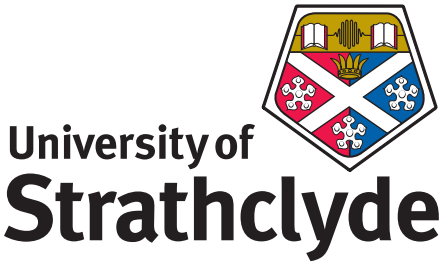

Efficient moving mesh methods for Q-tensor models of liquid crystals

C. S. MacDonald

Department of Mathematics and Statistics

University of Strathclyde

Glasgow, Scotland

February 2018

This thesis is submitted to the University of Strathclyde for the degree of Doctor of Philosophy in the Faculty of Science.

Acknowledgements

Firstly I would like to thank my supervisors, Dr John Mackenzie and Dr Alison Ramage, without whom this thesis would simply not have been possible. They have continually provided patient, valuable input over the last number of years, and for that I am very grateful.

I would also like to thank Dr Chris Newton, who provided valuable input into this research. Further thanks goes to HP Labs and EPSRC for the financial support provided.

I'd also like to acknowledge all who worked in the 839 office, who undoubtedly played a part in the progress of this thesis.

Lastly I would like to thank Dr Katherine Tant, and my parents, Alan and Marian MacDonald, who continually reminded me that I had a thesis to write. Their support is hugely appreciated.

Abstract

As devices using liquid crystals become ever smaller and increasingly complex, there is a commensurate increase in the need for more effective numerical modelling tools in the area. In this thesis, an adaptive finite element method is used to solve a non-linear singularly perturbed boundary value problem which arises from a one-dimensional **Q**-tensor model of liquid crystals. The adaptive non-uniform mesh is generated by equidistribution of a selection of strictly positive monitor functions. By an appropriate selection of the monitor function parameters, it is shown that the computed numerical solution converges at an optimal rate with respect to the mesh density and that the solution accuracy is robust to the size of the singular perturbation parameter.

A robust and efficient numerical scheme is then used to solve the system of six coupled partial differential equations which arises from $\boldsymbol{Q}$-tensor theory. The key novel feature is the use of a full moving mesh partial differential equation (MMPDE) approach to generate an adaptive mesh which accurately resolves im-

portant solution features. This includes the use of a new monitor function based on a local measure of biaxiality. The behaviour of the method is illustrated on a one-dimensional time-dependent problem in a π-cell geometry with an applied electric field. The numerical results show that, as well as achieving optimal rates of convergence in space and time, higher levels of solution accuracy and a considerable improvement in computational efficiency are obtained compared to other moving mesh methods used by previous authors on similar problems.

The numerical scheme is then extended to tackle a two-dimensional π-cell problem. It is shown that the adaptive moving mesh method copes well with the presence of moving defects, with the mesh adapting and relaxing to capture the motion, growth and annihilation of the defects.

Publications

The following publications contain some of the novel work presented in this thesis:

1. C. S. MacDonald, J. A. Mackenzie, A. Ramage, and C. J. P. Newton. Robust adaptive computation of a one-dimensional $\boldsymbol{Q}$-tensor model of nematic liquid crystals. Computers and Mathematics with Applications, 64(11):3627-3640, 2012. [48]

2. C. S. MacDonald, J. A. Mackenzie, A. Ramage, and C. J. P. Newton. Efficient moving mesh methods for $\boldsymbol{Q}$-tensor models of liquid crystals. SIAM Journal on Scientific Computing, 37(2):B215–B238, 2015. [49]

3. C. S. MacDonald, J. A. Mackenzie, A. Ramage, and C. J. P. Newton. Efficient moving mesh methods applied to a two-dimensional $\boldsymbol{Q}$-tensor model of nematic liquid crystals. In preparation.

Contents

Chapter 1

Introduction

It is the botanist Friedrich Reinitzer, pictured in Figure 1.1, who is credited with discovering liquid crystals in 1888 [64]. Whilst observing the material *cholesteryl benzoate*, Reinitzer observed that it had two melting points. He found that the material, solid at room temperature, melted to a cloudy liquid state at 145.5°C. Upon heating the sample further he observed that a second transition took place,

Figure 1.1: Austrian botanist and chemist Friedrich Reinitzer (1857 – 1927).

with the liquid becoming transparent at 178.5°C. This is considered the first experimental evidence of liquid crystals: materials that exhibit mesomorphic phases, that is, they are intermediate states of matter which occur between the crystalline solid state and the isotropic liquid state, and display some of the properties of both. Reinitzer, through this discovery, gave birth to a fertile area of research that has grown massively over the past century. Liquid crystals are now firmly embedded in the world around us, and contribute to many, varied areas of scientific research. We refer interested readers to works, which although outside the direct interests of this thesis, demonstrate some of the varied applications of liquid crystals: from the analysis of heat transfer characteristics in e.g. gas turbines [57, 66, 77, 43, 58]; as diagnostic tools for surface shear-stress visualization in hypersonic flows [63]; for measuring the temperature of flowing solid particles in food processing [8]; in digital particle image thermometry [24]; and in non-destructive testing [31]. Their optical properties have of course also seen them used extensively in LCDs, see e.g. [71, 22, 47, 67], which have been at the core of a vastly profitable multi-billion pound industry since their conception in the 1960s. In recent years, the use of LCDs in consumer goods such as televisions, laptops, mobile phones, tablets, e-readers, and an ever increasing number of other devices, has grown rapidly. As a result, there is increasing interest in the development of efficient simulation tools for accurately modelling the optical properties of liquid crystals.

Although discovered by Reinitzer in 1888, it was Friedel who later, in 1922, first

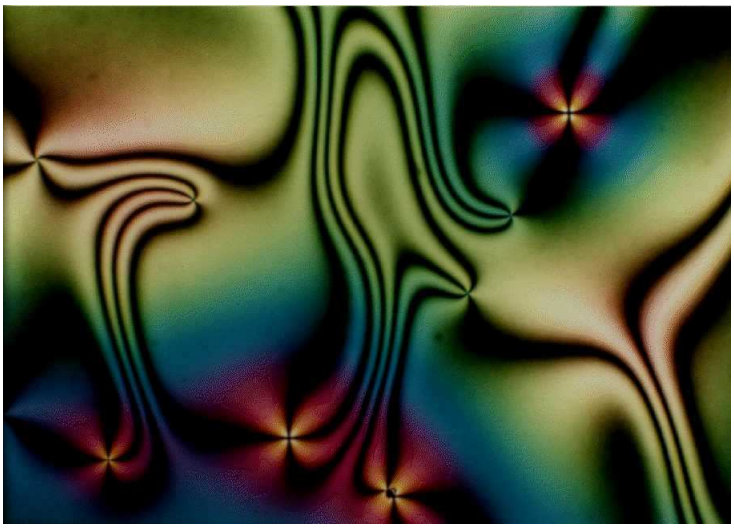

Figure 1.2: Schlieren texture of a nematic liquid crystal.

officially classified liquid crystals as a fourth phase of matter [32]. Additionally, he classified liquid crystals into three further phases: nematic, smectic and cholesteric. The liquid crystal phase most commonly used in modern devices is the nematic phase: throughout this thesis we shall consider only the nematic liquid crystal 4-cyano-4'-n-pentylbiphenyl, which at temperatures below 18 °C is in a crystalline state, between 18 °C and 35 °C is in a nematic liquid crystal state, and above 35 °C is in an isotropic state. Nematic, being derived from the Greek word for thread, describes liquid crystals that possess thread-like defects, described as Schlieren texture, as in Figure 1.2 [42]. Nematic liquid crystals, whilst exhibiting high orientational order, have no positional order (we expand on what this means in §1.1).

1.1 Properties of liquid crystals

Different liquid crystal phases may be classified by the amount and type of orientational and positional order of molecules within a particular material.

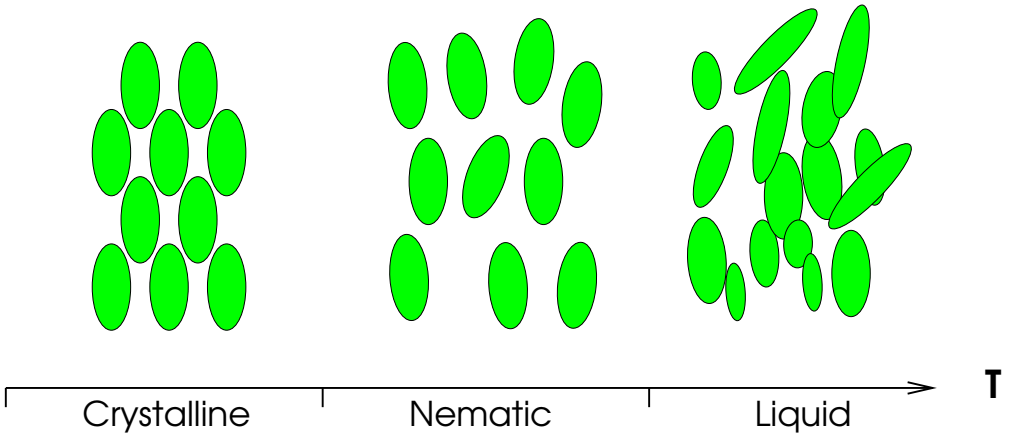

Figure 1.3: Liquid crystal phases varying with temperature, where temperature is increasing from left to right.

The degree of orientational order in most liquid crystals depends on temperature; that is, the liquid crystals are thermotropic. The liquid crystal phase occurs at temperatures that are sufficiently high so that the material is not in the crystalline phase, but low enough so the material is not in the liquid phase. Figure 1.3 provides an illustration of the molecular order of the possible material phases, changing from solid crystal, to nematic liquid crystal, to liquid as temperature is increased. In the solid crystal state the molecules are rigidly arranged, forming a lattice type configuration ensuring that, relative to neighbouring molecules, the molecules have strong orientational and positional ordering. At the other end of

the scale we have the isotropic liquid. In this state the molecules are randomly orientated and positioned, but have sufficient molecular strength of bonding to prevent the material entering a gaseous state. A liquid crystal falls between these two states. The molecules in a liquid crystal desire to point along a common axis, known widely as the *director*, retaining some of the orientational order but (in the case of a nematic liquid crystal) none of the positional order of the solid crystalline state. It is the tendency of the molecules to align parallel to a preferred axis, and the ability for this preference to be determined by external forces, such as electric and magnetic fields, that makes the liquid crystals useful in optical devices. Competition between the influences of bounding surfaces and the interaction between the permanent or induced electric dipoles of the liquid crystal molecules and an applied electric field can cause the material to switch between different orientational states, with the resulting change in optical characteristics allowing the material to be used in LCDs.

1.1.1 Computational modelling of liquid crystals

One of the difficulties in liquid crystal modelling is that the underlying physical problems frequently involve characteristic length and time scales which vary by many orders of magnitude. Areas where distortion of the liquid crystal occurs over small length scales (between 10–100 nm) are of great importance, and it is

crucial that the behaviour and nature of these so-called *defects* can be accurately represented by any numerical model applied to liquid crystal problems where defects occur. This can cause problems in terms of designing accurate and efficient numerical simulation techniques. The presence in the physical problem of characteristic lengths with large scale differences (the size of the defect is very small compared to that of the liquid crystal cell which is about 1–10 μm) suggests that sophisticated numerical modelling techniques could be used here to great effect. Typically such issues are tackled by introducing some form of adaptive meshing, which is known to be successful when dealing with vast differences in length scales.

In this thesis we focus on adaptive finite element methods (see e.g. [5, 75, 16, 6, 20, 41]), that can typically be classified in three groups: h-refinement, p-refinement and r-refinement. The term h-refinement describes the process of adding additional grid points locally in regions of high error in order to achieve a desired level of accuracy [37, 75, 5]. Critical to the method is identifying which mesh elements should be selected for refinement or coarsening. Degrees of freedom will be added to areas where some local measure of the solution difficulty is high, and degrees of freedom will be removed from areas where a local measure of the solution difficulty is low. The principal benefit of h-refinement is that, with appropriate mesh refinements and sufficient degrees of freedom, it should be possible to attain a required level accuracy. The disadvantages of h-refinement are that it is restricted to algebraic convergence, and for time-dependent problems, it can easily miss rapidly

evolving solution features (though this can also be said for other approaches, e.g. r-refinement).

The second strategy, p-refinement, differs from h-refinement in that it generally has fixed number of elements, and the local polynomial degree of the finite element basis functions, denoted by p, is increased or decreased based again on some local error indicator [37, 69, 74]. The benefits of p-refinement are that it can rapidly reduce solution error as the order of the approximating polynomials is increased. However, for time-dependent problems, we cannot presume to know where evolving solution features (such as defects) lie in the domain (this issue is not exclusive to p-refinement, and can equally apply to other refinement techniques). It is therefore far from obvious how best to apply p-refinement in a sensible manner.

In a finite element framework, it is popular to combine the h-refinement and p-refinement approaches. With hp-refinement the local polynomial degree of the basis is varied simultaneously with the coarsening and refinement process. The work of Babuška and Guo [37] discovered that the hp method converges exponentially in the number of degrees of freedom, whereas h-refinement only attains algebraic convergence, and p-refinement attains polynomial convergence, unless the problem is smooth, in which case p-refinement will also converge exponentially. The method has since been the focus of extensive research over the last few decades and we refer the reader to further works by Babuška and Guo [36, 35] and by Oden and Demkowicz [29, 55].

Another strategy is characterised by meshes that move continuously in time to adapt to evolving solution features. The method dynamically *relocates* mesh points to regions of high error while maintaining the same grid connectivity, and is thus described as r-refinement. This strategy will be the focus of this thesis. The advantages of this approach include relatively simple implementation, comparatively easy extension of existing software for fixed mesh methods and minimal numerical diffusion and dispersion (that is, the shape and speed of time-dependent features of the solution are accurately reproduced). Comprehensive accounts of the current state-of-the-art of adaptive moving mesh methods can be found in [20, 41, 50, 7, 19, 51, 79].

In terms of liquid crystal modelling, local mesh refinement techniques such as h-refinement, or increasing the degree of the polynomial approximation, p-refinement, have been used with some success [23, 33, 34, 44, 53, 59, 70]. However, such methods often involve a need for complicated evolving data structures, particularly for time-dependent problems when regions requiring high spatial resolution can move throughout the domain. There have also been several methods proposed which use adaptive moving meshes [2, 3, 61, 62]. It is accepted that moving mesh methods are an efficient and effective means of resolving solutions that contain sharp features, such as boundary and interior layers, and localised solution singularities. Further detail on existing work in applying moving mesh methods to liquid crystal problems, and where the research in this thesis advances on this

work, is contained in the introductions to Chapters 4, 5 and 6.

1.2 Outline of Thesis

In Chapter 2 we provide an introduction to liquid crystal modelling; specifically we present the $\boldsymbol{Q}$-tensor model which will be the focus of this thesis. By considering the total free energy of a liquid crystal cell under the influence of an applied electric field, we derive a system of time-dependent equations for the entries of the $\boldsymbol{Q}$-tensor. We also introduce Maxwell's equations as a means of calculating the electric field in the cell. Taken together, the equations for the entries of the $\boldsymbol{Q}$-tensor and the equation for calculating the electric field, provide the complete system of partial differential equations (PDEs) that will be the focus of the thesis.

In Chapter 3 we construct a weak arbitrary Lagrangian-Eulerian formulation of the PDEs presented in Chapter 2. In this formulation we must take special account of the fact that we are solving on an evolving mesh. We then present the finite element semi-discretisation of the equations, and the resulting highly non-linear differential algebraic system that will be solved in a variety of settings, in Chapters 3, 4 and 5.

In Chapter 4 we present a uniaxial one-dimensional liquid crystal model problem similar to that studied in [62]. Using a spatial rescaling, we highlight the singular perturbation nature of the model problem and discuss an asymptotic

expansion of the solution in the boundary layer region. We then present an explanation of the concept of mesh equidistribution, and the importance of the mesh density function in producing high quality adaptive meshes. An iterative algorithm is used to find approximately equidistributed grids, and to obtain solutions to a given degree of accuracy. We show that the use of the Beckett-Mackenzie monitor function [12, 13] can result in over a thousandfold decrease in CPU time compared to the use of the arc-length monitor function considered by previous authors. We also present numerical experiments using linear and quadratic finite elements which confirm that robust and optimal rates of convergence can be obtained for the one-dimensional nematic liquid crystal cell studied.

In Chapter 5 we consider a time-dependent switching process in a π-cell geometry which admits two topologically different equilibrium states. This so-called order reconstruction problem originates from attempts to model real phenomena first observed in laboratory experiments [9]. A moving mesh partial differential equation (MMPDE) approach is used to generate the adaptive mesh. A conservative finite element discretization using quadratic elements is used to update the solution on the adaptive moving mesh. Time integration of the $\boldsymbol{Q}$-tensor equations is achieved using a second order semi-implicit Runge–Kutta scheme and adaptive time-step control. These components form an adaptive algorithm that is carefully tested, and the computed solutions are shown to converge at optimal rates in both space and time. Evidence is presented to suggest that the computed solutions

exhibit nodal superconvergence, which is somewhat surprising given the highly nonuniform nature of the adaptive moving meshes. For the first time, a monitor function is constructed based upon a local measure of biaxiality. This is shown to lead to higher levels of solution accuracy and a considerable improvement in computational efficiency compared to those monitor functions used previously for liquid crystal problems.

In Chapter 6 we extend our MMPDE approach and the conservative finite element discretisation of the $\boldsymbol{Q}$-tensor equations to a two-dimensional setting. Adopting a similar strategy to previous chapters, we apply a number of different monitor functions to the problem, and present results which indicate that monitor functions based on a local measure of biaxiality produce good quality meshes. We then apply the biaxiality-based monitor function to a problem first presented by Bos [80]: a two dimensional π-cell problem with a sinusoidal perturbation across the centre of the cell. We observe that our choice of monitor function does a particularly good job of resolving the moving defects present in the liquid crystal cell. We also present convergence results, showing that computed solutions exhibit optimal rates of convergence when applied to a two-dimensional stationary defect first studied in [68].

In Chapter 7 we present a review of the work undertaken in the thesis, and finish by discussing the possibilities for extending this work in the future.

Chapter 2

Liquid crystal theory

The description of the general biaxial nematic state requires specification of two directors $\mathbf{n}(\mathbf{x},t)$ and $\mathbf{m}(\mathbf{x},t)$, which describe the mean molecular alignment at a point in a give sample volume, and two scalar order parameters $S_1(\mathbf{x},t)$ and $S_2(\mathbf{x},t)$, which measure how ordered the molecules are in the $\mathbf{n}$ and $\mathbf{m}$ directions respectively. The perpendicular unit directors $\mathbf{n}$ and $\mathbf{m}$, shown in Figure 2.1, can be written as

$$\begin{aligned}\mathbf{n} &= (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta),\\ \mathbf{m} &= (\sin\phi\cos\psi - \cos\phi\sin\psi\sin\theta, -\sin\phi\sin\psi\sin\theta - \cos\phi\cos\psi, \sin\psi\cos\theta).\end{aligned}$$

There are now five dependent variables, $S_1(\mathbf{x},t)$, $S_2(\mathbf{x},t)$, $\theta(\mathbf{x},t)$, $\phi(\mathbf{x},t)$ and $\psi(\mathbf{x},t)$, from which a theory can be constructed. However it should be noted that any theory based on these three Euler angles will encounter the following problems:

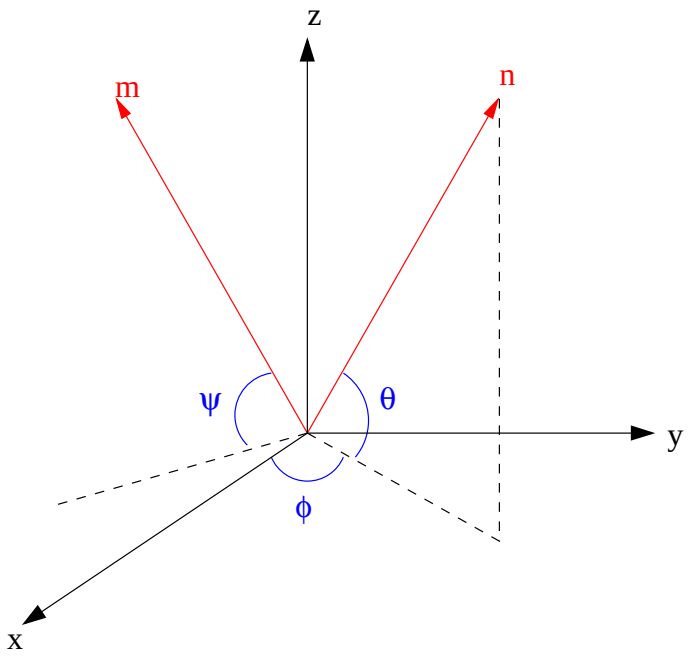

Figure 2.1: The directors **n** and **m** in terms of the Euler angles θ, ϕ and ψ. The axes x, y and z form the laboratory frame of reference [53].

when the zenithal angle $\theta = \pi/2$, the azimuthal angle ϕ is undefined, and $\phi = 0$ is equivalent to $\phi = 2\pi$ and so there may be problems with multi-valuedness. An alternative theory, $\boldsymbol{Q}$-tensor theory, removes the problems that arise from this angle representation.

