known that statistical error committed due to a time integration over a single realization is error-prone with an uncertainty of 20% in order, which reduces by a factor of \( 1/\sqrt{N} \), where \( N \) is the number of independent realization of the sample [78]. The variance \( \langle v^2 \rangle = k_BT \) is correctly reproduced, as is the exponential decay of the
3.5. Stochastic method of lines

Figure 3.2: Autocorrelation function for the Ornstein-Uhlenbeck process, showing \( \langle v(t)v(t+\tau) \rangle \) as a function of the time increment \( \tau \). The inset shows the histogram of fluctuations. It is Gaussian with the expected variance. The numerical parameters chosen are \( \Gamma = k_B T = 0.1, dt = 1.0 \). The average is taken over 10 independent realizations while the integration is performed for \( 10^6 \) SRK4 steps.

We conclude that SRK4 is suitable as an integrator for problems where the fluctuation-dissipation relation must be maintained.

3.5.2 Discretization of fluctuating nematodynamics

We now apply the MOL together with SRK4 to obtain a stochastic method lines (SMOL) discretization for the equations of fluctuating nematodynamics. We benchmark our numerical results by comparing autocorrelations within a harmonic theory which accurately describes fluctuations about the isotropic phase. We then consider expansions about the ordered state, comparing static correlations obtained analytically within the Frank approximation with our numerical results.

We use a FD discretization with nearest-neighbour stencils for gradients and the Laplacian. We implement periodic boundary conditions. Specifically in three dimensions, we consider a box of dimension \( L_x, L_y \) and \( L_z \) along the Cartesian directions, and grid these lengths with equal grid spacing \( \Delta x = \Delta y = \Delta z = 1 \). The latter defines lattice units for the spatial coordinate. We define corresponding discrete time units for the temporal variables by choosing \( \Delta t = 1 \). Fourier modes are labeled by the wave-vector \( \mathbf{q} = (q_x, q_y, q_z) \), where each component is of the form \( q_\alpha = 2\pi n_\alpha / L_\alpha \), with \( n_\alpha = 0, 1, 2, \ldots, (L_\alpha - 1) \).

With this discretization, the Laplacian in Fourier space is given by

\[
\mathcal{L}(\mathbf{q}) = 2[\cos(q_x) + \cos(q_y) + \cos(q_z) - 3].
\] (3.7)

The nearest-neighbour finite difference stencil suffers from lack of isotropy at high wavenumbers. This can be improved through the use of higher-point stencils \([1, 54, 65]\). Applying the
Chapter 3. Numerical method for order parameter dynamics

MOL semi-discretization to Eq. (1.34) reduces it to a system of stochastic ordinary differential equations, whose Fourier representation in the harmonic approximation of Eq. (2.12) is

\[ \partial_t a_i(q, t) = -\Gamma D a_i(q, t) + \xi_i(q, t). \]  

(3.8)

The Fourier representation of the drift-diffusion dynamics is encoded in the linear operator \( D \),

\[ D = A - L_1 \mathcal{L}(q). \]  

(3.9)

Fourier representations of the one and two-dimensional MOL semi-discretizations are obtained by setting the corresponding wavenumbers to zero.

3.5.3 Correlations in the linear theory

The static and dynamic correlations in Fourier space follow in a straightforward manner through the replacement of \( q^2 \) by its discrete Laplacian representation. The results are

\[ C_{ij}(q) = k_B T \frac{D}{\mathcal{D}} \delta_{ij}, \]  

(3.10)

\[ C_{ij}(q, \tau) = C_{ij}(q) \exp(-\Gamma \mathcal{D} \tau). \]  

(3.11)

It is also useful to define an angle-averaged structure factor \( C(q) = \sum_{|q|=q} C_{ij}(q) \) for comparison with the numerical simulation.