2.1 Q-tensor theory

For nematic systems in equilibrium, such as those that are considered in this thesis, the globally stable phase of the system is the stationary point of the free energy functional with least free energy. In Landau-de Gennes theory [27, 28], the free

energy density is usually assumed to depend on the tensor order parameter $\boldsymbol{Q}$ and its gradient. We define this second rank order tensor by

$$\boldsymbol{Q} = \sqrt{\frac{3}{2}} \left\langle \mathbf{u} \otimes \mathbf{u} - \frac{1}{3}\mathbf{I} \right\rangle, \tag{2.1}$$

where $\langle \cdots \rangle$ represents the local ensemble average over the unit vectors $\mathbf{u}$ along the molecular axes and $\mathbf{I}$ is the identity. It has five degrees of freedom, two of which specify the degree of order, and three of which specify the angles of the principal directions. For directors $\boldsymbol{n}$ and $\boldsymbol{m}$ and scalar order parameters S_1 and S_2, $\boldsymbol{Q}$ is given by

$$\boldsymbol{Q} = \sqrt{\frac{3}{2}} \left(S_1(\mathbf{n} \otimes \mathbf{n}) + S_2(\mathbf{m} \otimes \mathbf{m}) - \frac{1}{3}(S_1 + S_2)\mathbf{I} \right). \tag{2.2}$$

This symmetric, traceless tensor $\boldsymbol{Q}$ can be represented as

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & q_4 & q_5 \\ q_3 & q_5 & -q_1 - q_4 \end{bmatrix} \tag{2.3}$$

where each of the five quantities q_i, $i = 1, \ldots, 5$, is a function of temporal and spatial co-ordinates. The eigenframe of $\boldsymbol{Q}$ describes the molecular alignment of the liquid crystal, and therefore an understanding of the physical meaning of the eigenvalues and eigenvectors is critical. Specifically, $\boldsymbol{Q}$ has eigenvectors $\mathbf{n}$, $\mathbf{m}$ and

$\mathbf{n} \times \mathbf{m}$ with respective eigenvalues

$$\begin{aligned} \mathbf{n} \quad &: \quad \lambda_1 \;=\; \frac{1}{3}\left(2S_1 - S_2\right), \\ \mathbf{n} \times \mathbf{m} \quad &: \quad \lambda_2 \;=\; \frac{1}{3}\left(-S_1 - S_2\right), \\ \mathbf{m} \quad &: \quad \lambda_3 \;=\; \frac{1}{3}\left(2S_2 - S_1\right). \end{aligned}$$

When two eigenvalues of $\boldsymbol{Q}$ are the same the liquid crystal is in a uniaxial state, i.e., when $\lambda_1 = \lambda_2$, $\lambda_2 = \lambda_3$, or $\lambda_1 = \lambda_3$. An isotropic system exists when all eigenvalues of $\boldsymbol{Q}$ are equal, i.e., when $S_1 = 0$, $S_2 = 0$, so that $\boldsymbol{Q} = \mathbf{0}$.

The principle advantage of the $\boldsymbol{Q}$-tensor description over the director-based approaches, discussed briefly in §2, is that topological defects do not appear in the model as mathematical singularities. This is clearly important when we are seeking to model liquid crystal systems where defects, which present themselves as singularties, are present. Further comment on the benefits of each method is outside the scope of this thesis, however an in-depth discussion on the $\boldsymbol{Q}$-tensor model, and further detail on its connection to the more traditional Frank-Oseen director-based models, can be found in [52, 53].

2.2 Phenomenological theory

The theoretical framework is formed by the construction of the total free energy of the liquid crystal cell, $\mathcal{F}$. If it is assumed that distortions of $\boldsymbol{Q}$ are small, then

$\mathcal{F}$ can be taken to depend only on $\boldsymbol{Q}$ and its spatial gradients. That is, we may write

$$\mathcal{F} = \int_V \mathcal{F}_{bulk}(\boldsymbol{Q}, \nabla \boldsymbol{Q})\, dV + \int_{\mathcal{S}} \mathcal{F}_{surface}(\boldsymbol{Q})\, dS. \tag{2.4}$$

In this thesis we will primarily be interested in distortions in the liquid crystal cell due to an applied electric field. For a liquid crystal cell under the influence of an applied electric field, we can rewrite (2.4) as

$$\mathcal{F} = \int_V (\mathcal{F}_t(\boldsymbol{Q}) + \mathcal{F}_e(\boldsymbol{Q}, \nabla \boldsymbol{Q}) + \mathcal{F}_u(\boldsymbol{Q}, \nabla \boldsymbol{Q}))\, \mathrm{d}V + \int_{\mathcal{S}} \mathcal{F}_s(\boldsymbol{Q})\, \mathrm{d}S, \tag{2.5}$$

where $\mathcal{F}_t$, $\mathcal{F}_e$, $\mathcal{F}_u$ and $\mathcal{F}_s$ represent the thermotropic, elastic, electrostatic and surface energy terms respectively. The structure of these terms will be examined individually below. Note that in this thesis we will apply fixed boundary conditions (known as *strong anchoring*), therefore the surface energy term in the free energy expression is constant so can be ignored in the minimisation, and (2.5) becomes

$$\mathcal{F}_b = \int_V (\mathcal{F}_t(\boldsymbol{Q}) + \mathcal{F}_e(\boldsymbol{Q}, \nabla \boldsymbol{Q}) + \mathcal{F}_u(\boldsymbol{Q}, \nabla \boldsymbol{Q}))\, \mathrm{d}V, \tag{2.6}$$

where $\mathcal{F}_b$ denotes the free energy in the bulk. In the $\boldsymbol{Q}$-tensor model the thermotropic energy is typically presented in the form of a Landau power series expansion: taking the thermotropic energy, $\mathcal{F}_t$, up to fourth order in $\boldsymbol{Q}$ we obtain

$$\mathcal{F}_t = \frac{1}{2}A(T - T^*)\ \mathrm{tr}\ \boldsymbol{Q}^2 - \frac{\sqrt{6}}{3}B\ \mathrm{tr}\ \boldsymbol{Q}^3 + \frac{1}{4}C(\mathrm{tr}\ \boldsymbol{Q}^2)^2, \tag{2.7}$$

where A, B, C are coefficients independent of temperature. The temperature dependence is restricted solely to the term $T - T^*$, where T represents temperature

and T^* is the pseudocritical temperature at which the isotropic phase becomes unstable (see [26]). Taking the elastic energy, $\mathcal{F}_e$, up to second order in the gradient of $\boldsymbol{Q}$, we obtain

$$\mathcal{F}_e = \frac{1}{2}L_1(\text{div } \boldsymbol{Q})^2 + \frac{1}{2}L_2|\nabla \times \boldsymbol{Q}|^2, \tag{2.8}$$

where L_1 and L_2 are positive material constants related to the liquid crystal elastic constants. The contribution to the bulk energy from the applied electric field, $\boldsymbol{E}$ say, can be written as

$$\mathcal{F}_u = -\frac{1}{2}\epsilon_0 \boldsymbol{E} \cdot \boldsymbol{\epsilon} \boldsymbol{E} - \boldsymbol{P}_{fl} \cdot \boldsymbol{E}, \tag{2.9}$$

where $\boldsymbol{\epsilon} = \bar{\epsilon}\boldsymbol{I} + \Delta\epsilon^* \boldsymbol{Q}$ is the dielectric tensor, $\bar{\epsilon} = (\epsilon_\parallel + 2\epsilon_\perp)/3$ is the average permittivity, $\Delta\epsilon^* = \sqrt{2}(\epsilon_\parallel - \epsilon_\perp)/\sqrt{3}$ is the scaled dielectric anisotropy and ϵ_0 is the permittivity of free space. The flexoelectric contribution is taken to be $\boldsymbol{P}_{fl} = \bar{e}$ div $\boldsymbol{Q}$.

Note that the thermotropic coefficients, A, B and C can be measured, while the elastic constants, L_1 and L_2 are obtained by using the connection between the more traditional Frank-Oseen director-based free energy density and the $\boldsymbol{Q}$-tensor representation (see, e.g., [52]).

The electric field within the cell may be found by solving Maxwell's equation. If we define an (unknown) scalar electric potential U such that $\boldsymbol{E} = -\nabla U$, this reduces to solving $\nabla \cdot \boldsymbol{D} = 0$ where the electric displacement $\boldsymbol{D}$ is given by

$$\boldsymbol{D} = \epsilon_0 \boldsymbol{\epsilon} \nabla U + \boldsymbol{P}_{fl}. \tag{2.10}$$

2.3 Material constants

The values used for material constants throughout this thesis are taken from [9]:

$$\begin{aligned} A &= 0.13 \times 10^6 \mathrm{JK}^{-1}\mathrm{m}^{-3}, \\ B &= 1.6 \times 10^6 \mathrm{Jm}^{-3}, \quad C = 3.9 \times 10^6 \mathrm{Jm}^{-3}, \\ L_1 &= 9.7 \times 10^{-12}\mathrm{N}, \quad L_2 = 2.4 \times 10^{-12}\mathrm{N}, \\ \epsilon_\perp &= 5, \quad \epsilon_\parallel = 20, \\ \bar{e} &= -27 \times 10^{-12}\mathrm{Cm}^{-1}. \end{aligned}$$

These values are commensurate with a liquid crystal cell of the 5CB compound 4-cyano-4'-n-pentylbiphenyl and correspond to a nematic coherence length of $\zeta \approx 4.06$ nanometres (which is the characteristic length over which the local molecular order of the liquid crystal persists [28]).

2.4 The static equations

For static problems the quantities q_i in (2.3) can be found by solving the Euler-Lagrange equations, which minimises the total free energy of the system. The Euler-Lagrange equations for the total free energy (2.6) are

$$\sum_{j=1}^{3} \frac{\partial}{\partial x_j}\left(\frac{\partial \mathcal{F}_b}{\partial q_{i,j}}\right) = \frac{\partial \mathcal{F}_b}{\partial q_i}, \qquad i = 1, \ldots, 5, \tag{2.11}$$

where

$$q_{i,j} = \frac{\partial q_i}{\partial x_j}. \tag{2.12}$$

The five equations (2.11) can then be solved, subject to suitable boundary conditions, to find $\mathbf{Q}$. We discuss boundary conditions in more detail in Chapter 4, where we present a simplified static problem.

2.5 The dynamic equations

For dynamic systems we may derive time-dependent equations for the quantities q_i in (2.3) using a dissipation principle based on the function

$$\mathcal{D} = \mathrm{tr}\left[\left(\frac{\partial \boldsymbol{Q}}{\partial t}\right)^2\right] = 2(\dot{q}_1\dot{q}_4 + \dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2 + \dot{q}_4^2 + \dot{q}_5^2), \tag{2.13}$$

where the dot represents differentiation with respect to time. Specifically, with spatial coordinates $\{x_1, x_2, x_3\}$, and viscosity ν this leads to a system of five equations

$$\nu\frac{\partial \mathcal{D}}{\partial \dot{q}_i} = \nabla \cdot \hat{\mathbf{\Gamma}}_i - \hat{f}_i \qquad i = 1, \ldots, 5, \tag{2.14}$$

where the vector $\hat{\mathbf{\Gamma}}_i$ has entries

$$(\hat{\mathbf{\Gamma}}_i)_j = \frac{\partial \mathcal{F}_b}{\partial q_{i,j}}, \qquad q_{i,j} = \frac{\partial q_i}{\partial x_j}, \qquad j = 1, 2, 3,$$

$\hat{f}_i$ is given by

$$\hat{f}_i = \frac{\partial \mathcal{F}_b}{\partial q_i},$$

and $\dot{q}_i$ is given by

$$\dot{q}_i = \frac{\partial q_i}{\partial t}.$$

Note that the viscosity ν is related to the standard nematic viscosity γ_1 by $\nu = 1/\sqrt{3S^2}\gamma_1$. The equations for $i = 2, 3, 5$ are of the form

$$4\nu\dot{q}_i = \nabla \cdot \hat{\mathbf{\Gamma}}_i - \hat{f}_i, \tag{2.15}$$

and for $i = 1, 4$ we have

$$2\nu(2\dot{q}_4 + 4\dot{q}_1) = \nabla \cdot \hat{\mathbf{\Gamma}}_1 - \hat{f}_1, \tag{2.16a}$$

$$2\nu(2\dot{q}_1 + 4\dot{q}_4) = \nabla \cdot \hat{\mathbf{\Gamma}}_4 - \hat{f}_4, \tag{2.16b}$$

respectively. From equations (2.15), (2.16a), and (2.16b) we obtain

$$\begin{aligned}
6\nu\dot{q}_1 &= \nabla \cdot (2\hat{\mathbf{\Gamma}}_1 - \hat{\mathbf{\Gamma}}_4) - (2\hat{f}_1 - \hat{f}_4), \\
4\nu\dot{q}_2 &= \nabla \cdot \hat{\mathbf{\Gamma}}_2 - \hat{f}_2, \\
4\nu\dot{q}_3 &= \nabla \cdot \hat{\mathbf{\Gamma}}_3 - \hat{f}_3, \\
6\nu\dot{q}_4 &= \nabla \cdot (2\hat{\mathbf{\Gamma}}_4 - \hat{\mathbf{\Gamma}}_1) - (2\hat{f}_4 - \hat{f}_1), \\
4\nu\dot{q}_5 &= \nabla \cdot \hat{\mathbf{\Gamma}}_5 - \hat{f}_5.
\end{aligned} \tag{2.17}$$

On combining (2.10) and (2.17), after some manipulation, we arrive at the final system involving six coupled non-linear PDEs for q_i, $i = 1, \ldots, 5$, and the electric potential U, given by

$$\frac{\partial q_i}{\partial t} = \nabla \cdot \mathbf{\Gamma}_i - f_i, \qquad i = 1, \ldots, 5, \tag{2.18a}$$

$$\nabla \cdot \boldsymbol{D} = 0, \tag{2.18b}$$

where

$$\begin{aligned}
\boldsymbol{\Gamma}_1 &= \frac{1}{6\nu}(2\hat{\boldsymbol{\Gamma}}_1 - \hat{\boldsymbol{\Gamma}}_4), \qquad f_1 = \frac{1}{6\nu}(2\hat{f}_1 - \hat{f}_4), \\
\boldsymbol{\Gamma}_4 &= \frac{1}{6\nu}(2\hat{\boldsymbol{\Gamma}}_4 - \hat{\boldsymbol{\Gamma}}_1), \qquad f_4 = \frac{1}{6\nu}(2\hat{f}_4 - \hat{f}_1), \\
\boldsymbol{\Gamma}_i &= \frac{1}{4\nu}\hat{\boldsymbol{\Gamma}}_i, \qquad f_i = \frac{1}{4\nu}\hat{f}_i, \quad i = 2, 3, 5.
\end{aligned} \tag{2.19}$$

Solving the system of PDEs presented in (2.18) will be the focus of the remainder of this thesis. Detailed description of the terms in (2.19) can be found in Appendix A. The boundary and initial conditions applied to (2.18) are discussed in later chapters, when we present the specific problems to which we apply our numerical method.

Chapter 3

Finite element method

The system of PDEs (2.18) is highly non-linear, and for the problems we consider in this thesis no known analytic solution exists. An appropriate numerical method is thus required to obtain an approximate solution to the equations. The underlying idea of any numerical method for solving PDEs is to reduce the continuous problem to a system of algebraic equations with a finite number of degrees of freedom. There are a number of different approaches that can prove successful for solving non-linear systems of PDEs. There are three main mesh-based approaches, i.e., finite differences, finite elements, and finite volumes, and a number of mesh-free numerical methods. This thesis will focus on the finite element method: it is reasonably straightforward to implement; can deal with complex domains; allows simple implementation of boundary conditions; and can also handle unstructured meshes with ease.

3.1 Moving mesh finite element method

With a moving mesh method, the finite element mesh adapts over time, but retains its original structure and connectivity. There are two main computational challenges here: the governing physical PDEs need to be reformulated to account for the movement of the mesh, and a new adaptive mesh has to be generated at each time step. To tackle the first of these, it is convenient to introduce a family of bijective mappings $\mathcal{A}_t$ such that, at time t, point ξ of a computational reference configuration Ω_c is mapped to point $\boldsymbol{x}$ of the current physical domain Ω. That is,

$$\mathcal{A}_t : \Omega_c \subset \mathbb{R}^2 \to \Omega \subset \mathbb{R}^2, \qquad \boldsymbol{x}(\xi, t) = \mathcal{A}_t(\xi). \tag{3.1}$$

If a mapping $g : \Omega \to \mathbb{R}$ is defined on the physical domain, and $T \subseteq \mathbb{R}^+$ represents the time domain, then the temporal derivative of g in the computational frame is defined as

$$\left.\frac{\partial g}{\partial t}\right|_{\xi} : \Omega \to \mathbb{R}, \quad \left.\frac{\partial g}{\partial t}\right|_{\xi} (\boldsymbol{x}, t) = \frac{\partial \hat{g}}{\partial t}(\xi, t), \quad \xi = \mathcal{A}_t^{-1}(\boldsymbol{x}),$$

where $\hat{g} : \Omega_c \times T \to \mathbb{R}$ is the corresponding function in the computational frame, that is, $\hat{g}(\xi, t) = \hat{g}((\boldsymbol{x}, t), t) = g(\mathcal{A}_t(\xi))$. We also define the *mesh velocity* $\dot{\boldsymbol{x}}$ as

$$\dot{\boldsymbol{x}}(\boldsymbol{x}, t) = \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\xi} (\mathcal{A}_t^{-1}(\boldsymbol{x})).$$

In general, if a function $q : \Omega \to \mathbb{R}$ is smooth enough, then applying the chain rule for differentiation gives

$$\left.\frac{\partial q}{\partial t}\right|_{\xi} = \left.\frac{\partial q}{\partial t}\right|_{\boldsymbol{x}} + \dot{\boldsymbol{x}} \cdot \nabla q.$$

We can therefore reformulate (2.18) to take account of a moving mesh as follows:

$$\left.\frac{\partial q_i}{\partial t}\right|_{\xi} - \dot{\boldsymbol{x}} \cdot \nabla q = \nabla \cdot \Gamma_i - f_i \qquad i = 1, \ldots, 5, \tag{3.2a}$$

$$\nabla \cdot \boldsymbol{D} = 0. \tag{3.2b}$$

Note that the main difference between (2.18a) and (3.2a) is the appearance of an additional convection-like term which is due to the movement of the mesh.

3.2 Conservative weak formulation

To construct a weak formulation of (3.2a) and (3.2b) we consider a space of test functions $\hat{v} \in H_0^1(\Omega_c)$. The mesh mapping (3.1) then defines the test space

$$\mathcal{H}_0(\Omega) = \left\{ v : \Omega \to \mathbb{R} : v = \hat{v} \circ \mathcal{A}_t^{-1},\ \hat{v} \in H_0^1(\Omega_c) \right\}.$$

A weak formulation of (3.2a) can be obtained using Reynolds' transport formula which states that if $\psi(\boldsymbol{x}, t)$ is a function defined on Ω, and $V_t \subseteq \Omega$ is such that $V_t = \mathcal{A}_t(V_c)$ with $V_c \subseteq \Omega_c$, then

$$\begin{aligned} \frac{d}{dt}\int_{V_t} \psi(\boldsymbol{x}, t)\mathrm{d}\boldsymbol{x} &= \int_{V_t} \left(\left.\frac{\partial \psi}{\partial t}\right|_{\xi} + \psi \nabla \cdot \dot{\boldsymbol{x}} \right) \mathrm{d}\boldsymbol{x} \\ &= \int_{V_t} \left(\left.\frac{\partial \psi}{\partial t}\right|_{\boldsymbol{x}} + \nabla\psi \cdot \dot{\boldsymbol{x}} + \psi \nabla \cdot \dot{\boldsymbol{x}} \right) \mathrm{d}\boldsymbol{x}\,. \end{aligned} \tag{3.3}$$

If functions $\hat{v} \in H_0^1(\Omega_c)$ do not depend on time, then for any $v \in H_0(\Omega)$ we can establish from (3.3) that

$$\frac{d}{dt}\int_{\Omega} v \,\mathrm{d}\boldsymbol{x} = \int_{\Omega} v \nabla \cdot \dot{\boldsymbol{x}} \,\mathrm{d}\boldsymbol{x} \tag{3.4}$$

and

$$\frac{d}{dt}\int_\Omega v\psi \,\mathrm{d}\boldsymbol{x} = \int_\Omega v\left(\left.\frac{\partial\psi}{\partial t}\right|_\xi + \psi\nabla\cdot\dot{\boldsymbol{x}}\right)\,\mathrm{d}\boldsymbol{x}. \tag{3.5}$$

A conservative weak formulation can then be obtained by multiplying (3.2a) and (3.2b) by a test function $v \in \mathcal{H}_0(\Omega)$, integrating over Ω and using (3.4) and (3.5). If $\mathcal{H}_{E_q}$ and $\mathcal{H}_{E_U}$ denote the approximation spaces with essential boundary conditions on q_i and U, respectively, then the resulting weak form is: find $q_i \in \mathcal{H}_{E_q}(\Omega)$, $i = 1,\ldots,5$, and $U \in \mathcal{H}_{E_U}(\Omega)$ such that $\forall v \in \mathcal{H}_0(\Omega)$ and

$$\frac{d}{dt}\int_\Omega q_i v \,\mathrm{d}\boldsymbol{x} - \int_\Omega (\nabla\cdot(\dot{\boldsymbol{x}}q_i))\, v \,\mathrm{d}\boldsymbol{x} = \int_\Omega \Gamma_i\cdot\nabla v \,\mathrm{d}\boldsymbol{x} - \int_\Omega f_i v \,\mathrm{d}\boldsymbol{x}, \tag{3.6a}$$

$$\int_\Omega \boldsymbol{D}\cdot\nabla v \,\mathrm{d}\boldsymbol{x} = 0. \tag{3.6b}$$

3.3 Finite element semi-discretisation

We now assume that the reference domain Ω_c is covered by a uniform partition $\mathcal{T}_{h,c}$ so that $\Omega_c = \cup_{I\in\mathcal{T}_{h,c}} I$. We will use $\mathcal{N}$ to denote the set of nodes of the finite element mesh and $\mathcal{N}_{int} \subset \mathcal{N}$ to denote the set of internal nodes. We also introduce the Lagrangian finite element spaces

$$\begin{aligned}
\mathcal{L}^k(\Omega_c) &= \{\hat{v}_h \in H^1(\Omega_c) \,:\, \hat{v}_h|_I \in \mathbb{P}_k(I),\ \ \forall\, I \in \mathcal{T}_{h,c}\}\\
\mathcal{L}^k_0(\Omega_c) &= \{\hat{v}_h \in H^1(\Omega_c) \,:\, \hat{v}_h|_I \in \mathcal{L}^k(\Omega_c) \,:\, \hat{v}_h(\xi) = 0,\ \ \xi \in \partial\Omega_c\},
\end{aligned}$$

where $\mathbb{P}_k(I)$ is the space of polynomials on I of degree less than or equal to k.

The mesh mapping (3.1) is discretised spatially using piecewise linear elements giving rise to a discrete mapping $\mathcal{A}_{h,t} \in \mathcal{L}^1(\Omega_c)$ of the form

$$\boldsymbol{x}_h(\xi, t) = \mathcal{A}_{h,t}(\xi) = \sum_{i=1}^{\mathcal{N}} \boldsymbol{x}_i(t)\hat{\phi}_i(\xi),$$

where $\boldsymbol{x}_i(t) = \mathcal{A}_{h,t}(\xi_i)$ denotes the position of node i at time t and $\hat{\phi}_i$ is the associated nodal basis function in $\mathcal{L}^1(\Omega_c)$. We denote the image of the reference interval $\mathcal{T}_{h,c}$ under the discrete mesh mapping $\mathcal{A}_{h,t}$ by $\mathcal{T}_{h,t}$. The finite element spaces on Ω are defined as

$$\begin{aligned}
\mathcal{L}^k(\Omega) &= \{v_h : \Omega \to \mathbb{R} : v_h = \hat{v}_h \circ \mathcal{A}_{h,t}^{-1},\ \hat{v} \in \mathcal{L}^k(\Omega_c)\},\\
\mathcal{H}_{h,0}(\Omega) &= \{v_h : \Omega \to \mathbb{R} : v_h = \hat{v}_h \circ \mathcal{A}_{h,t}^{-1},\ \hat{v} \in \mathcal{L}_0^k(\Omega_c)\},
\end{aligned}$$

and $\mathcal{H}_{h,E_q} \subset \mathcal{L}^k(\Omega)$ and $\mathcal{H}_{h,E_U} \subset \mathcal{L}^k(\Omega)$ are the finite dimensional approximation spaces satisfying the Dirichlet boundary conditions for q_i and U, respectively. With this notation, the finite element spatial discretisation of the conservative weak formulations (3.6a) and (3.6b) then takes the form: find $q_{ih}(t) \in \mathcal{H}_{h,E_q}(\Omega_t)$, $i = 1, \ldots, 5$, and $U_h \in \mathcal{H}_{h,E_U}(\Omega)$ such that $\forall v_h \in \mathcal{H}_{h,0}(\Omega)$

$$\frac{d}{dt}\int_\Omega q_{ih}v_h \,\mathrm{d}\boldsymbol{x} - \int_\Omega (\nabla \cdot (\dot{\boldsymbol{x}} q_{ih}))\, v_h \,\mathrm{d}\boldsymbol{x} = \int_\Omega \Gamma_{ih} \cdot \nabla v_h \,\mathrm{d}\boldsymbol{x} - \int_\Omega f_{ih}v_h \,\mathrm{d}\boldsymbol{x}, \tag{3.7a}$$

$$\int_\Omega \boldsymbol{D}_h \cdot \nabla v_h \,\mathrm{d}\boldsymbol{x} = 0. \tag{3.7b}$$

We apply strong anchoring, i.e., Dirichlet conditions, to (3.7a). The boundary condition can be written as $\boldsymbol{Q} = \boldsymbol{Q}_s$, where $\boldsymbol{Q}_s$ is the prescribed $\boldsymbol{Q}$-tensor at

the boundary. Further details on boundary conditions are given in later chapters, where we present the specific problems to which we apply our numerical method.

Note that the choice of basis functions can greatly influence the accuracy and cost of the approximate solution. In our one-dimensional work in Chapter 4 we consider linear and quadratic basis functions. Increasing the polynomial order of the basis function should improve solution accuracy, however it also increases the number of degrees of freedom in the finite element discretisation, and thereby the overall cost of the numerical method. Any discussion comparing different choices of basis function therefore must consider both accuracy and computational cost. Following from results in Chapter 4, we choose to restrict our one-dimensional work in Chapter 5, and our two-dimensional work in Chapter 6, to quadratic basis functions. Further details on these choices of basis functions can be found in Appendix B.

3.3.1 Treatment of the mesh movement terms

We now turn our attention to the term introduced as a result of mesh movement in (3.7a), i.e.,

$$\int_{\Omega} (\nabla \cdot (\dot{\boldsymbol{x}} q_{ih}))\, v_h \, \mathrm{d}\boldsymbol{x}. \tag{3.8}$$

We could apply the product rule to (3.8) and obtain

$$\int_\Omega (\nabla \cdot (\dot{\boldsymbol{x}} q_{ih}))\, v_h \, \mathrm{d}\boldsymbol{x} = \underbrace{\int_\Omega (\nabla \cdot \dot{\boldsymbol{x}}) q_{ih}\, v_h \, \mathrm{d}\boldsymbol{x}}_{A_1} + \underbrace{\int_\Omega \dot{\boldsymbol{x}} \cdot \nabla q_{ih}\, v_h \, \mathrm{d}\boldsymbol{x}}_{B_1}. \tag{3.9}$$

Alternatively, we could integrate (3.8) by parts to obtain

$$\int_\Omega (\nabla \cdot (\dot{\boldsymbol{x}} q_{ih}))\, v_h \, \mathrm{d}\boldsymbol{x} = \underbrace{\int_{\partial\Omega_t} (\dot{\boldsymbol{x}} \cdot \mathbf{n}) q_{ih}\, v_h \, \mathrm{d}l}_{A_2} - \underbrace{\int_\Omega (\dot{\boldsymbol{x}} q_{ih}) \cdot \nabla v_h \, \mathrm{d}\boldsymbol{x}}_{B_2}. \tag{3.10}$$

Note that A_1, B_1, and B_2 are integrals over the whole domain, whilst A_2 is an integral on the boundary of our domain. With Dirichlet boundary conditions the test function v_h vanishes on the boundary and A_2, the boundary integral, will be exactly zero. It follows from (3.10) that

$$\int_\Omega (\nabla \cdot (\dot{\boldsymbol{x}} q_{ih}))\, v_h \, \mathrm{d}\boldsymbol{x} = B_2 = -\int_\Omega (\dot{\boldsymbol{x}} q_{ih}) \cdot \nabla v_h \, \mathrm{d}\boldsymbol{x}. \tag{3.11}$$

If the vector $\boldsymbol{q}_i(t)$ contains the degrees of freedom defining q_{ih} then we can express (3.9) and (3.11) respectively as

$$\begin{aligned}
\int_\Omega (\nabla \cdot (\dot{\boldsymbol{x}} q_{ih}))\, v_h \, \mathrm{d}\boldsymbol{x} &= [A_1(t, \dot{\boldsymbol{x}}_h(t)) + B_1(t, \dot{\boldsymbol{x}}_h(t))]\, \boldsymbol{q}_i(t), \\
\int_\Omega (\nabla \cdot (\dot{\boldsymbol{x}} q_{ih}))\, v_h \, \mathrm{d}\boldsymbol{x} &= [B_2(t, \dot{\boldsymbol{x}}_h(t))]\, \boldsymbol{q}_i(t),
\end{aligned}$$

where

$$\begin{aligned}
[A_1(t, \dot{\boldsymbol{x}}_h(t))]_{nm} &= \int_\Omega \nabla \cdot \dot{\boldsymbol{x}}_h(t) \phi_m(t) \phi_n(t) \, \mathrm{d}\boldsymbol{x}, \\
[B_1(t, \dot{\boldsymbol{x}}_h(t))]_{nm} &= \int_\Omega (\dot{\boldsymbol{x}}_h(t) \nabla \cdot \phi_m(t))\, \phi_n(t) \, \mathrm{d}\boldsymbol{x}, \\
[B_2(t, \dot{\boldsymbol{x}}(t))]_{nm} &= -\int_\Omega \dot{\boldsymbol{x}} \phi_m(t) \cdot \nabla \phi_n(t) \, \mathrm{d}\boldsymbol{x}.
\end{aligned}$$

Note that where we use Dirichlet boundary conditions, the integration by parts approach is more computationally efficient than the product rule approach, as we must only calculate one matrix term. For more general boundary conditions either approach can be followed, but care should be taken with the integration by parts approach where the boundary integral will no longer vanish.

3.3.2 The semi-discrete system of equations

If the vector $\boldsymbol{q}_i(t)$ contains the degrees of freedom defining q_{ih}, and the vector $\boldsymbol{u}(t)$ the degrees of freedom defining U_h, then we may express (3.7a) as the system of non-linear ordinary differential equations

$$\frac{d}{dt}(M(t)\boldsymbol{q}_i(t)) - \left[A(t, \dot{\boldsymbol{x}}_h(t)) + B(t, \dot{\boldsymbol{x}}_h(t))\right] \boldsymbol{q}_i(t) = G(t) - F(t), \qquad i = 1, \ldots, 5, \tag{3.12}$$

where

$$[M(t)]_{nm} = \int_\Omega \phi_n(t)\phi_m(t)\ \mathrm{d}\boldsymbol{x}$$

is the (time-dependent) finite element mass matrix, and

$$\begin{aligned}
[A(t, \dot{\boldsymbol{x}}_h(t))]_{nm} &= \int_\Omega \nabla \cdot \dot{\boldsymbol{x}}_h(t)\phi_m(t)\phi_n(t)\ \mathrm{d}\boldsymbol{x}, \\
[B(t, \dot{\boldsymbol{x}}_h(t))]_{nm} &= \int_\Omega \left(\dot{\boldsymbol{x}}_h(t)\nabla \cdot \phi_m(t)\right)\phi_n(t)\ \mathrm{d}\boldsymbol{x}, \\
[F(t)]_n &= \int_\Omega f(t)\phi_n(t)\ \mathrm{d}\boldsymbol{x}, \\
[G(t)]_n &= \int_\Omega \Gamma_i(t) \cdot \nabla\phi_n(t)\ \mathrm{d}\boldsymbol{x}.
\end{aligned}$$

Rearranging (3.12) we have

$$\frac{d}{dt}(M(t)\boldsymbol{q}_i(t)) = \boldsymbol{G}_i(t, \boldsymbol{q}_i(t), \boldsymbol{u}(t)), \tag{3.13a}$$

where

$$\boldsymbol{G}_i(t, \boldsymbol{q}_i(t), \boldsymbol{u}(t)) = [A(t) + B(t)]\, \boldsymbol{q}_i(t) + G(t) - F(t).$$

The discrete weak formulation of the Maxwell equation (3.7b) results in the non-linear algebraic system

$$\boldsymbol{C}(\boldsymbol{q}_i(t), \boldsymbol{u}(t)) = \boldsymbol{0}, \qquad i = 1, \ldots, 5, \tag{3.13b}$$

with

$$[\boldsymbol{C}(\boldsymbol{q}_i(t), \boldsymbol{u}(t))]_n = \int_\Omega \boldsymbol{D} \cdot \nabla \phi_n(t)\, \mathrm{d}\boldsymbol{x}.$$

The equations (3.13) form a highly non-linear differential algebraic system which is non-trivial to solve efficiently.

Chapter 4

A simple Q-tensor model in one dimension

4.1 Introduction

To our knowledge, the first studies of the use of moving mesh methods for liquid crystal problems were carried out by Ramage and Newton [61, 62] who considered a $\boldsymbol{Q}$-tensor description of a nematic liquid crystal. In [62], a static uniaxial one-dimensional model, not under the influence of an applied electric field, was considered where the orientation depends solely on the scalar Maier-Saupe order parameter, S. Minimising the free energy density gives rise to a non-linear second-order boundary value problem for S, and an appropriate choice of boundary conditions produces a single boundary layer. A series of calculations were

performed in [62] for a range of liquid crystal cell sizes, assuming fixed material properties. Therefore, while the boundary layer thickness remained constant for these experiments, the proportion of the cell covered by the boundary layer decreased as the size of the domain increased thus making the numerical computation using a fixed number of elements more challenging. The non-uniform mesh used in [62] was obtained using a well-known technique of equidistributing a positive monitor function across the available mesh elements. The monitor function considered was a scaled solution arc-length (AL); a popular monitor function which has been used previously to generate solution adaptive grids for steady and time-dependent problems [17, 39, 45, 60]. The computations in [62] indicate that solution adaptive meshes can help deliver accurate numerical results over a range of different sized domains and that the solution accuracy far exceeds that obtainable using uniform meshes with the same number of elements. The AL monitor function was subsequently used to solve a one-dimensional, time-dependent problem describing the order reconstruction of a nematic liquid crystal inside a π-cell subject to an electric field [61] (we tackle this problem ourselves in Chapter 5).

While the calculations in [62] highlight the potential use of adaptive mesh methods to solve problems in liquid crystals, a careful examination of the results reveals that the accuracy of the computed solutions deteriorates as the size of the domain increases using a fixed number of mesh elements. In addition, the optimal rate of convergence expected using linear and quadratic finite elements is

not obtained when the domain is sufficiently large and a relatively small number of mesh elements are used. In this sense the accuracy of the adaptive method is not robust to changes in the parameters defining the problem, which is clearly undesirable.

We consider the use of solution adaptive meshes which are obtained by equidistribution of an alternative monitor function which has been used to solve linear singular perturbation problems by Beckett and Mackenzie (BM) [12, 13]. The BM monitor function contains few user-defined parameters and the effects of those that do appear are well understood. In particular, it has been shown that the adaptive meshes obtained by equidistribution of the BM monitor function are well suited to singular perturbation problems in the sense that the solution accuracy is independent of the boundary layer thickness (in contrast to the meshes obtained by equidistribution of the AL monitor function). Here we carry out a detailed study of the comparative accuracy obtained using the AL and BM monitor functions for the non-linear $\boldsymbol{Q}$-tensor boundary value problem considered in [62], and explain in detail how the boundary layer thickness depends on the physical properties of the liquid crystal material (see §2.3). The BM monitor function was also used by Amoddeo *et al.* [2, 3] to solve the time-dependent order reconstruction problem considered in [61]. These authors claim that the BM monitor function produces adaptive meshes which are better suited to capturing the highly-localised behaviour of a scalar measure of biaxiality than the adaptive meshes obtained using

the AL monitor function. The detailed results which we present here clarify why this improvement in accuracy is to be expected for such time-dependent problems.

4.2 Q-tensor theory

The most commonly-used continuum model for nematic liquid crystals is the Ericksen-Leslie theory for the nematic director, a unit vector which describes the mean molecular alignment at a point in a given sample volume (see, e.g., [73] for details). Implicit in this theory are the assumptions that, everywhere in the region of interest, the material is uniaxial and the degree of order is constant. However, these assumptions are not always valid, for example, when modelling the core of a defect. Recall from §2.1 that an alternative model avoids these assumptions by using the symmetric traceless tensor, $\boldsymbol{Q}$, to characterise the liquid crystal alignment (see e.g.[28]). Recall that the second-rank tensor $\boldsymbol{Q}$ is defined by

$$\boldsymbol{Q} = \sqrt{\frac{3}{2}} \left\langle \mathbf{u} \otimes \mathbf{u} - \frac{1}{3}\mathbf{I} \right\rangle, \tag{4.1}$$

where $\langle \cdots \rangle$ represents the local ensemble average over the unit vectors $\mathbf{u}$ along the molecular axes and $\mathbf{I}$ is the identity. It has five degrees of freedom, two of which specify the degree of order, and three of which specify the angles of the principal directions.

For static problems, recall from §2.4 that the equilibrium configuration of the dependent variables can be found by solving the set of differential equations which

result from minimising the total free energy, F, of the liquid crystal sample. In the absence of an applied electric field, we obtain F from (2.6), and we have

$$F = \int_V \left(\mathcal{F}_t(\boldsymbol{Q}) + \mathcal{F}_e(\boldsymbol{Q}, \nabla\boldsymbol{Q})\right) \mathrm{d}V, \tag{4.2}$$

where $\mathcal{F}_t$ and $\mathcal{F}_e$ are the thermotropic and elastic energy terms respectively, and are defined in §2.2.

We have a set of five coupled partial differential equations to solve to obtain the equilibrium configuration of the five dependent variables. This is a challenging problem and hence for these initial investigations we consider a simplified example which still poses significant numerical challenges because of the different characteristic lengths in the problem.

4.2.1 The model problem

We now introduce the model problem which we will study in depth for the remainder of the chapter. We use cartesian co-ordinates and consider a liquid crystal sample confined between two infinite horizontal plates a distance d apart. We assume that the anchoring on the plates is fixed and that the problem is "one-dimensional", by which we mean that $\boldsymbol{Q}$ varies only in the z-direction, and is independent of x and y. We define the physical domain as $\Omega_p = z \in [0, d]$.

Recall the $\boldsymbol{Q}$-tensor theory presented in Chapter 2. In this section we assume that the material is uniaxial throughout the domain. In this case two eigenvalues

of $\boldsymbol{Q}$ are equal and there is a unique direction along which the material behaves isotropically (the material is in fact isotropic in the plane perpendicular to the director, i.e., it is transversely isotropic). This direction is defined by a unit vector, the nematic director $\mathbf{n}$, and then $\boldsymbol{Q}$ can be written as

$$\boldsymbol{Q} = \sqrt{\frac{3}{2}}\, S \left(\mathbf{n} \otimes \mathbf{n} - \frac{1}{3}\mathbf{I} \right), \tag{4.3}$$

where $S = \langle P_2(\mathbf{u} \cdot \mathbf{n}) \rangle$ is the Maier-Saupe scalar order parameter. Here P_2 is the second Legendre polynomial and comparing (2.1) and (4.3) shows that $-1/2 \leq S \leq 1$. The factor $\sqrt{3/2}$ in (2.1) and (4.3) has been chosen such that $\mathrm{tr}(\mathbf{Q}^2) = S^2$. The thermotropic energy density in (2.7) then becomes

$$\mathcal{F}_t = \frac{1}{2}A(T - T^*)\, S^2 - \frac{1}{3}B\, S^3 + \frac{1}{4}C\, S^4. \tag{4.4}$$

Note that the thermotropic polynomial $\mathcal{F}_t$ has stationary points when $dF_t/dS = 0$. Two of these are physically relevant here: $S = 0$ corresponds to the isotropic phase, and

$$S = \frac{B + \sqrt{B^2 - 4AC(T - T^*)}}{2C} \equiv S_{eq} \tag{4.5}$$

corresponds to the nematic phase. The clearing temperature, T_c, is the value of T at which the phase transition takes place. Here, the isotropic and nematic phases have the same thermotropic energy, that is, at $T = T_c$, $\mathcal{F}_t(0) = \mathcal{F}_t(S_{eq}) = 0$. From this, it can be deduced that

$$A(T_c - T^*) = \frac{2B^2}{9C} \tag{4.6}$$

which follows from the definition of S_{eq} given by (4.5) and the fact that $F_t(S_{eq}) = 0$ when $T = T_c$. If the temperature is high enough such that

$$T > T^+ \equiv T^* + \frac{B^2}{4AC},$$

then the thermotropic polynomial has only one stationary point corresponding to the isotropic phase $S = 0$. A plot of the thermotropic energy polynomial is shown in Figure 4.1 for various values of T.