We now compare theoretical and numerical results: in Fig. (3.3) we show the histogram of the \( a_i(q) \) for a particular Fourier mode. This is normally distributed, as expected, with zero
Figure 3.4: (a) Contour plot of the structure factor $C_{ij}(q)$ and (b) angular average of $C_{ij}(q)$. The parameters are $k_B T = A = 0.05, L_1 = 0.5, \Gamma = 1.0$. The time averaging is over $5 \times 10^4$ time steps and ensemble averaging is over 20 independent realizations in a box with $L_x = L_y = 64$. The relaxation time scale for the shortest Fourier mode is $\tau = (\Gamma A)^{-1} = 20$ and the correlation length is $\lambda = \sqrt{L_1/A} = 3.16$. 
Chapter 3. Numerical method for order parameter dynamics

Figure 3.5: Autocorrelation $C_{ij}(\mathbf{q}, \tau)$ for the linear Langevin equation, calculated for small wavenumbers. Numerical parameters are $k_B T = A = 0.0035$, $L_1 = 0.015$, $\Gamma = 1.0$. The integration is performed for $5 \times 10^4$ time steps on a $16^2$ grid. The time average is taken over $4 \times 10^4$ time steps and an ensemble average is taken over 25 independent realizations. Time scale for the longest relaxation of the shortest Fourier mode is $\tau = (\Gamma A)^{-1} = 285.7$, the fastest relaxation of the largest Fourier mode is $\tau = 8.1$. The correlation length $\lambda = \sqrt{L_1/A} = 2.1$.

mean and variance as required by thermal equilibrium. Similarly, all Fourier modes examined have correct normal distributions. The variances obtained are compared in Fig.[3.4(a)] with the analytical values by plotting contours of $C_{ij}(\mathbf{q})$. There appears to be excellent agreement.

A closer inspection reveals some degree of anisotropy in both the analytical and numerical results at high wavenumbers. This is attributed to the lack of isotropy of the nearest-neighbour finite-difference Laplacian mentioned earlier. However, the anisotropies are removed upon angular averaging, as shown in Fig.[3.4(b)]. Thus, the present discretization should be adequate in most cases, unless highly accurate isotropies are required from the simulation. From these results, we conclude that correlations in thermal equilibrium are accurately captured by the stochastic method of lines approach.

We next compare the dynamics of fluctuations at equilibrium, by comparing two-point autocorrelation functions calculated analytically and numerically. Fig.(3.5) shows $C_{ij}(\mathbf{q}, \tau)$ for three sets of Fourier modes. The exponential decay of the autocorrelation function is reproduced accurately within the numerics and fit the theoretical curve very closely. We conclude, therefore, that the stochastic method of lines accurately reproduces both static and dynamic fluctuations in a harmonic theory.

### 3.5.4 Correlations in the nonlinear theory

Finally, we compare theory and simulation in a situation where a linearization of the $Q_{\alpha\beta}$ equations about $Q_{\alpha\beta} = 0$ is inapplicable, that of director fluctuations within the nematic
Figure 3.6: Angularly averaged structure factor of the nematic phase. The numerical parameters are $k_B T = 0.05$, $A = -3.5$, $B = -10 k_B T$, $C = 2.67$, $L_1 = 32.0$, $\Gamma = 0.01$ on a $64^2$ grid. The relaxation time scale is $\tau = (\Gamma A)^{-1} = 28.57$, the diffusion time scale of the shortest Fourier mode is $\tau_d = L^2 \pi^2 / (4 \pi^2 L_1 \Gamma) = 324.23$ and the correlation length is $\lambda = \sqrt{L_1 / A} = 3.02$. The time average is taken over $5 \times 10^4$ time steps and an ensemble average is taken over 20 independent realizations.

In the semi discrete representation, Eq. (2.16) takes the form,

$$\langle \theta(q) \theta(-q) \rangle = -\frac{k_B T}{K \mathcal{L}(q)}.$$  \hspace{1cm} (3.12)

Fig. (3.6) shows the angular average of the static correlations of director fluctuations $C_\theta(q) = \sum_{|q|=q} \langle \theta(q) \theta(-q) \rangle$. The formally divergent $q = 0$ mode is excluded both from the numerical data and analytical result. Given that the analytical result is obtained from a linearization about the aligned state whereas the numerical solution is calculated from the equations of motion arising from the full non-linear free energy, the agreement between theory and simulation is satisfactory. The SMOL thus accurately captures equilibrium fluctuations in the ordered state as well as in the disordered one.