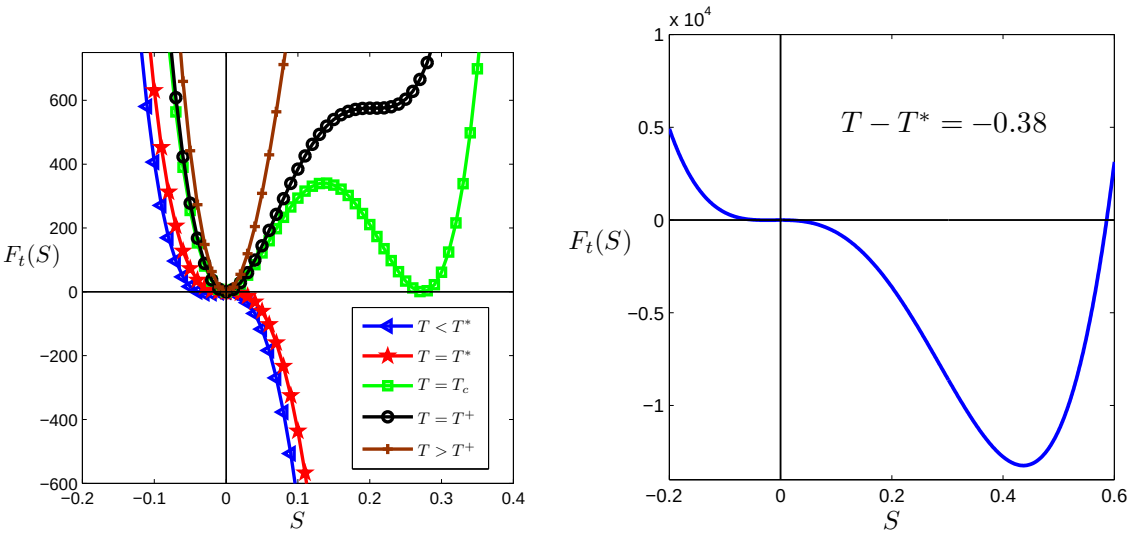

Figure 4.1: Thermotropic energy density F_t as a function of the scalar order parameter S.

To simplify the problem even further we assume that $\mathbf{n}$ is aligned parallel to

the z-axis so that (4.3) takes the form

$$\boldsymbol{Q} = \sqrt{\frac{3}{2}}\, S \begin{bmatrix} -\frac{1}{3} & 0 & 0 \\ 0 & -\frac{1}{3} & 0 \\ 0 & 0 & \frac{2}{3} \end{bmatrix}.$$

The $\boldsymbol{Q}$-tensor now depends only on the scalar order parameter S and, in this simplified case, the elastic energy in (2.8) becomes

$$\mathcal{F}_e = \frac{1}{6}(2L_1 + L_2)\left(\frac{\mathrm{d}S}{\mathrm{d}z}\right)^2. \tag{4.7}$$

The total free energy can therefore be minimised (and hence the equilibrium configuration identified) by solving a single Euler-Lagrange equation for S, subject to suitable boundary conditions.

For computational purposes, it is useful to non-dimensionalise lengths with respect to the nematic coherence length $\zeta = \sqrt{9CL_2/2B^2}$. Following [72], we also make a second transformation to remove the material specific constants B and C, using relationship (4.6) to simplify further. Rescaling with

$$\bar{z} = \frac{1}{\zeta}\, z, \qquad \bar{S} = \frac{3C}{2B}\, S,$$

the problem is now one of minimising the sum of the non-dimensional energy densities

$$\bar{F}_t = \frac{\chi}{2}\bar{S}^2 - \bar{S}^3 + \frac{1}{2}\bar{S}^4, \qquad \bar{F}_e = \frac{\psi}{2\zeta^2}\left(\frac{\mathrm{d}\bar{S}}{\mathrm{d}\bar{z}}\right)^2, \tag{4.8}$$

where

$$\chi = \frac{T - T^*}{T_c - T^*}, \quad \text{and} \quad \psi = \frac{2}{3}\left(\frac{L_1}{L_2} - 1\right) + 1.$$

For the rest of this chapter, we will work with these dimensionless quantities, but will omit the overbars for ease of notation. Note that the problem domain has also been scaled in accordance with the above non-dimensionalisation; that is, we now have the scaled domain $\Omega_s = z \in [0, d_s]$ when the true length of the physical domain is d (with $d = d_s\zeta$).

With this scaling, the equilibrium nematic scalar order parameter (4.5) becomes

$$S_{eq} = \left(3 + \sqrt{9 - 8\chi}\right)/4.$$

When $\chi > 1$ (that is, $T > T_c$), the minimum of $\mathcal{F}_t$ corresponds to the isotropic phase ($S = 0$), and for $\chi < 1$, its minimum corresponds to the nematic phase ($S = S_{eq}$). There is a first-order phase transition when $\chi = 1$ ($T = T_c$). When $\chi < 0$ ($T < T^*$), the isotropic phase is unstable whereas the nematic phase is stable. The nematic phase becomes unstable when $\chi > 9/8$ ($T > T^+$).

All of the numerical experiments in §4.4 are carried out with $\chi \approx -0.3455$ and $\psi \approx 3.0278$. These values are commensurate with the temperatures and material constants used in [62], so the results can be directly compared. We note that, for these values, the associated nematic coherence length is $\zeta \approx 4nm$ and the (scaled) equilibrium value is $S_{eq} \approx 1.6075$. This corresponds to an unscaled value of approximately 0.4396 which lies within the admissible range for the scalar order parameter.

As in the next section we will make use of results from singular perturbation

analysis, we make a final scaling of the z-co-ordinate axis to z/d_s so that the computational domain is $\Omega_c \equiv z \in [0, 1]$.

With the above scalings, the governing differential equation for $S(z)$ is the Euler-Lagrange equation

$$\frac{\partial F}{\partial S} - \frac{d}{dz}\left(\frac{\partial F}{\partial S_z}\right) = 0 \qquad \leftrightarrow \qquad \frac{\psi}{{d_s}^2}\frac{\mathrm{d}^2 S}{\mathrm{d}z^2} = \chi S - 3S^2 + 2S^3, \quad z \in (0, 1). \quad (4.9)$$

We solve (4.9) subject to the boundary conditions $S = 0$ at $z = 0$ and $S = S_{eq}$ at $z = 1$. That is, at the left edge of the domain the sample is isotropic, and at the right edge S has reached its equilibrium value of S_{eq}. With these boundary conditions the solution develops a steep boundary layer close to $z = 0$ and this provides a suitable challenge for the proposed algorithm. Similar steep layers are present in physically meaningful situations such as domain walls and close to solution defects.

4.3 Estimates of the boundary layer component of the solution

It is not immediately obvious why (4.9) is a singularly perturbed boundary value problem. However, if we let $\varepsilon = \sqrt{\psi}/d_s$, then (4.9) can be written in the more

familiar form

$$-\varepsilon^2 \frac{\mathrm{d}^2 S}{\mathrm{d}z^2} + (\chi S - 3S^2 + 2S^3) = 0, \quad z \in (0,1), \quad S(0) = 0 \quad \text{and} \quad S(1) = S_{eq}. \tag{4.10}$$

If $S_0(z)$ denotes the solution of the reduced problem which is defined by setting $\varepsilon = 0$ in (4.10), then

$$b(z, S_0(z)) \equiv \chi S_0 - 3S_0^2 + 2S_0^3 = 0, \quad z \in (0,1). \tag{4.11}$$

As mentioned earlier, we are interested in the temperature regime $T < T^*$ and hence $\chi < 0$. Under these circumstances we can factorise $b(z, S_0(z))$ so that

$$2S_0(S_0 - \tilde{S})(S_0 - S_{eq}) = 0,$$

where $\tilde{S} < 0$. The reduced problem therefore has three constant solutions: $S_0 = 0$, $S_0 = \tilde{S}$, and $S_0 = S_{eq}$. Only $S_0 = \tilde{S}$ and $S_0 = S_{eq}$ are stable reduced solutions in the sense that, for both,

$$b_S(z, S_0) = \chi - 6S_0 + 6S_0^2 > \lambda^2 > 0, \quad z \in (0,1).$$

Clearly, the solution $S_0 = S_{eq}$ satisfies the right-hand boundary condition of the full problem (4.10). At the other boundary we have

$$-\int_v^0 b(z, S_{eq} + t)\, \mathrm{d}t = -\int_v^0 2t(S_{eq} + t - \bar{S})(S_{eq} + t)\, \mathrm{d}t > 0, \quad \forall\, t \in [-S_{eq}, 0). \tag{4.12}$$

The solution $S_0 = S_{eq}$ is therefore said to have a stable boundary layer at $z = 0$ [46, 56]. We seek a solution of the full problem (4.10) that, away from the boundary

layer at $z = 0$, is close to $S_0 = S_{eq}$. Thus we may take $S_0 = S_{eq}$ as the zeroth-order smooth component in an asymptotic expansion of $S(z)$. Let $v(z)$ denote the zeroth-order boundary layer term. The following Lemma, which is equivalent to Lemma 2.3 in [46], establishes the existence of such a boundary layer term and gives bounds on its derivatives.

Lemma 4.3.1 *Let* $\lambda_0 = \sqrt{b_S(0, S_{eq})}$. *Then, for each* $\delta \in (0, \lambda_0)$, *there exists a positive constant* C_δ *such that*

$$\left|\frac{\mathrm{d}^k v}{\mathrm{d}z^k}\right| \leq C_\delta \varepsilon^{-k} e^{-((\lambda_0 - \delta)z/\varepsilon)}, \qquad k = 0, 1, \ldots, 4. \tag{4.13}$$

A zeroth-order asymptotic solution of (4.10) is therefore given by

$$S_{asym} = S_{eq}(1 - e^{-\lambda_0 z/\varepsilon}). \tag{4.14}$$

Using this approximate solution it is possible to get an idea of how the scaled boundary layer thickness varies in terms of the physical parameters defining our model. Here we will define the estimated boundary layer thickness as the distance z_{bl} such that $S_{asym}(z_{bl}) = \Phi S_{eq}$, where typically $\Phi \approx 0.99$. It follows from (4.14) that

$$z_{bl} = -\frac{\varepsilon}{\lambda_0} \ln(1 - \Phi). \tag{4.15}$$

We can therefore see that the scaled boundary layer thickness is an increasing function of ε. The scaled boundary layer thickness is therefore a decreasing function of the physical size of the liquid crystal cell d, but is an increasing function of the

ratio of the elastic constants L_1/L_2. The value of λ_0 depends on the temperature through the parameter χ. A plot of z_{bl} versus χ for $\varepsilon = 0.01$ ($d \approx 0.7$ microns) is shown in Figure 4.2 from which we can deduce that the boundary layer thickness is a increasing function of temperature in this case.

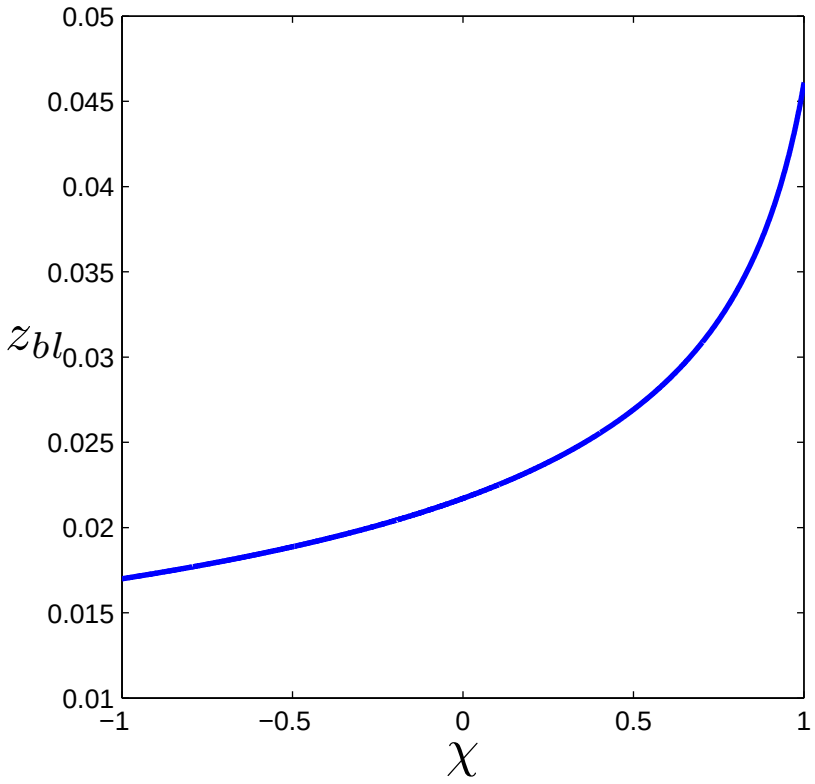

Figure 4.2: Variation of boundary layer thickness z_{bl} in terms of the parameter χ with $\varepsilon = 0.01$, and $d \approx 0.7$ microns.

4.4 Adaptive grid generation

4.4.1 Grid equidistribution

Guided by the singular perturbation analysis of the previous section, we now consider the adaptive numerical solution of (4.10). In one-dimensional moving mesh methods, the mesh is usually updated via a mesh generating equation based on the equidistribution of a positive monitor function. That is, grid points are located in order to limit some measure of the solution error by distributing it equally across each subinterval. For this type of adaptivity, the new mesh is usually constructed as the image under a suitably defined mapping of a fixed mesh over an auxiliary domain. We assume a uniform mesh

$$\xi_i = i/N, \quad i = 0, 1, \ldots, N, \tag{4.16}$$

where N is a positive integer, is imposed on the computational domain Ω_c, and denote the corresponding mesh in the physical domain Ω_p by

$$\Delta_N \equiv \{a = z_0 < z_1 < \ldots < z_{N-1} < z_N = b\}. \tag{4.17}$$

With this notation,

$$z = z(\xi, t) \colon [0, 1] \to [a, b], \quad \xi \in \Omega_c = [0, 1],$$

denotes a mapping from a computational coordinate ξ to a physical coordinate z. Given a function representing a particular physical quantity from a prob-

lem, $\mathcal{T}(z,t)$ say, and an associated strictly positive integrable *monitor function* $\rho(\mathcal{T}(z,t))$, the one-dimensional *equidistribution principle* can be expressed as

$$\int_a^{z_i(\xi,t)} \rho(\mathcal{T}(s,t))\,\mathrm{ds} = \xi_i \int_a^b \rho(\mathcal{T}(z,t))\,\mathrm{dz} = \frac{i}{N}\int_a^b \rho(\mathcal{T}(z,t))\,\mathrm{dz}, \quad i = 0,1,\ldots,N. \tag{4.18}$$

Many monitor functions depend on derivatives of the exact solution and hence these have to be approximated using derivatives of the numerical solution. Furthermore, most monitor functions are not easily integrable functions and so the practical computation of equidistributing meshes requires some form of quadrature to approximate the integrals. A simple way of finding an approximation of an equidistributed grid is to find the mesh that equidistributes a piecewise constant approximation of $\rho(\mathcal{T}(z,t))$. A popular method for doing so is the so-called de Boor algorithm [25]. If (4.17) denotes a partition of the domain Ω_p using N elements, and $\rho(\mathcal{T}(z,t)) > 0$ is a continuous function on Ω_c, then a grid is said to be equidistributed if $\rho(\mathcal{T}(z,t))$ is evenly distributed over the mesh elements in the sense that

$$\int_{z_{i-1}}^{z_i} \rho(\mathcal{T}(z,t))\,\mathrm{d}z = \frac{1}{N}\int_a^b \rho(\mathcal{T}(z,t))\,\mathrm{d}z, \qquad i = 1,\ldots,N. \tag{4.19}$$

In particular, if

$$\Delta_N^{(k)} = \{a = z_0^{(k)} < z_1^{(k)} < \ldots < z_{N-1}^{(k)} < z_N^{(k)} = b\}, \quad h_i^{(k)} = z_i^{(k)} - z_{i-1}^{(k)}, \quad 1 \le i < N,$$

and $\rho_i^{(k)}(\mathcal{T}(z,t))$ is a constant on $(z_{i-1}^{(k)}, z_i^{(k)})$, then inverse linear interpolation can be used to find the mesh

$$\Delta_N^{(k+1)} = \{a = z_0^{(k+1)} < z_1^{(k+1)} < \ldots < z_{N-1}^{(k+1)} < z_N^{(k+1)} = b\}$$

which exactly equidistributes $\rho^{(k)}(\mathcal{T}(z,t))$ in the sense that

$$\int_{z_{i-1}^{(k+1)}}^{z_i^{(k+1)}} \rho^{(k)}(\mathcal{T}(z,t))\,\mathrm{d}z = \frac{1}{N}\sum_{i=1}^{N} \rho_i^{(k)}(\mathcal{T}(z,t))h_i^{(k)}, \quad i = 1,\ldots,N.$$

Once the mesh has been updated, a new approximation can be obtained using this mesh. It is then possible to define a new piecewise constant monitor function that can be used to update the mesh further. It is immediately clear that in regions of the domain where the monitor function is large, the mesh spacing will be small, and in regions of the domain where the monitor function is small, the mesh spacing will be large. One can consider the monitor function as measure of the local difficulty in solving the underlying physical problem, whose purpose is to generate a grid that adapts to features of the solution. The approximate equidistribution strategy presented above was used in [25] to generate solution-adaptive meshes to solve two-point boundary value problems, and the algorithm will form part of our iterative solution method.

The choice of an appropriate monitor function is essential to the success of the use of the equidistribution principle. In [62] the authors considered the scaled

solution arc-length (AL) monitor function

$$\rho(S_N(z)) = \sqrt{\mu + \left(\frac{\mathrm{d}S_N}{\mathrm{d}z}\right)^2}, \tag{4.20}$$

where μ is a user-prescribed scaling parameter. Using this monitor function to solve (4.10), and assuming that $S_N \approx S_{asym}$, we see that, external to the boundary layer region, $\mathrm{d}S_N/\mathrm{d}z \approx 0$ and $\rho \approx \sqrt{\mu}$, whereas in the boundary layer region, $\rho \approx S_{eq}\lambda_0\varepsilon^{-1}e^{-\lambda_0 z/\varepsilon}$.

As mentioned in the introduction to this chapter, the AL monitor function has some shortcomings when applied to the resolution of exponential-like boundary layers. We therefore consider the alternative monitor function

$$\rho(S_N(z)) = \alpha + \left(\left|\frac{\mathrm{d}S_N}{\mathrm{d}z}\right|\right)^{1/m}, \quad \text{where} \quad \alpha = \int_0^1 \left(\left|\frac{\mathrm{d}S_N}{\mathrm{d}z}\right|\right)^{1/m} \mathrm{d}z, \tag{4.21}$$

and m is a positive parameter. Beckett and Mackenzie (BM) used a similar monitor function to solve singularly perturbed reaction-diffusion boundary value problems using finite difference methods [12] and Galerkin finite element methods [13]. As the main difficulty in solving (4.10) is resolving the exponential boundary layer, where the solution first derivative is similar to the second derivative, we have, for simplicity, used the first derivative of the numerical solution in the definition of the monitor function whereas the second derivative was used in [12, 13].

The positive factor α in (4.21) acts as a lower bound, or floor, on the monitor function and its role is to prevent mesh starvation in areas of the domain where $\mathrm{d}S_N/\mathrm{d}z$ is close to zero. For our model problem, this is the region external to the

boundary layer at $z = 0$. It is important to note that α is not a user-specified parameter as its value is determined a posteriori from the numerical approximation itself. This is in contrast to the parameter μ in the definition of the AL monitor function. It has been shown in [13] that the meshes obtained by equidistribution of the BM monitor are balanced in the sense that approximately half the available mesh elements are automatically located external to the boundary layer and the remaining half are located within the boundary layer region.

If the parameter m in (4.21) is greater than 1, then its effect is to smooth potentially large variations in the value of $\mathrm{d}S_N/\mathrm{d}z$. Increasing m will lead to fewer mesh elements in the boundary layer region, leaving more to be deployed to cover the rest of the domain. This can potentially lead to improved accuracy for problems with boundary layers and additional solution features that need to be resolved in the interior of the domain. Such a situation arises in the order reconstruction problem considered in [2, 3, 9, 61] (and in Chapter 5 of this thesis). This is one of the reasons put forward by Amoddeo *et al.* [2, 3] for preferring the use of the BM monitor with $m = 2$, as opposed to the AL monitor function: the latter gives rise to a mesh similar to that obtained with the BM monitor function with $m = 1$.

To provide additional guidance on an appropriate choice of the parameter m to resolve a function with a boundary layer, we have the following result (equivalent to Theorem 7 from [12]).

Theorem 4.4.1 *Let $v(z) = e^{-(\lambda_0 z)/\varepsilon}$, and let $I_p v(z)$ denote the piecewise polynomial interpolant of $v(z)$ of degree p on each mesh element. Then, on the mesh that exactly equidistributes the BM monitor function, there exists a constant C which is independent of ε and N such that*

$$\max_{z\in[z_{i-1},z_i]} |v(z) - I_p v(z)| < C \begin{cases} N^{-(p+1)}, & m \geq p+1, \\ N^{-m}, & m < p+1 \end{cases} \tag{4.22}$$

for $i = 1, 2, \ldots, N$.

Therefore, to obtain the optimal rate of convergence using polynomial basis functions of degree p, it is necessary that $m \geq p+1$. In particular, if linear elements are used, then we need $m \geq 2$. This result explains why sub-optimal rates of convergence were observed using the AL monitor function in [62], where both linear and quadratic elements were used. Amoddeo *et al.* [2, 3] use quadratic elements in their simulations using the BM monitor with $m = 2$. This combination clearly would also lead to sub-optimal rates of convergence if it were used to resolve a function with an exponential boundary layer.

4.4.2 Adaptive solution procedure

The generation of the adaptive grid requires the global equidistribution principle (4.19) and the monitor function (4.21) be discretised. The resulting set of equations could then be coupled with the finite element discretisation of (4.10) to give a nonlinear system for Δ_N and S_N. However, this system is large and expensive to solve,

and also dictates that the grid and numerical solution be evaluated with the same level of accuracy. Thus, a popular alternative is to decouple the calculation of the grid from the finite element solution, and solve the two sets of equations in an iterative manner. Here we will use an algorithm proposed by Kopteva and Stynes [45] which is based on a modified version of the de Boor algorithm described in (4.4.1).

To test how close the grid is to being equidistributed at iteration k we first define the two quantities

$$E_i{}^{(k)} = \rho_i{}^{(k)} h_i{}^{(k)}, \quad i = 1, \ldots, N \quad \text{and} \quad I^{(k)} = \sum_{i=1}^{N} \rho_i{}^{(k)} h_i{}^{(k)}.$$

The iterative remeshing algorithm (described below) is executed until

$$\max_{1 \le i \le N} \{E_i{}^{(k)}\} \le C_0 \frac{I^{(k)}}{N}, \tag{4.23}$$

where $h_i^{(k)} = z_i^{(k)} - z_{i-1}^{(k)}$ and C_0 is a user defined constant. The choice of C_0 determines a balance between computational efficiency and solution accuracy. For example, with a large value of C_0 it is easier to find an approximately equidistributed grid, but this might be at the expense of solution accuracy. Conversely, using a value of C_0 very close to one may require a large number of iterations to converge but many of these iterations will be superfluous and have very little effect on the solution accuracy. This test for convergence and guidance on suitable choices of C_0 are discussed in more detail in [21]; here we set $C_0 = 1.1$.

The overall algorithm is as follows:

1. Set the initial mesh $\Delta_N^{(0)}$ to be a uniform grid of N cells. Set the initial guess $S_N^{(0)}$ to be the linear function interpolating the boundary values $S = 0$ and $S = S_{eq}$. Set $k = 0$.

2. Obtain the finite element approximation $S_N^{(k)}$ using the mesh $\Delta_N^{(k)}$ and use Newton's method to solve the system of non-linear algebraic equations.

3. Set $h_i^{(k)} = z_i^{(k)} - z_{i-1}^{(k)}$ for each i and calculate $\rho_i^{(k)}$, $E_i^{(k)}$ and $I^{(k)}$.

4. If

$$\max_{1 \le i \le N} \{E_i{}^{(k)}\} \le C_0 \frac{I^{(k)}}{N},$$

then go to Step 6. Otherwise, continue to Step 5.

5. Use the de Boor algorithm [25] to find the mesh $\Delta_N^{(k+1)}$ that equidistributes $\rho^{(k)}$. Set $k = k + 1$ and go to Step 2.

6. Output the final mesh $\Delta_N = \Delta_N^{(k)}$ and solution $S_N = S_N^{(k)}$ and stop.

4.5 Numerical results

In this section we present some numerical results concerned with solving (4.10) using the algorithm in the previous section. Specifically, we compare computed solutions for three values of d giving boundary layers of different thicknesses, using ψ and χ as in (4.2.1). As the analytical solution $S(z)$ of (4.10) is not available

in a convenient form, we have instead compared the computed solutions with the numerical solution obtained using quadratic elements and a fine non-uniform adaptive mesh which equidistributes the BM monitor function using $N = 2048$ elements. We will use $S_f(z)$ to denote this fine grid approximation to $S(z)$ with the assumption that

$$|S_f(z) - S(z)| \ll |S_f(z) - S_N(z)|.$$

As an estimate of the true L_∞ error we have calculated

$$\|e^N\|_{L_\infty(0,1)} \equiv \max_{i=1,\ldots,N} \left(\max_{1\le j\le 11} |S_f(z_{ij}) - S_N(z_{ij})| \right), \tag{4.24}$$

where the error sampling points are taken to be

$$z_{ij} = z_{i-1} + \frac{j-1}{10}(z_i - z_{i-1}), \qquad i = 1,\ldots,N, \quad j = 1,\ldots,11. \tag{4.25}$$

Since the sampling points z_{ij} will not in general coincide with any fine background grid points, the solution $S_f(z_{ij})$ is interpolated using the local solution defined on the fine grid element that includes the point z_{ij}.

We also estimate the maximum error computed at grid nodes

$$\|\mathbf{e}^N\|_{l_\infty} = \max_{i=0,\ldots,N} |S_f(z_i) - S_N(z_i)|. \tag{4.26}$$

Figures 4.3 – 4.8 show a typical set of numerical results obtained for three different values of d. The examples shown have been obtained using quadratic basis functions on adaptive meshes with $N = 32$ elements using the BM monitor

function with the parameter $m = 3$. The solution and computed error at the grid nodes are denoted by circles. In the error plots the solid line denotes the linear interpolant of the error computed at the set of sampling points $\{z_{ij}\}$ defined in (4.25). In each solution plot we have also included the zeroth-order asymptotic solution S_{asym}. The vertical dotted line denotes the location of the estimated boundary layer thickness z_{bl}. We note that, for each value of d, the boundary layer nature of the solution is captured well by the adaptive meshes even though the relative boundary layer thickness decreases by an order of magnitude as we increase d. From the plots of the solution errors, we see that the errors are largest in the boundary layer region or close to the boundary layer edge. Note however that the maximum solution error is relatively insensitive to the value of d. We can also see that the computed solutions are considerably more accurate at the grid points defining the finite element mesh in comparison to the error within mesh elements.

To quantify solution accuracy, in Table 4.1 we first present the L_∞ and l_∞ norms of the error obtained using linear elements and the BM monitor function with $m = 2$. The table also includes the number of iterations required for the remeshing algorithm to converge according to the condition (4.23) with $C_0 = 1.1$ and estimates of the rates of convergence in each norm. We observe that the L_∞ norm of the error converges at the rate $\mathcal{O}(N^{-2})$, which is the optimal rate expected using linear elements. More importantly, the accuracy obtained

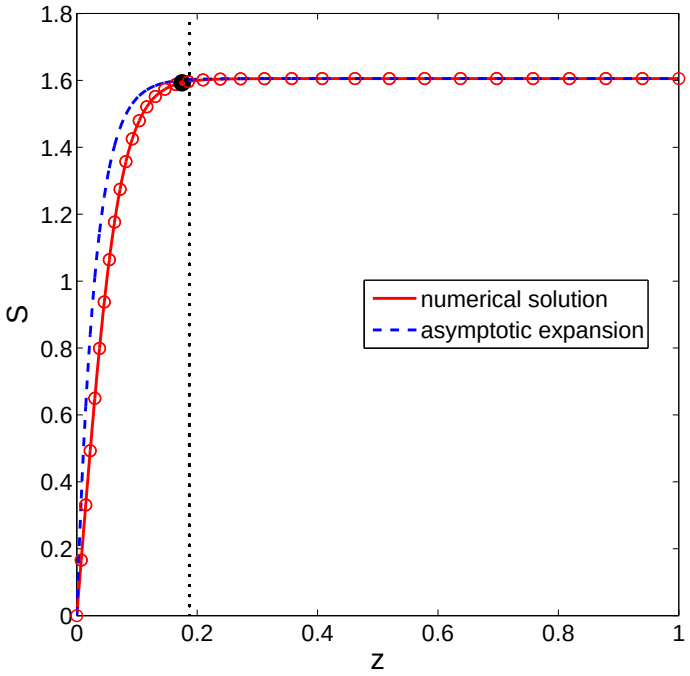

Figure 4.3: Numerical solution for $d = 0.1$ microns.

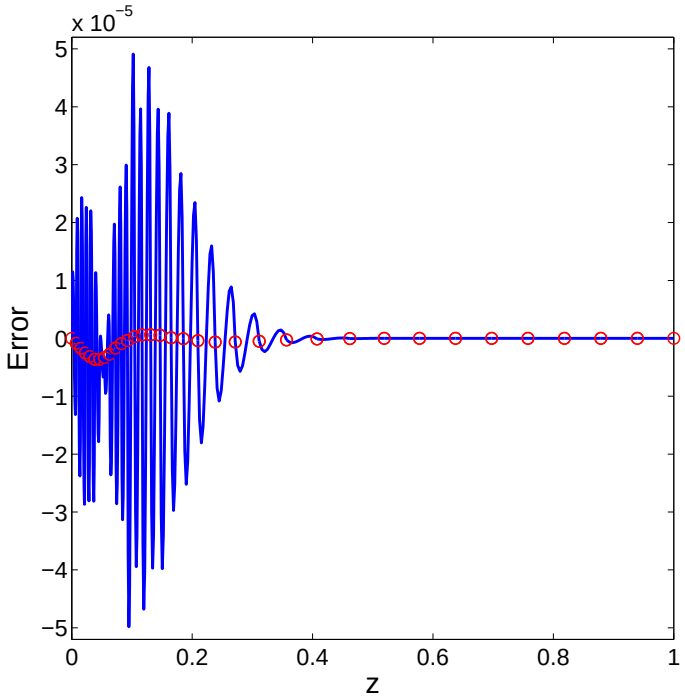

Figure 4.4: Error for $d = 0.1$ microns.

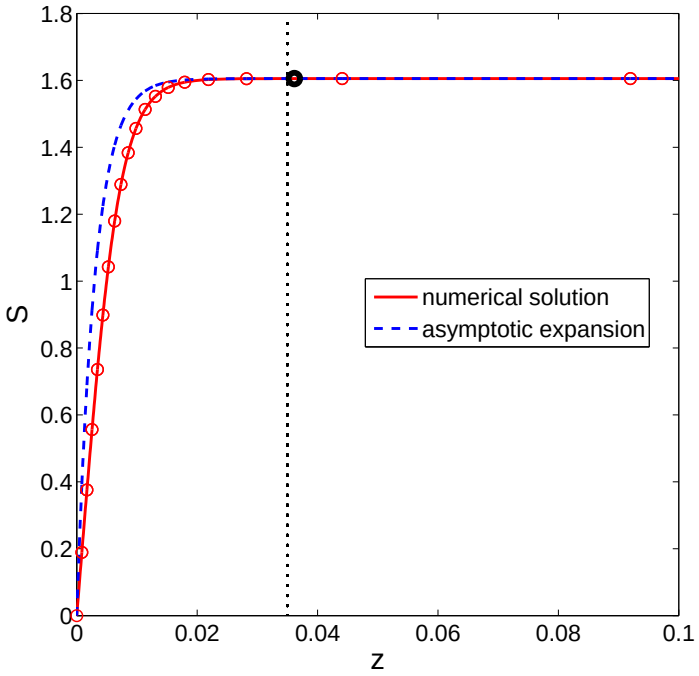

Figure 4.5: Numerical solution for $d = 1$ micron.

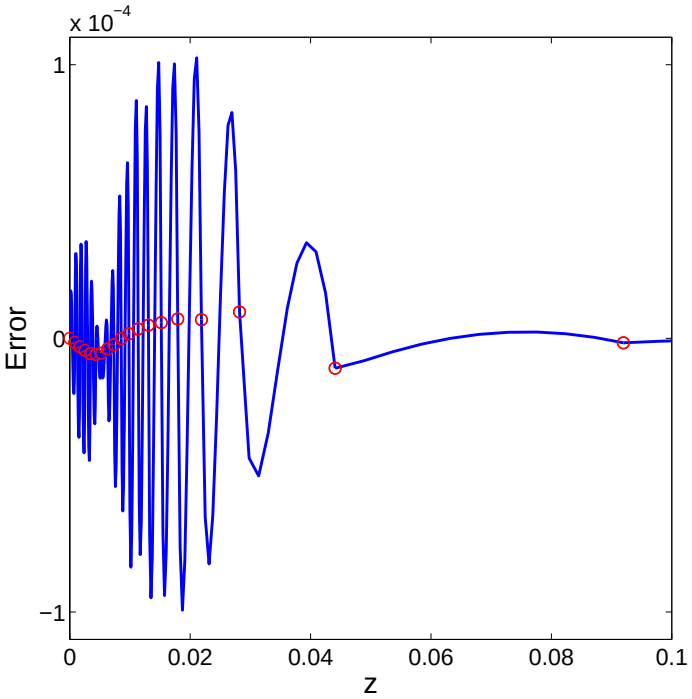

Figure 4.6: Error for $d = 1$ micron.

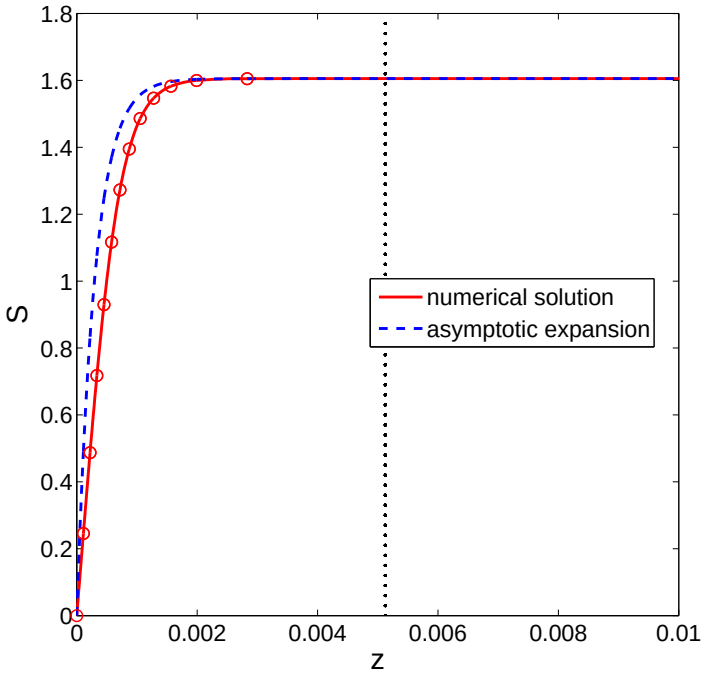

Figure 4.7: Numerical solution for $d = 10$ microns.

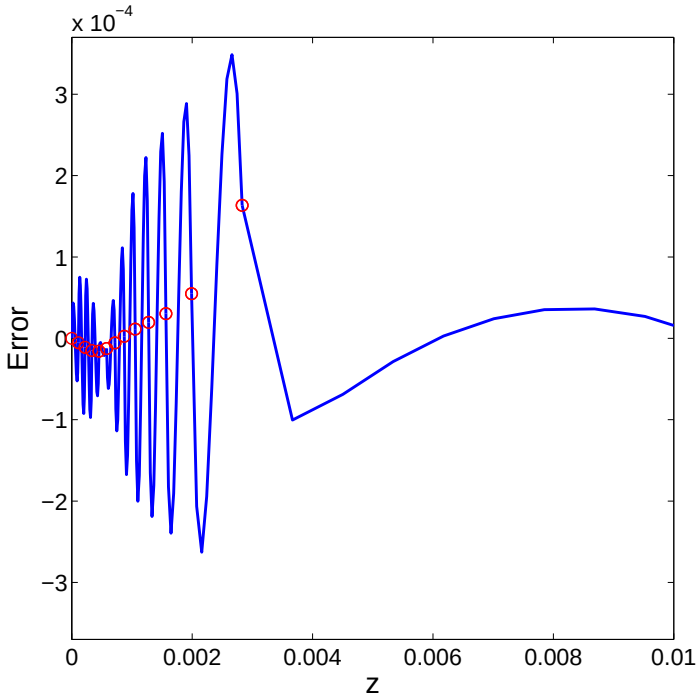

Figure 4.8: Error for $d = 10$ microns.

appears to be robust to changes in the physical size of the liquid crystal cell d. Similar uniform convergence behaviour has been established theoretically for Galerkin finite element approximations to linear reaction-diffusion problems using the BM monitor function in [13]. We can also see that the number of iterations of the remeshing algorithm required to find the adaptive mesh depends only mildly on d and N. Table 4.1 also shows that the maximum nodal error behaves in a similar fashion to the L_∞ error.

The analogous results obtained using quadratic elements are shown in Table 4.2. The convergence rate for quadratics appears to be $\mathcal{O}(N^{-3})$ in the L_∞ norm and the accuracy is again robust to changes in d. The numbers of iterations required to find the adaptive meshes are comparable to those used with linear elements. The convergence rate of the error at grid nodes appears to be $\mathcal{O}(N^{-4})$. A similar convergence rate was observed using the AL monitor function in [62]. It is well known that the finite element method can exhibit nodal superconvergence, when the numerical solution at node points is much more accurate than at intermediate points. Theoretical results in this direction go back to Douglas and Dupont [30] who showed that for linear two-point boundary value problems, the standard Galerkin approximation using polynomial elements of degree p converges in the L_∞ norm at $\mathcal{O}(N^{-(p+1)})$, whereas the solution at the grid nodes converges at $\mathcal{O}(N^{-2p})$. These convergence rates are consistent with our experiments in that there is no indication of this with linear elements, but with quadratic elements,

the solution converges more quickly at the grid nodes.

We have also compared the accuracy of the computed solutions using adaptive meshes to those obtained using a fine uniform mesh. Figure 4.9 and Figure 4.10 show the results obtained using quadratic elements and adaptive meshes using the BM monitor function with $m = 3$ and a fine uniform mesh with $N = 10^5$ elements. It is clear that the difference between the solution obtained using the fine adaptive grid and the fine uniform grid is very small thus justifying the use of the fine adaptive grid solution to estimate solution errors. Furthermore, we can see that in the most challenging case where $d = 10$, we obtain comparable accuracy using an adaptive grid with $N = 256$ elements compared to a uniform mesh using $N = 10^5$ elements, thus highlighting the efficiency gains from using adaptive meshes.

We now consider the effects of varying the parameter m in the BM monitor function. In Figure 4.11 and Figure 4.12 we present the computed errors measured in the L_∞ and l_∞ norms using linear and quadratic basis functions on meshes with $N = 64$ elements and $d = 1$. We can see that the minimum error appears to occur close to $m = 2$ using linear elements and somewhere between $m = 3$ and $m = 4$ for quadratic elements. Moreover, the error increases more quickly for values of m smaller than the optimal value. Theorem 4.4.1 indicates that if $m \geq p + 1$, then the approximation will converge at the optimal rate with respect to N. This is consistent with our findings, namely that the optimal rates of convergence, and best approximations, are attained with $m \geq 2$ for linears and $m \geq 3$ for quadratics.

Although we have presented results here only for the case $d = 1$, we observe similar behaviour when $d = 0.1$ and $d = 10$ microns.