### 3.6 High performance computation

In principle, the nematic equations of motion can be numerically integrated on any platform. The time scale for the smallest Fourier mode in the problem to relax goes as $\tau \sim L^d$ where $d$ is the number of space dimensions and $L$ is the system size. However, there are occasions, such as the study of phase transitions, in which behaviour in the thermodynamic limit $L \to \infty$ must be studied, which then implies $\tau \to \infty$. Also, whenever multiple length and time scales coexist, a long time simulation of large system sizes must be performed.

A serial compute machine performs a time update by executing discrete sets of instructions on each lattice node in the subsequent clock time utilizing the CPU memory resulting
### Table 3.2: Scaling of the parallel code with the number of cores in a node for a 128\(^2\) lattice.

<table>
<thead>
<tr>
<th>Number of processors</th>
<th>CPU time (in secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000392</td>
</tr>
<tr>
<td>2</td>
<td>0.000229</td>
</tr>
<tr>
<td>6</td>
<td>0.000122</td>
</tr>
<tr>
<td>10</td>
<td>0.000095</td>
</tr>
<tr>
<td>14</td>
<td>0.000084</td>
</tr>
<tr>
<td>18</td>
<td>0.000086</td>
</tr>
<tr>
<td>22</td>
<td>0.000076</td>
</tr>
<tr>
<td>26</td>
<td>0.000075</td>
</tr>
<tr>
<td>30</td>
<td>0.000070</td>
</tr>
</tbody>
</table>

This increase with system size implies that serial computing cannot access very large system sizes within achievable computational time. This is, instead, achieved through the use of parallel clusters, where the lattice is domain-decomposed into variable sized sub-lattices which are distributed to the processing elements (PE) that contain local information about the sub-lattice. A haloed region of grid spacing \(\Delta x\) owned by the neighboring processors is used to communicate the data between the PE’s. Time updates are performed simultaneously on each compute machine of the cluster, thus reducing the computational time by a factor of \(1/N\) where \(N\) is the number of machines in the cluster.

To remove the instabilities arising due to the use of explicit time update discussed in the previous sections and to achieve convergence of the algorithm of any order in \(\Delta t\), an implicit time update have to be used. For this purpose, the Jacobian matrix \(J\) has to be inverted in each time update. In our algorithm, the Jacobian matrix has to be provided to the integrator to use this.

### 3.6.1 Benchmark of Allen-Cahn equation

To illustrate the idea, we simulate the dynamics of the Allen-Cahn equation, defined in Eq. (3.4) on a three dimensional lattice with linear dimension \(L_x, L_y\) and \(L_z\). We use a lexicographic projection to construct a single index from the lattice index, defined as \(m = i + jL_x + kL_xL_y\) where, \(i, j,\) and \(k\) varies from 0 to \(L_x - 1, L_y - 1\) and \(L_z - 1\) respectively. In the FD approximation, the equation reduces to the form,

\[
\partial_t \psi_m = \left[1 - \psi_m^2 - \frac{6\epsilon}{(\Delta x)^2}\right] \psi_m + \frac{\epsilon}{(\Delta x)^2} \left(\psi_{m+1} + \psi_{m-1} + \psi_{m+L_x} + \psi_{m-L_x} + \psi_{m+L_xL_y} + \psi_{m-L_xL_y}\right)
\]

and the Jacobian matrix, defined as \(J_{m;n} = \partial_t \psi_m / \partial \psi_n\) is,

\[
J_{m;n} = \left[1 - 3\psi_m^2 - \frac{6\epsilon}{(\Delta x)^2}\right] \delta_{m;n} + \frac{\epsilon}{(\Delta x)^2} (\delta_{m+1;n} + \delta_{m-1;n} + \delta_{m+L_x;n} + \delta_{m-L_x;n} + \delta_{m+L_xL_y;n} + \delta_{m-L_xL_y;n})
\]
3.7 Visualization of order parameter fields