The result in Theorem 4.4.1 assumes that the mesh is obtained by exactly equidistributing the BM monitor function. In practice, the derivatives appearing in (4.21) are those of the numerical solution. To give some indication of the accuracy of the computed solutions using the adaptive algorithm, in Figure 4.13 and Figure 4.14 we compare the results to those obtained using the mesh obtained from equidistribution of the fine grid solution $S_f(z)$. This mesh we refer to as the exactly equidistributed mesh. It is clear that the differences in the computed solutions using the different meshes are small and tend to zero as the number of elements is increased as expected due to the reduction of the error in the estimation of the solution derivatives and from approximating the continuous monitor function by a piecewise constant approximation.

We can also compare the accuracy obtained using the BM monitor function to that obtained using the AL monitor function. Figure 4.15 and Figure 4.16 show the errors in the L_∞ and l_∞ norms with linear basis functions. It can be seen that, for all values of d, the error using the BM monitor is smaller than the error obtained using the AL monitor with the same number of mesh points. Note also that the errors using the AL monitor are very sensitive to the size of the physical domain d. In particular, we see that for a fixed number of grid points the errors increase as d is increased. Figure 4.17 and Figure 4.18 show equivalent plots for quadratic

basis functions. Similar observations can be made as for the linear case, with BM outperforming AL, both in terms of accuracy and robustness to variations in d.

Having established that the use of the BM monitor function can improve accuracy (for a fixed number of grid points), we now consider the overhead associated with the computation of the adaptive grid which should be taken into account when assessing the overall efficiency of the approach. To illustrate this we look at the CPU time required using the BM and AL monitor functions to solve our problem to a given degree of accuracy in the L_∞ norm using quadratic basis functions. It has been shown that uniform meshes perform poorly in comparison to those obtained using the AL monitor function [61] and hence we will not consider uniform meshes here as they are unlikely to be competitive. The number of elements and CPU times required are listed in Table 4.3. The results indicate that BM outperforms AL in each case and that the relative efficiency of BM improves significantly as higher accuracy is required and as d is increased.

	$d = 0.1$			$d = 1$			$d = 10$		
N	$\|e^N\|_{L_\infty}$	$\|\mathbf{e}^N\|_{l_\infty}$	it	$\|e^N\|_{L_\infty}$	$\|\mathbf{e}^N\|_{l_\infty}$	it	$\|e^N\|_{L_\infty}$	$\|\mathbf{e}^N\|_{l_\infty}$	it
	Rate	Rate		Rate	Rate		Rate	Rate	
8	3.172e-02	9.487e-03	2	5.224e-02	4.242e-02	10	6.632e-02	4.540e-02	25
	-	-		-	-		-	-	
16	9.703e-03	2.867e-03	2	1.300e-02	1.213e-02	6	1.490e-02	1.490e-02	22
	1.71	1.72		2.01	1.81		2.15	1.61	
32	2.818e-03	7.884e-04	1	3.398e-03	2.675e-03	3	3.670e-03	3.097e-03	110
	1.78	1.86		1.94	2.18		2.02	2.27	
64	7.044e-04	1.992e-04	1	8.057e-04	5.300e-04	3	8.161e-04	7.157e-04	20
	2.00	1.98		2.08	2.34		2.17	2.11	
128	1.733e-04	5.005e-05	1	2.024e-04	1.257e-04	4	2.120e-04	1.707e-04	3
	2.02	1.99		1.99	2.08		1.94	2.07	
256	4.291e-05	1.260e-05	1	5.073e-05	3.145e-05	2	5.231e-05	4.405e-05	4
	2.01	1.99		2.00	2.00		2.02	1.95	

Table 4.1: Errors, convergence rate, and iterations required (it) using linear basis functions and the BM monitor function with $m = 2$.

	$d = 0.1$			$d = 1$			$d = 10$		
N	$\|e^N\|_{L_\infty}$	$\|\mathbf{e}^N\|_{l_\infty}$	it	$\|e^N\|_{L_\infty}$	$\|\mathbf{e}^N\|_{l_\infty}$	it	$\|e^N\|_{L_\infty}$	$\|\mathbf{e}^N\|_{l_\infty}$	it
	Rate	Rate		Rate	Rate		Rate	Rate	
8	2.999e-03	1.108e-03	1	2.519e-02	1.100e-02	5	8.495e-02	8.4949e-02	9
	-	-		-	-		-	-	
16	3.221e-04	6.585e-05	1	1.376e-03	2.493e-04	5	2.864e-03	6.312e-04	11
	3.21	4.07		4.19	5.46		4.89	7.07	
32	5.014e-05	3.635e-06	1	1.230e-04	1.047e-05	3	2.203e-04	3.339e-05	6
	2.68	4.18		3.48	4.57		3.70	4.24	
64	6.026e-06	2.389e-07	1	1.612e-05	5.246e-07	2	1.882e-05	2.500e-06	10
	3.06	3.93		2.93	4.31		3.55	3.73	
128	7.930e-07	1.522e-08	1	1.726e-06	3.995e-08	2	2.122e-06	3.494e-07	4
	2.93	3.97		3.22	3.71		3.14	2.83	
256	9.819e-08	9.779e-10	1	2.119e-07	2.211e-09	2	2.538e-07	2.345e-08	3
	3.01	3.96		3.02	4.17		3.06	3.90	

Table 4.2: Errors, convergence rate, and iterations required (it) using quadratic basis functions and the BM monitor function with $m = 3$.

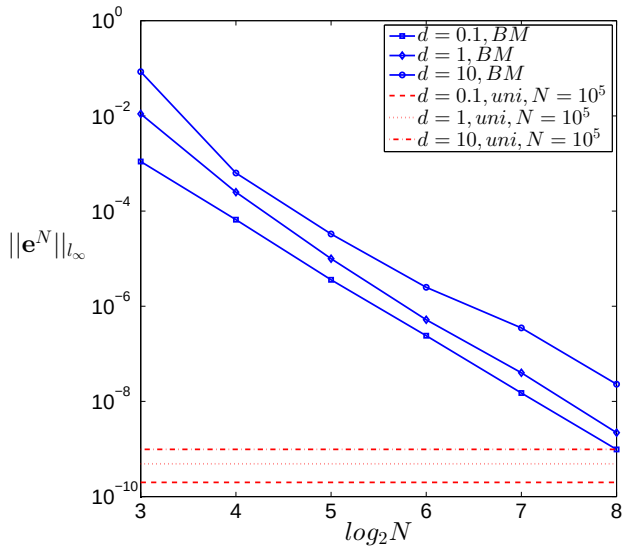

Figure 4.9: $\|\mathbf{e}^N\|_{l_\infty}$ obtained using adaptive meshes and a fine uniform mesh.

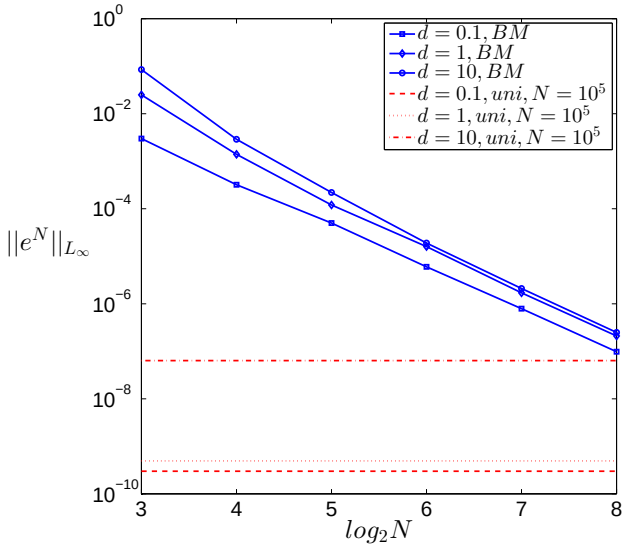

Figure 4.10: $\|e^N\|_{L_\infty}$ obtained using adaptive meshes and a fine uniform mesh.

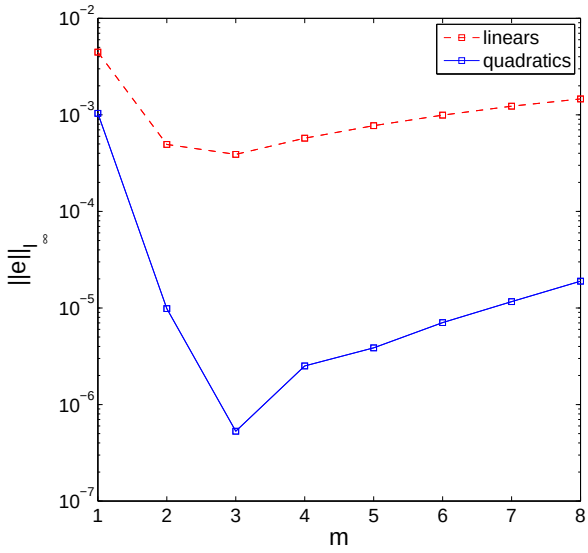

Figure 4.11: $\|\mathbf{e}^N\|_{l_\infty}$ error variation with respect to the parameter m in the BM monitor function.

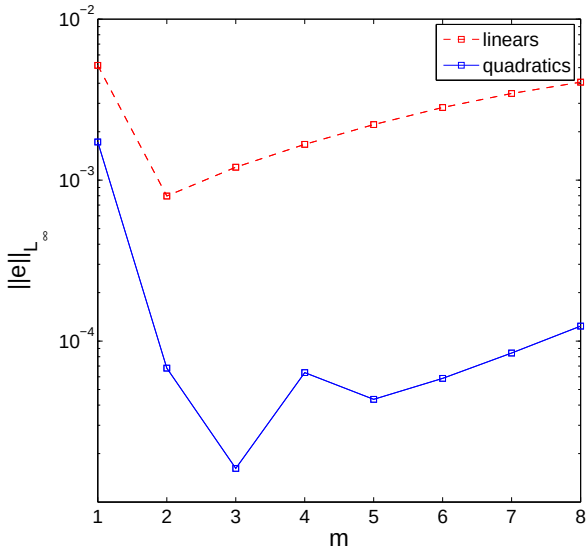

Figure 4.12: $\|e^N\|_{L_\infty}$ error variation with respect to the parameter m in the BM monitor function.

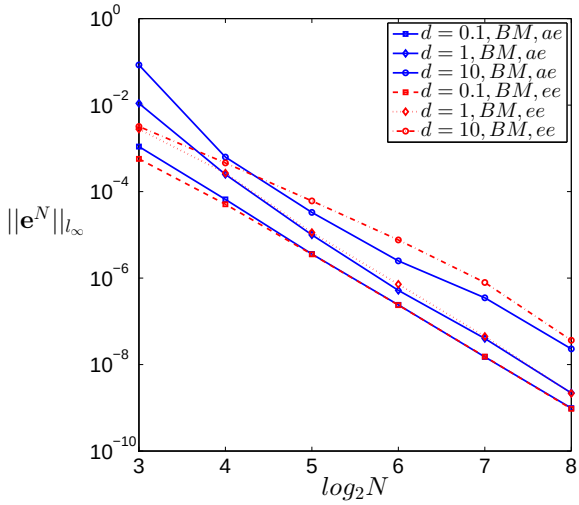

Figure 4.13: $\|\mathbf{e}^N\|_{l_\infty}$ obtained using approximate and exactly equidistributed adaptive meshes.

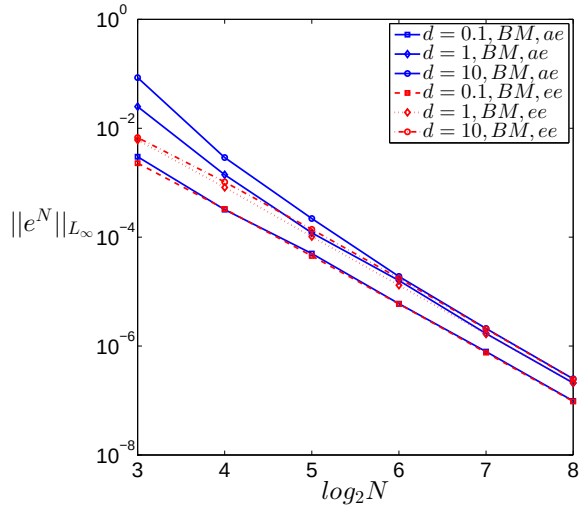

Figure 4.14: $\|e^N\|_{L_\infty}$ obtained using approximate and exactly equidistributed adaptive meshes.

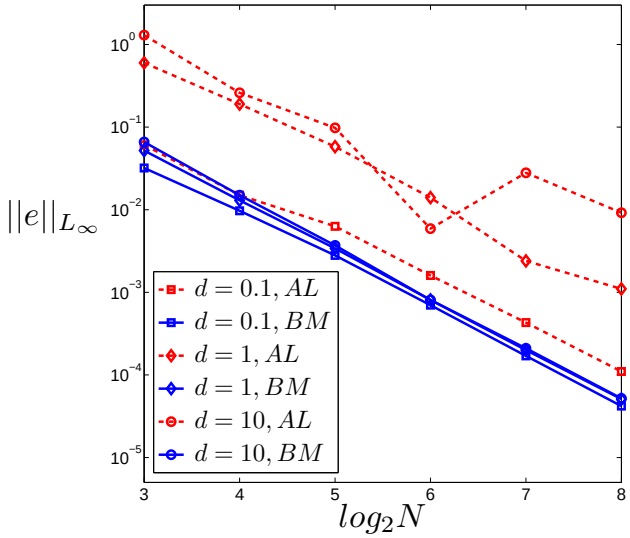

Figure 4.15: Comparison of accuracy obtained by equidistribution of the AL and BM monitor functions: convergence of $\|e^N\|_{L_\infty}$ using linear elements.

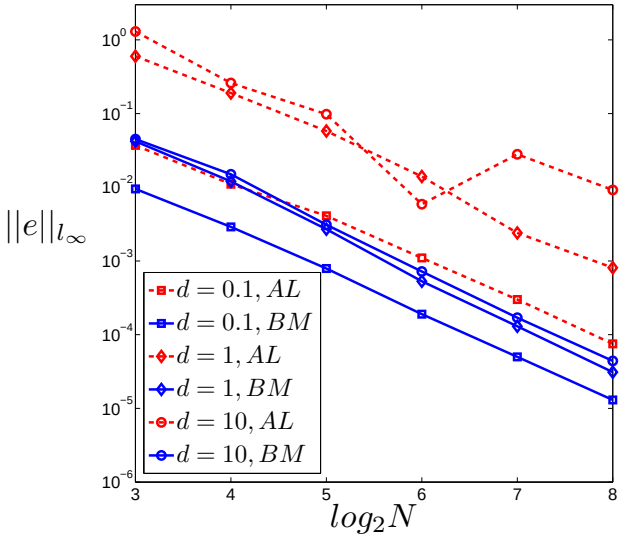

Figure 4.16: Comparison of accuracy obtained by equidistribution of the AL and BM monitor functions: convergence of $\|\mathbf{e}^N\|_{l_\infty}$ using linear elements.

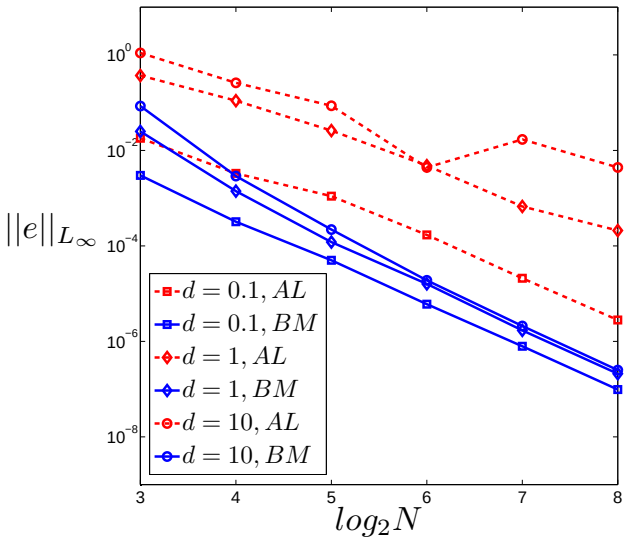

Figure 4.17: Comparison of accuracy obtained by equidistribution of the AL and BM monitor functions: convergence of $\|e^N\|_{L_\infty}$ using quadratic elements.

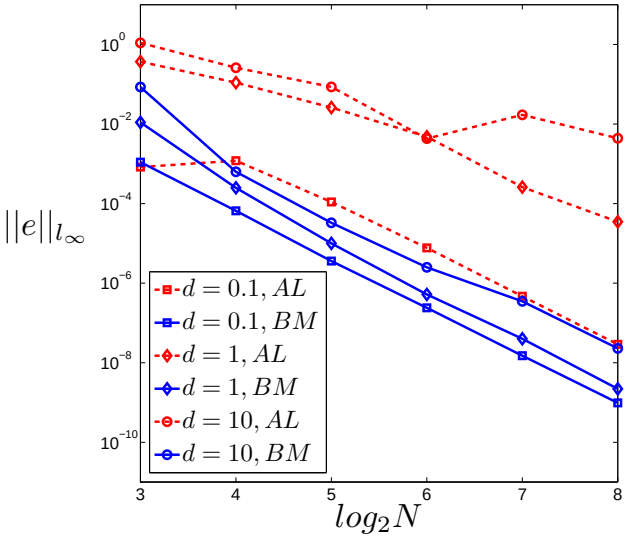

Figure 4.18: Comparison of accuracy obtained by equidistribution of the AL and BM monitor functions: convergence of $\|\mathbf{e}^N\|_{l_\infty}$ using quadratic elements.

	AL			BM			
$d = 0.1$							
TOL	N	it	$t(s)$	N	it	$t(s)$	% speedup
1e-3	32	1	0.091922	12	1	0.082384	12
1e-5	168	1	0.410808	56	1	0.093883	338
$d = 1$							
TOL	N	it	$t(s)$	N	it	$t(s)$	% speedup
1e-3	110	2	0.321125	16	2	0.113015	184
1e-5	816	1	39.4324	100	1	0.216643	18102
$d = 10$							
TOL	N	it	$t(s)$	N	it	$t(s)$	% speedup
1e-3	166	3	8.78359	26	10	0.708193	1140
1e-5	2042	1	4483.39	88	6	3.70893	120781

Table 4.3: Comparison of CPU times required to produce an error in the L_∞ norm below the value TOL using quadratic elements and the AL and BM monitor functions.

4.6 Conclusions

In this chapter we have considered the adaptive solution of a non-linear boundary value problem arising from a $\boldsymbol{Q}$-tensor model of a nematic liquid crystal. The solution-adaptive mesh is obtained by equidistribution of the BM monitor function, which has previously been used to solve linear reaction-diffusion problems. Numerical experiments have been carried out which show that the computed errors are robust to variations in the size of the liquid crystal cell size – a desirable property which is not realised when a uniform grid or an adaptive mesh based on equidistribution of the AL monitor function is used. An iterative algorithm has been used to find approximately equidistributed grids, and to obtain solutions to a given degree of accuracy. We have shown that the use of the BM monitor function can result in over a thousand fold decrease in CPU time compared to the use of the AL monitor function. The work in this chapter has been published in [48].

Although the results in this chapter deal with a somewhat idealised model problem, they do have important implications for the solution of more realistic physical problems such as the biaxial order reconstruction problem considered in [2, 3, 4, 1, 61, 62] and the following chapter. For this problem the full $\boldsymbol{Q}$-tensor must be used resulting in five coupled non-linear PDEs, corresponding to the components $q_i,\ i = 1, \ldots, 5$. These equations are solved in conjunction with an equation describing the electric potential. The improvements in efficiency presented here

using the BM monitor function are extremely promising and suggest that similar reductions in grid densities can be achieved for time-dependent problems in liquid crystal modelling, such as the π-cell problem tackled in the next chapter.

Chapter 5

π-cell problem

A common feature of the moving mesh studies listed in the conclusion of the previous chapter [2, 3, 4, 1, 61, 62] is the use of interpolation to transfer the numerical solution between meshes as it is evolved in time. While this procedure is possible in one dimension, it is not easily extended to higher dimensions. In addition, the moving mesh methods used previously have been based on a discretisation of the well known mesh equidistribution principle. More recently, however, it has been accepted that greater control (and hence robustness) can be obtained using a moving mesh partial differential equation (MMPDE) [41]: this is the approach we adopt in this chapter. Additional improvements on previously published studies include using a better adaptivity criterion together with a fully adaptive time-stepping procedure, leading to a more robust and accurate method overall. Most of the moving mesh papers above have studied the same test problem from Barberi

et al. [9], namely, using a one-dimensional model to investigate the dynamics of the biaxial switching of a nematic π-cell subjected to a strong electric field. A full numerical study of convergence properties of our adaptive algorithm is presented. As well as showing optimal convergence in time and space (with quadratic finite elements), we again observe nodal superconvergence. For the first time, a monitor function is constructed based upon a local measure of biaxiality. This is shown to lead to higher levels of solution accuracy and a considerable improvement in computational efficiency compared to those monitor functions previously applied to the π-cell problem.

5.1 Description of the π-cell problem

We consider a time-dependent switching process in a π-cell geometry which admits two topologically different equilibrium states. As mentioned previously, this problem (taken from [9]) has been used as a test example by previous authors interested in moving mesh methods [2, 3, 62]. Although it has only one space dimension, it embodies many of the features of our real target applications, and thus provides a satisfactory proof of concept for our approach.

The geometry is that of a π-cell [18], where the cell structure consists of a liquid crystal sample sandwiched between two horizontal plates a distance d apart. We consider a cell where $d = 1$ micron, and the liquid crystal parameters used

are taken from [9] (as described in §2.3). At both boundaries, the cell surface is treated so as to induce alignments uniformly tilted by a specified *tilt angle*, θ_T, but oppositely directed. This allows two topologically different equilibrium states: in one case, there is mostly horizontal alignment of the director with a slight splay and, in the other, there is mostly vertical alignment with a bend of almost π radians. Depending on the tilt angle and ratio of the elastic moduli of the liquid crystal material, either of these states might have a lower elastic energy, but the energy barrier between them is always large enough to prevent a spontaneous transition. In this chapter, an electric field is applied and the biaxial order reconstruction which occurs when the applied voltage is sufficiently large is modelled. Representative configurations of both states are illustrated in Figure 5.1.

As the focus of our study is the π-cell order reconstruction problem described in [9], we restrict our attention to one space dimension, with a single spatial co-ordinate z. In this case, it can be shown that equations (2.18) reduce to

$$\frac{\partial q_i}{\partial t} = \frac{\partial \Gamma_{iz}}{\partial z} - f_i \qquad i = 1, \ldots, 5, \tag{5.1a}$$

$$\frac{\partial D_z}{\partial z} = 0. \tag{5.1b}$$

We apply strong anchoring, i.e., Dirichlet conditions, to (5.1a). The boundary condition can be written as

$$\boldsymbol{Q} = \boldsymbol{Q}_s, \tag{5.2}$$

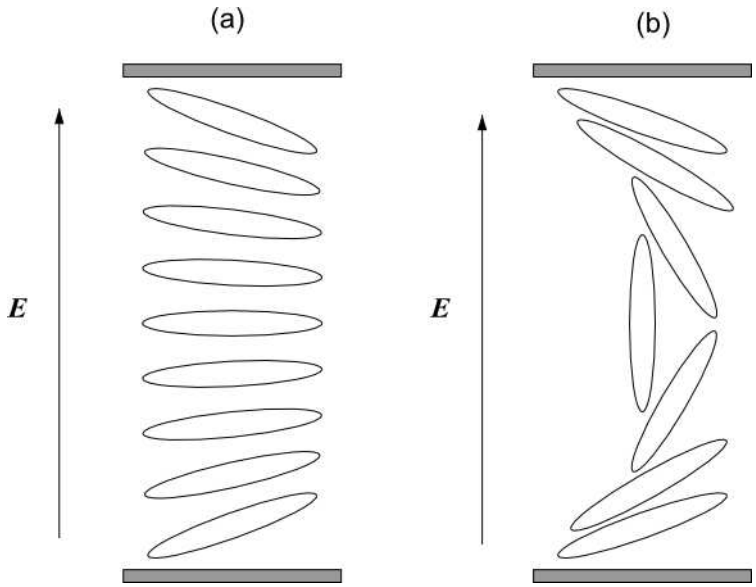

Figure 5.1: Cell configuration showing (a) horizontal and (b) vertical states under the influence of an electric field **E**.

where $\boldsymbol{Q}_s$ is the prescribed $\boldsymbol{Q}$-tensor at the boundary. For the problem considered in this chapter we have

$$\boldsymbol{Q}_s = \boldsymbol{Q}(\theta_{T_0}) \quad \text{at} \quad z = 0, \tag{5.3}$$

$$\boldsymbol{Q}_s = \boldsymbol{Q}(\theta_{T_d}) \quad \text{at} \quad z = d, \tag{5.4}$$

where θ_{T_0} and θ_{T_d} are the tilt angles at the bottom and top cell boundaries respectively. We assume that the director is initially aligned in the horizontal state shown in Figure 5.1, and apply an initial condition where θ_T varies linearly between θ_{T_0} at $z = 0$ and θ_{T_d} at $z = d$. We solve (5.1b) subject to boundary conditions $U = 0$ at $z = 0$ and $U = V_{\max}$ at $z = d$, where $V_{\max}$ is the maximum strength of the applied electric field. Initially U is assumed to be zero throughout the cell.

For computational purposes, we non-dimensionalise the equations in (5.1), scaling length with respect to the nematic coherence length $\zeta = \sqrt{9CL_2/2B^2}$ (as was done in Chapter 4), and the energies are scaled by the quantity $A(T - T^*)$.

5.2 Moving the mesh

We have already discussed how the governing physical PDEs need to be reformulated to account for the movement of the mesh: we presented a conservative weak reformulation of the $\boldsymbol{Q}$-tensor equations in §3.1, and the subsequent finite element semi-discretisation of these equations in §3.2. We now discuss the procedure used to generate the adaptive mesh at each time step. Mesh movement is generally performed by solving a moving mesh partial differential equation (MMPDE) derived from the *equidistribution principle*. Recall from §4.4.1 that the equidistribution principle makes use of a mesh density function (referred to as the *monitor function*) which is some measure of local solution difficulty, and which is supposed to be distributed equally throughout the domain. It is common to formulate an MMPDE based on a variational formulation of the equidistribution principle. Using the variational approach, we obtain an MMPDE derived from the Euler-Lagrange equations associated with a functional that has been chosen specifically for the purposes of generating an adpative mesh. The variational approach falls under the umbrella of *location-based* strategies as described by Huang & Russell in [41].

This approach controls directly the location of the mesh points, as opposed to a *velocity-based* approach that generates a mesh by integrating the velocity field. We do not consider any velocity-based method, and instead direct the interested reader to Chapter 7 in [41] for further reading on these methods.

5.2.1 Moving mesh partial differential equation

Recall that in §4.4.1 we presented the one-dimensional equidistribution principle as a means of updating our adaptive mesh. Alternatively, a differential equation for $z(\xi, t)$ can be obtained by differentiating (4.18) with respect to ξ to give

$$\rho(\mathcal{T}(z,t))\frac{\partial z}{\partial \xi} = \theta(t), \tag{5.5}$$

where

$$\theta(t) = \int_a^b \rho(\mathcal{T}(z,t)\,\mathrm{d}z. \tag{5.6}$$

A discretisation of (5.5) was used in the moving mesh method of Ammodeo et al. [2, 3]. However, one major drawback of using (4.18) or (5.5) is the lack of control of the grid trajectories. This can lead to instabilities in the resulting algorithms that can only be avoided by the use of excessively small time steps, which is clearly undesirable. Differentiating (4.18) with respect to ξ twice we obtain a *quasi-static equidistribution principle*

$$\frac{\partial}{\partial \xi}\left(\rho(\mathcal{T}(z,t))\frac{\partial z}{\partial \xi}\right) = 0. \tag{5.7}$$

If the roles of the dependent and independent variables are swapped then, if (5.7) holds, in terms of the inverse mapping $\xi(x,t)$ we have

$$\frac{\partial}{\partial z}\left(\frac{1}{\rho(\mathcal{T}(z,t))}\frac{\partial \xi}{\partial z}\right) = 0. \tag{5.8}$$

The equidistribution equation (5.5) can alternatively be interpreted as the Euler-Lagrange equation associated with the functional

$$I[\xi(z,t)] = \frac{1}{2}\int_0^1 \frac{1}{\rho(\mathcal{T}(z,t))}\left(\frac{\partial \xi}{\partial z}\right)^2 \mathrm{d}z. \tag{5.9}$$

An MMPDE can then be defined in terms of the gradient flow equation,

$$\frac{\partial \xi}{\partial t} = -\frac{\delta I}{\delta \xi}. \tag{5.10}$$

The motivation for considering the gradient flow equation comes from a stability result presented in [41]. The first order functional derivative has opposite direction to the direction of fastest descent of (5.9), however in practice the fastest direction of descent is not necessarily the best choice for our adaptive strategy. We therefore introduce a positive constant temporal smoothing parameter τ, and a spatial balancing operator $P(z,t)$, and define our MMPDE in terms of the modified gradient flow equation

$$\frac{\partial \xi}{\partial t} = -\frac{P}{\tau}\frac{\delta I}{\delta \xi} = \frac{P}{\tau}\frac{\partial}{\partial z}\left(\frac{1}{\rho(\mathcal{T}(z,t))}\frac{\partial \xi}{\partial z}\right). \tag{5.11}$$

In practice it is more convenient to swap the roles of the dependent and independent variables, as $\xi = \xi(z,t)$ does not explicitly specify the mesh on the physical

domain. Doing so, we obtain

$$\frac{\partial z}{\partial t} = \frac{P}{\tau}\left(\rho(\mathcal{T}(z,t))\frac{\partial z}{\partial \xi}\right)^{-2}\frac{\partial}{\partial \xi}\left(\rho(\mathcal{T}(z,t))\frac{\partial z}{\partial \xi}\right), \quad z(0,t) = a, \quad z(1,t) = b. \tag{5.12}$$

In principle we could solve (5.8) to determine the coordinate transformation $z = z(\xi, t)$. However, there are multiple benefits from instead determining the coordinate transformation via a method that explicity includes the mesh velocity. Firstly, we note that the semi-discretisation of (5.8), subject to identical boundary conditions as in (5.12), gives rise to a system of algebraic equations, which results in a system of differential algebraic equations having to be solved. A semi-discretisation of (5.12) results instead in a system of ordinary differential equations, which is often easier to integrate. Furthermore, introducing mesh velocity into the MMPDE has the effect of introducing a degree of temporal smoothing, making for a more robust adaptive strategy.

We note that (5.12) contains the left-hand-side of the one dimensional equidistribution principle as presented in (5.7), i.e.,

$$\frac{\partial}{\partial \xi}\left(\rho(\mathcal{T}(z,t))\frac{\partial z}{\partial \xi}\right). \tag{5.13}$$

When the equidistribution principle is satisfied exactly this term disappears and we have no mesh movement. It can thus be interpreted that this term plays a significant role in influencing how the mesh adapts, and serves to force the mesh to adapt in such a way as to equidistribute the monitor function $\rho(\mathcal{T}(z,t))$.

5.2.2 Spatial balancing

The spatial balancing coefficient $P(z,t)$ in (5.11) was introduced with the motivation that by choosing a suitable $P(z,t)$, the entire mesh adapts on a uniform time scale, thereby reducing the stiffness of the MMPDE and making it more robust for a constant choice of temporal smoothing parameter τ. However, we find from experimental experience that for the π-cell problem considered in this chapter, a sophisticated choice of $P(z,t)$ is unnecessary, and we simply set $P(z,t) = 1$. Note that in two dimensions we find that more consideration must be given to the choice of the spatial balancing coefficient, and that it has significant impact on the stiffness of the MMPDE; we explore this in further detail in the next chapter.

5.2.3 Monitor functions

As seen in the previous chapter, essential to the success of any moving mesh method is the choice of an appropriate monitor function. Previous studies have used the scaled solution arc-length (AL) monitor function (cf. (4.20))

$$\rho(\mathcal{T}(z,t)) = \sqrt{\mu + \left(\frac{\partial \mathcal{T}(z,t)}{\partial z}\right)^2}, \tag{5.14}$$

[61], and the Beckett-Mackenzie (BM1) monitor function (cf. (4.21))

$$\rho(\mathcal{T}(z,t)) = \alpha(z,t) + \left|\frac{\partial \mathcal{T}(z,t)}{\partial z}\right|^{\frac{1}{2}}, \qquad \alpha(z,t) = \int_0^1 \left|\frac{\partial \mathcal{T}(z,t)}{\partial z}\right|^{\frac{1}{2}} \mathrm{d}z, \tag{5.15}$$

[2, 3], where μ and $\alpha(z,t)$ are scaling parameters that were discussed in depth in the previous chapter. Here we also consider the alternative monitor function based

on second derivatives of $\mathcal{T}(z,t)$, that is

$$\rho(\mathcal{T}(z,t)) = \alpha(z,t) + \left|\frac{\partial^2 \mathcal{T}(z,t)}{\partial z^2}\right|^{\frac{1}{m}}, \quad \alpha(z,t) = \max\left\{\tilde{\alpha}, \int_0^1 \left|\frac{\partial^2 \mathcal{T}(z,t)}{\partial z^2}\right|^{\frac{1}{m}} \mathrm{d}z\right\}, \tag{5.16}$$

[12] which we will refer to as the BM2 monitor function.

Recall from §4.3.1 that appropriate choices for μ and $\alpha(z,t)$ are vital to the success of our adaptive strategy. Heuristically we can say a small intensity parameter is likely to locate too large a proportion of mesh points in areas of high solution difficulty, and as the intensity parameter increases the mesh is distributed more evenly across the domain. It is common to simply set $\mu = 1$ in (5.14), and the the resulting monitor function is exactly that considered in [61]: for comparison we shall also consider this monitor function. Ideally, however, we wish to have control over the size of the intensity parameter throughout the temporal integration of our problem. A solution-dependent intensity parameter has been considered by various authors, and following the work in [11], we have introduced $\alpha(z,t)$ in (5.15), where

$$\alpha(z,t) = \int_0^1 \left|\frac{\partial \mathcal{T}(z,t)}{\partial z}\right|^{\frac{1}{2}} \mathrm{d}z. \tag{5.17}$$

The above definition of the intensity parameter tends to result in a more evenly distributed mesh than simply using a user defined constant. By being solution dependent, it has the added benefit of not always needing to be tailored to a specific problem. The principle disadvantage of this definition of $\alpha(z,t)$ is that it

can be very sensitive to small changes in $\mathcal{T}(z,t)$. More precisely, when $\alpha(z,t) \ll 1$, any error introduced in calculating the monitor function will be amplified, thus causing poor quality meshes to be generated. We can alleviate this problem by bounding $\alpha(z,t)$ away from zero. We therefore consider instead in (5.16)

$$\alpha(z,t) = \max\left\{\tilde{\alpha}, \int_0^1 \left|\frac{\partial^2 \mathcal{T}(z,t)}{\partial z^2}\right|^{\frac{1}{m}} \mathrm{d}z\right\}. \tag{5.18}$$

By introducing a lower bound on $\alpha(z,t)$ we restrict the adaptation of the mesh so that points will only relocate to areas where the derivatives of the monitor function are significantly greater than $\tilde{\alpha}$. We now encounter a problem similar to that which we faced at the beginning of this section, i.e., how best to choose $\tilde{\alpha}$? Experimentally, we find that simply choosing $\tilde{\alpha} = 1$ is sufficient. We note that the overall definition of $\alpha(z,t)$ is still a significant improvement over simply using $\mu = 1$.

In practice, the computed monitor function is often very non-smooth, and so we smooth the monitor function throughout our adaptive strategy. This has the effect of smoothing the mesh, and also making it easier, and thus less costly, to integrate the MMPDE. In addition to the smoothing of the MMPDE described in (5.3.1), recall that the parameter m in (5.16) has the effect of smoothing potentially large variations in the value of the $\mathcal{T}(z,t)$. Increasing m will lead to fewer mesh elements in the boundary layer region leaving more to be deployed to cover the rest of the domain.

Having identified monitor functions, it remains to decide on an appropriate input function $\mathcal{T}(z,t)$. In previous studies [2, 3, 61, 62], the authors set $\mathcal{T}(z,t) = \mathrm{tr}(\boldsymbol{Q}^2)$ which is known to vary rapidly in regions where order reconstruction occurs. Also, we have shown in Chapter 4 that, for the uniaxial boundary value problem considered there, the ideal quantity on which to base the monitor function is the scalar order parameter S (recall that, for a uniaxial state, $\mathrm{tr}(\boldsymbol{Q}^2) = S^2$). However, it is not immediately apparent that $\mathrm{tr}(\boldsymbol{Q}^2)$ is the ideal quantity on which to base a monitor function for problems involving biaxiality. In this chapter, we therefore compare results computed using $\mathcal{T}(z,t) = \mathrm{tr}(\boldsymbol{Q}^2)$ with those computed using a direct measure of biaxiality. That is, we also use $\mathcal{T}(z,t) = b(z,t)$, where

$$b(z,t) = \left[1 - \frac{6\,\mathrm{tr}(\boldsymbol{Q}^3)^2}{\mathrm{tr}(\boldsymbol{Q}^2)^3}\right]^{\frac{1}{2}} \tag{5.19}$$

is an invariant measure of biaxiality [9]. The range of this measure is $b \in [0,1]$, with uniaxial states corresponding to $b = 0$ and totally biaxial states corresponding to $b = 1$. In the experiments described in §5.4, we use $\mathcal{T}(z,t) = \mathrm{tr}(\boldsymbol{Q}^2)$ with the AL and BM1 monitor functions (as studied in [2, 3, 61, 62]), and distinguish between our two variants of BM2 by using BM2a for (5.16) with $\mathcal{T}(z,t) = \mathrm{tr}(\boldsymbol{Q}^2)$, and BM2b for (5.16) with $\mathcal{T}(z,t) = b$.

5.3 Iterative solution algorithm

We now describe a decoupled iterative procedure to update the mesh and the

solution of the physical PDEs. This strategy is similar to that originally proposed in [14] and [15]. One of the major advantages of decoupling the solution procedures is that it allows the flexibility of using different convergence criteria for the mesh and the physical solution. This is important, as it is well appreciated that the computational mesh is rarely required to be resolved to the same degree of accuracy as the physical solution.

We integrate forward in time in an iterative manner, solving for the grid and the physical solution alternatingly. The following algorithm is used, where MAXPASS is the total number of passes allowed to reach a degree of convergence between successive estimates of the grid at the forward time level.

Set an initial uniform mesh Δ_N^0. Set the initial guess $\boldsymbol{q}_i^0$ and $\boldsymbol{u}^0$.

Select an initial Δt^0. Set $n = 0$.

while $(t^n < t^{\max})$;

pass $= 0$.

$\Delta_N^{\text{pass}} = \Delta_N^n$, $\boldsymbol{q}_i^{\text{pass}} = \boldsymbol{q}_i^n$, $\boldsymbol{u}^{\text{pass}} = \boldsymbol{u}^n$.

while (pass $<$ MAXPASS);

$O\Delta_N = \Delta_N^{\text{pass}}$.

Evaluate monitor function using Δ_N^{pass} and $\boldsymbol{q}_i^{\text{pass}}$.

Integrate (5.12) forward in time to obtain new grid $\Delta_N^{\text{pass}+1}$.

Integrate (2.18) forward in time to obtain $\boldsymbol{q}_i^{\text{pass}+1}$, $\boldsymbol{u}^{\text{pass}+1}$.

if $(||\Delta_N^{\text{pass}+1} - O\Delta_N||_{l_\infty} < \text{M}_{\text{tol}})$, **break**;

pass $=$ pass $+ 1$.

end while

$\Delta_N^{n+1} = \Delta_N^{\text{pass}}$, $\boldsymbol{q}_i^{n+1} = \boldsymbol{q}_i^{\text{pass}}$, $\boldsymbol{u}^{n+1} = \boldsymbol{u}^{\text{pass}}$.

$n := n + 1$.

end while

For efficiency, it is strongly desirable to only perform a small number of iterations at each time-step so we use MAXPASS $= 4$. If the grid converges quickly, the loop will be stopped before four passes have been completed. Note that one

of the major advantages of using an MMPDE fully coupled with the PDEs from the physical problem (as opposed to the static regridding type methods used in [2, 3, 23]) is that there is no need for any interpolation as the solution of the physical PDEs is approximated on the mesh at the forward time level.

5.3.1 Discretisation of the MMPDE

Although the MMPDE could also be discretised using a finite element method, here we use a finite difference approximation (primarily for the convenience of adapting existing code). Specifically, we discretise (5.12) using second-order central differences on the uniform mesh $\xi_i = i/N$, $i = 0, 1, \ldots, N$ and obtain the semi-discrete system of moving mesh equations

$$\dot{z}_i = \frac{4}{\tau}\left(\tilde{\rho}_i(z_{i+1} - z_{i-1})\right))^{-2}\left(\tilde{\rho}_{i+\frac{1}{2}}(z_{i+1} - z_i) - \tilde{\rho}_{i-\frac{1}{2}}(z_i - z_{i-1})\right). \tag{5.20}$$

In (5.20), $\tilde{\rho}_{i+\frac{1}{2}}$ is a smoothed monitor function defined as in [54] by

$$\tilde{\rho}_{i+\frac{1}{2}} = \frac{\sum_{k=i-p}^{i+p} \rho_{k+\frac{1}{2}} \left(\frac{q}{q+1}\right)^{|k-i|}}{\sum_{k=i-p}^{i+p} \left(\frac{q}{q+1}\right)^{|k-i|}},$$

where q is a positive real constant and p is a non-negative integer. Based on our experience in practice, we set $q = 2$ and $p = 3$. The term $\tilde{\rho}_i$ is given by

$$\tilde{\rho}_i = \frac{\tilde{\rho}_{i-\frac{1}{2}}(z_{i+\frac{1}{2}} - z_i) + \tilde{\rho}_{i+\frac{1}{2}}(z_i - z_{i+\frac{1}{2}})}{z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}}},$$

where $z_{i+\frac{1}{2}} = \frac{1}{2}(z_{i+1} + z_i)$. The mesh at time level $t = t^{n+1}$ is computed using an implicit Euler approximation to (5.20).