3.7.1 Scalar field in Allen-Cahn equation

To visualize the statics and dynamics of the scalar field, we consider the Allen-Cahn equation in three dimensions, defined in Eq.(3.4). We use the MOL technique for the spatial discretization and Crank-Nicholson implicit integrator for the time update. We use the standard PETSc library [6] to domain-decompose the right hand side of the Eq.(3.13) as well as the Jacobian matrix defined in Eq.(3.14). For visualization in three dimensions purpose, we draw isosurfaces of the scalar field corresponding to a specific field value (isovalue).

Fig.[3.7(a)] shows the isosurfaces of the scalar magnetization at an isovalue $\psi = 0.1$. In Fig.[3.7(b)] we plot the isosurfaces shown in the previous frame, but in false colours so that the isosurface corresponding to $\psi = 0.1$ appear blue while $\psi = -0.1$ appear green. We also render the volume with false colours corresponding to the equilibrium field values $\pm 1$, superimposed on the coloured isosurfaces. We notice the surface pinch-off effect at the upper-right corner of the right face of the rectangle. This creates a complete picture of spinodal quench, where the scalar order is set to $\pm 1$ initially and the isosurfaces of lower order tries to equilibrate through the surface pinch off. As magnetization is not a conserved

The Jacobian matrix for the nematic equations of motion is shown in the Appendix A.
quantity, the amount of surface area between ±1 is not constant.

3.7.2 Tensor field in nematics

The nematic orientation tensor consist of two scalar fields, which are the degree of uniaxial ordering $S$ and degree of biaxial ordering $T$ and three vector fields: the nematic director $\mathbf{n}$, codirector $\mathbf{l}$ and the joint normal to this as $\mathbf{m}$. The time update of the nematodynamic equations in the basis of $a_i$ results in a set of $a_i$ on each of the lattice points. With these values, the nematic orientation tensor $Q$ can be reconstructed. This is symmetric and traceless by construction, but not necessarily diagonal. To extract the scalar and vectorial components from the $Q$ tensor, we again choose a reference system aligned with the principal axis that diagonalizes the matrix. The largest eigenvalue of the diagonal matrix corresponds to $S$, while the difference between the other two eigenvalue correspond to $T$ and the corresponding eigenvectors designate the $[\mathbf{nlm}]$ triad.

To visualize the scalar field, we colour the field values on every lattice point with its respective value, using a specified convention, e.g. dark blue for the lowest value of the field and orange/red for the highest, and all other colours correspond to an ascending field values from blue to orange/red. We also perform a polynomial interpolation of the colours, which smoothen out the colour patches and a continuous coloured plane in two dimensions or volume in three dimensions results.

To visualize the vector field, we draw vectors at each lattice point with either the components in the three Cartesian directions or with the angular coordinates $(\theta, \phi)$ whichever is available. As the parallel and anti-parallel direction are equivalent of the nematic director, we plot a ‘needle’ as opposed to an ‘arrow’ to designate the vector field. We obtain the director field imposed on top of the scalar field, as in Fig.(3.8) to easily visualize the physical situation.
3.8 Conclusion

In this chapter we have described the numerical techniques we will apply in subsequent chapters to solve the deterministic and the stochastic form of the dynamics of nematic orientation tensor. We have shown how the non dimensionalization of the parameters can lead to a simplification in the parameters. We have presented the method of lines applied to the dynamics followed by a spectral collocation technique. The $tanh$ interface for the Allen-Cahn equation benchmarks our algorithm.

We then discuss the stochastic method of lines applied to the fluctuating dynamics of nematics. We benchmark the algorithm by calculating the velocity autocorrelation of a Brownian particle. We then exhibit a finite discretization of the dynamic equations, which can be used in the calculation of the static and dynamic correlators. We describe high performance computation applied to the dynamics of a scalar field. We conclude this part of the thesis with a visualization technique for the scalar field and show how a superimposed vector field description can be used for the visualization of the properties of the tensorial field.