5.3.2 Time integration

To integrate the physical equations (3.12) forward in time, we employ a second-order singly diagonally implicit Runge-Kutta (SDIRK2) method similar to that used in [14]. This Runge-Kutta method is represented by the Butcher array

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array} = \begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array}$$

where $\gamma = (2 - \sqrt{2})/2$.

Integration of each equation in (3.13a) from $t = t^n$ to $t = t^{n+1}$ takes place via intermediate stages $\boldsymbol{K}_{i,1}$ and $\boldsymbol{K}_{i,2}$ with

$$\begin{aligned} M^{n+\gamma}\boldsymbol{K}_{i,1} &= \Delta t \boldsymbol{G}_i(t + c_1\Delta t, (M^{n+\gamma})^{-1}M^n\boldsymbol{q}_i^n + a_{11}\boldsymbol{K}_{i,1}, \boldsymbol{u}^{n+\gamma}) \\ M^{n+1}\boldsymbol{K}_{i,2} &= \Delta t \boldsymbol{G}_i(t + c_2\Delta t, (M^{n+1})^{-1}M^n\boldsymbol{q}_i^n \\ &\quad + a_{21}\boldsymbol{K}_{i,1} + a_{22}\boldsymbol{K}_{i,2}, \boldsymbol{u}^{n+1}) \\ M^{n+1}\boldsymbol{q}_i^{n+1} &= M^n\boldsymbol{q}_i^n + b_1\boldsymbol{K}_{i,1} + b_2\boldsymbol{K}_{i,2}, \end{aligned} \tag{5.21}$$

where $\boldsymbol{q}_i^n$ denotes the value of $\boldsymbol{q}_i$ at time level n and M^n is the finite element mass matrix at time level n. The intermediate stages $\boldsymbol{K}_{i,1}$ and $\boldsymbol{K}_{i,2}$ are found using Newton iteration. That is, for $r = 1, 2$ we solve

$$\left[M^{n+c_r} - a_{rr}\Delta t \left(\frac{\partial G}{\partial \boldsymbol{q}_i}\right)^p_{\boldsymbol{q}^p_{i,r}}\right]\left(\boldsymbol{K}^{p+1}_{i,r} - \boldsymbol{K}^p_{i,r}\right) = \Delta t \boldsymbol{G}(t + c_r\Delta t, \boldsymbol{q}^p_{i,r}) - M^{n+c_r}\boldsymbol{K}^p_{i,r},$$

where $\boldsymbol{q}_{i,r}^{p}$ denotes the estimate of $\boldsymbol{q}_{i,r}$ at the p^{th} step of the Newton iteration, with

$$\boldsymbol{q}_{i,r}^{p} \equiv (M^{n+c_r})^{-1} M^{n} \boldsymbol{q}_{i}^{n} + a_{r1} \boldsymbol{K}_{i,1}^{p} + a_{r2} \boldsymbol{K}_{i,2}^{p}.$$

At the p^{th} step of the Newton iteration we solve (3.13b) for $\boldsymbol{u}^{p}$ to update $\boldsymbol{u}$ in (5.21). We note in passing that if we choose not to update $\boldsymbol{u}$ after each iteration of the Newton method, we find that the temporal convergence rates presented in §5.4.1 are first order as opposed to second order. Newton's method is terminated when $\|\boldsymbol{K}_{i,r}^{p+1} - \boldsymbol{K}_{i,r}^{p}\|_{\infty} \leq \mathrm{K}_{\text{tol}}$. In the numerical experiments in §5.4 we set $\mathrm{K}_{\text{tol}} = 10^{-7}$.

5.3.3 Adaptive time-step control

It has been shown in [61] that the π-cell problem is well suited for spatial adaptivity. However, it appears that this problem is also well suited for temporal adaptivity, as the events of most interest, namely the switching on and off of the electric field and the biaxial switching, occur over time-scales several orders of magnitude smaller than the total simulation time. We therefore expect that efficiency gains can be made by implementing an adaptive time-stepping algorithm that makes use of the fact that temporal gradients throughout large periods of our simulations are relatively small compared to the time interval over which biaxial switching takes place.

In the course of integrating the solution forward in time, we employ adaptive time-stepping based on the computed solutions for $\boldsymbol{q}_i$ and on the solution of the

MMPDE. To measure the solution error for $\boldsymbol{q}_i$ we use the embedded first-order SDIRK approximation

$$\hat{\boldsymbol{q}}_i^{n+1} = \boldsymbol{q}_i^n + \Delta t_n \boldsymbol{K}_{i,1},$$

which we obtain at no extra computational cost from the SDIRK2 scheme outlined in §5.3.2. The error indicator used is then

$$\mathrm{E}_i = \left(\sum_{j=0}^{N-1} (z_{j+1}^{n+1} - z_j^{n+1}) \left(\frac{e_{i,j}^{n+1} + e_{i,j+1}^{n+1}}{2} \right)^2 \right)^{\frac{1}{2}},$$

where

$$e_{i,j}^{n+1} = q_{i,j}^{n+1} - \hat{q}_{i,j}^{n+1}.$$

Here z_j^n denotes the j^{th} node of the mesh at time level n, and $q_{i,j}^{n+1}$ denotes the value of the j^{th} entry of $\boldsymbol{q}_i$ at node j at time level $n+1$. The time-step is then adapted via the formula

$$\Delta t_{\text{sol}}^{n+1} = \Delta t^n \times \min \left(\text{maxfac}, \max \left[\text{minfac}, \eta \left(\frac{\mathrm{E}_{\text{tol}}}{\max_i(\mathrm{E}_i)} \right)^{\frac{1}{2}} \right] \right).$$

In the computations in §5.4, we set the tolerance $\mathrm{E}_{\text{tol}} = 5 \times 10^{-5}$, maxfac = 6.0, minfac = 0.1 and $\eta = 0.9$. With these parameters, the mesh based control is similar to that of the computed solution. We measure the accuracy of a particular mesh using the quantity

$$\mathrm{g}_{\text{err}} = \max_{j=0,\ldots,N} |z_j^{n+1} - z_j^n|.$$

The predicted time-step based on the mesh error is then given by

$$\Delta t_{\text{mesh}}^{n+1} = \Delta t^n \times \min\left(\text{maxfac}, \max\left[\text{minfac}, \frac{\log(\text{g}_{\text{err}})}{\log(\text{g}_{\text{bal}})}\right]\right),$$

where we choose $\text{g}_{\text{bal}} = 0.8 \times \text{M}_{\text{tol}}$ and $\text{M}_{\text{tol}} = 5 \times 10^{-2}$. The time-step at the forward time level is then given by

$$\Delta t^{n+1} = \min\left(\Delta t_{\text{sol}}^{n+1}, \Delta t_{\text{mesh}}^{n+1}\right).$$

A similar algorithm has previously been used successfully for solving the one-dimensional viscous Burgers' equation [14].

5.4 Numerical results

The numerical tests in this section were carried out for the π-cell problem, with an applied electric field of sufficient strength to induce switching. We re-emphasise here that although this is a one-dimensional problem, it is still very challenging numerically and contains several features typical of problems in this area. In all of our experiments, strong anchoring is applied at the upper and lower cell boundaries; specifically, we assume that the cell surface has been treated so as to induce pre-tilt angles of $\theta = \pm 20^\circ$. At time $t = 0$ ms, the director angle varies linearly between these two angles, as in the horizontal state in Figure 5.1. Initially, the biaxiality is negligible in the bulk, with two small-amplitude ($b \approx 4 \times 10^{-2}$) boundary layers forming due to the boundary conditions. An electric field of

strength 11.35 V μm^{-1} is applied parallel to the z-axis at time $t = t_{\text{on}} = 0.005$ ms. The director then begins to align vertically, parallel to the electric field, but is initially prevented from doing so, at the cell centre by the energy barrier and at the boundaries by the strong anchoring. Once the field is switched on, a thin layer forms in the biaxiality at the cell centre which, as time evolves, steadily increases in size until switching takes place at $t = t_{\text{switch}} \approx 0.112$ ms. Figure 5.2 shows a blow-up of the complicated structure seen close to the switching time: the biaxiality at the cell centre has a volcano-like structure with a rim where $b = 1$ representing the purely biaxial state and a planar uniaxial point ($b = 0$) at the cell centre at the exact switching time.

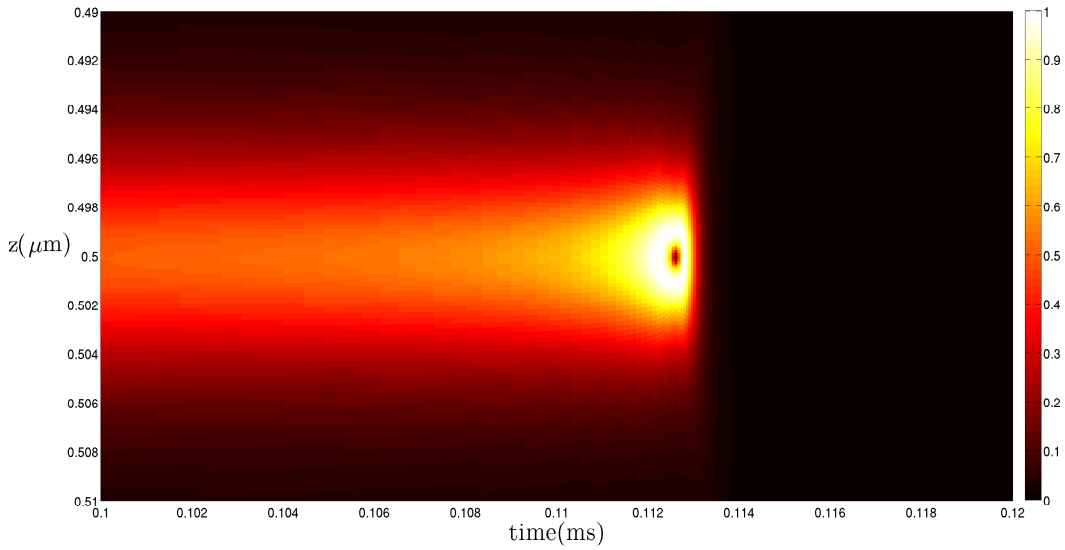

Figure 5.2: Surface plot of biaxiality at the cell centre with electric field strength 11.35 $V\ \mu m^{-1}$ using the BM2b monitor function with 256 quadratic elements.

After the transition, the size of the biaxial wall at the cell centre rapidly decreases until b is again close to zero in the bulk and only the two boundary layers remain (again with $b \approx 4 \times 10^{-2}$). Finally, the field is switched off at time $t = t_{\text{off}} = 0.15$ ms, after which the biaxiality at the cell boundaries decreases further to an almost negligible level.

5.4.1 Convergence rates in space and time

In Chapter 4 we presented convergence results for a scalar model of a one-dimensional uniaxial problem. In a similar vein, we now investigate the convergence rates of both spatial and temporal errors for the much more complicated physical problem studied here. This would clearly be very difficult to do at the exact moment of switching, but we can still obtain valid results by choosing a pre-switching time (for studying temporal convergence) and a post-switching time (for studying spatial convergence).

As an analytical solution to this problem is not available, we compare our computed solutions with a reference solution obtained on an adaptive mesh obtained using the BM2b monitor function with 2048 quadratic elements and a uniform time step $\Delta t = 10^{-9}$ seconds. We will use $q_{i*}(z,t)$ to denote this reference approximation to $q_i(z,t)$, and assume throughout that

$$|q_{i*}(z,t) - q_i(z,t)| \ll |q_{i*}(z,t) - q_{iN}(z,t)|\,,$$

where q_{iN} denotes the finite element approximation calculated on a grid with N quadratic elements. Note that the results presented are independent of this specific choice of reference function: calculations using reference functions based on the other three monitor functions give very similar results. The error in the approximation q_{iN} will be denoted by $e^N_{q_i}$.

To estimate the L_∞ norm of the error, we subdivide each element using the 11 error sampling points given by

$$z_{jk} = z_{j-1} + \frac{k-1}{10}(z_j - z_{j-1}), \qquad j = 1, \ldots, N, \quad k = 1, \ldots, 11,$$

and estimate the error at $t = t^*$ to be

$$\|e^N_{q_i}\|_{L_\infty} = \max_{j=1,\ldots,N} \left(\max_{1 \le k \le 11} |q_{i*}(z_{jk}, t^*) - q_{iN}(z_{jk}, t^*)| \right). \tag{5.22}$$

Since the sampling points z_{jk} will not in general coincide with the reference grid points, the solution $q_{i*}(z_{jk}, t^*)$ is interpolated using the quadratic shape functions and the local solution defined on the reference grid that includes the point z_{jk}. We also estimate the spatial error in the l_∞ norm using the maximum error computed at the grid nodes, that is,

$$\|\boldsymbol{e}^N_{q_i}\|_{l_\infty} = \max_{j=0,\ldots,N} |q_{i*}(z_j, t^*) - q_{iN}(z_j, t^*)|. \tag{5.23}$$

We first consider convergence of the time discretisation scheme (for a fixed time-step Δt). We estimate the error at time $t = t^* = 0.1024$ ms, that is, before switching has occurred. The values of (5.22) and (5.23) for q_1, q_3, q_4 and U are

presented in Figure 5.3 for various values of Δt (components q_2 and q_5 are exactly zero for this problem).

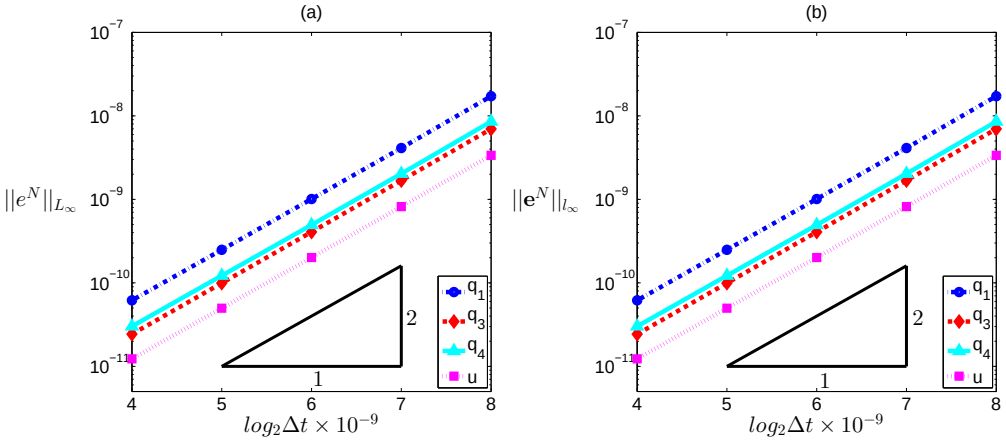

Figure 5.3: Temporal error convergence of non-zero components of $\boldsymbol{Q}$ at the cell centre at time 0.1024 ms with electric field strength 11.35 $V\ \mu m^{-1}$ using the BM2b monitor function with 256 quadratic elements.

We see that, in both norms, the errors converge at a rate which is $\mathcal{O}(\Delta t^{-2})$ as we would expect when using a second order method to integrate forward in time. It is important to note that this optimal rate of convergence is only achieved if equation (3.13b) is solved for the electric potential at every Newton iteration. If, to increase efficiency, $\boldsymbol{u}$ is updated only after obtaining $\boldsymbol{q}_i^{n+1}$ (that is, once per time-step) we only achieve first order convergence in time.

We now turn to estimating the rate of spatial convergence. We examine the

error at time $t^* = 2$ ms as by this time the solution has entered a steady state with only boundary layers present in the biaxiality. The error norms (5.22) and (5.23) for the non-zero components of and $\boldsymbol{Q}$ and U are presented in Figure 5.4 for various values of N.

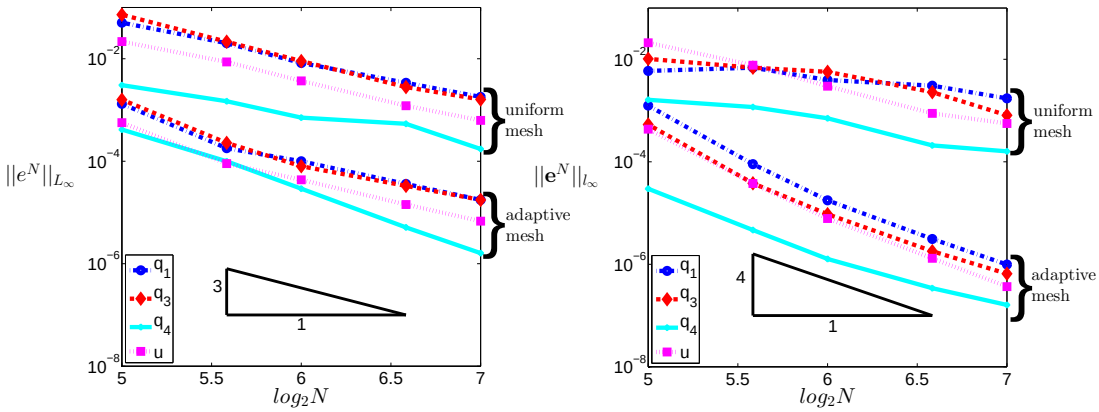

Figure 5.4: Spatial error convergence of non-zero components of $\boldsymbol{Q}$ at the cell centre at time 1 ms with electric field strength 11.35 $V\ \mu m^{-1}$ using the BM2b monitor function.

We observe that $\|e^N_{q_i}\|_{L_\infty}$ appears to converge at the rate $\mathcal{O}(N^{-3})$, which is the optimal rate expected using quadratic elements. However, the convergence rate of the error at the mesh nodes, that is, $\|\boldsymbol{e}^N_{q_i}\|_{l_\infty}$, appears to be $\mathcal{O}(N^{-4})$. A similar convergence rate was observed in Chapter 4 for a one-dimensional uniaxial problem. Although it is well known that the finite element method can exhibit

nodal supercovergence, it is remarkable that this property still holds when solving a system of highly non-linear PDEs.

5.4.2 Modelling the order reconstruction

In this section, we look in detail at the accuracy and efficiency of the various monitor functions in capturing the transient features of the order reconstruction process. Specifically, we will compare the four different monitor functions described in §5.2.3, namely, AL ((5.14) with $\mathcal{T}$ =tr($\boldsymbol{Q}^2$)), BM1 ((5.15) with $\mathcal{T}$ =tr($\boldsymbol{Q}^2$)), BM2a ((5.16) with $\mathcal{T}$ =tr($\boldsymbol{Q}^2$)) and BM2b ((5.16) with $\mathcal{T} = b$).

5.4.2.1 Mesh trajectories

The trajectories for the adaptive grids obtained using the four monitor functions above are shown in Figure 5.5.

The red vertical lines indicate the times at which the electric field is switched on, when switching occurs, and when the field is switched off. In each case 256 elements have been used, although only every eighth node is plotted for clarity. We observe that, shortly after the electric field is switched on at $t = t_{\text{on}}$, all four meshes adapt to resolve the large solution gradients at both the cell centre and the cell boundaries. However, although the mesh generated using AL adapts well, it does so sharply; we also observe that mesh points move continuously in regions far from the cell centre and cell walls when $t_{\text{on}} < t < t_{\text{switch}}$, even though solution gradients

are small in these areas. These are undesirable properties as it is well known that smooth meshes are likely to allow larger time-steps to be taken during the course of time integration. The continually large variations in the meshes generated using AL explain the inefficiencies that are discussed in §5.4.2.2. In contrast, it can be seen that the meshes obtained using BM1, BM2a and BM2b evolve more smoothly. Switching occurs at roughly the same time ($t = t_{\text{switch}}$) for each grid: the exact switching times are tabulated in Table 5.2 in §5.4.2.4. After switching, all of the meshes relax gradually (in the cases of BM1, BM2a and BM2b) or sharply (AL) at the cell centre due to the disappearance of the large solution gradient there. After the order reconstruction, but while the electric field is still switched on, the meshes are only adapted at the boundaries where layers remain due to competition between the electric field and the strong anchoring boundary condition. After the electric field is switched off ($t = t_{\text{off}}$), we can see that all meshes relax further at the boundaries.

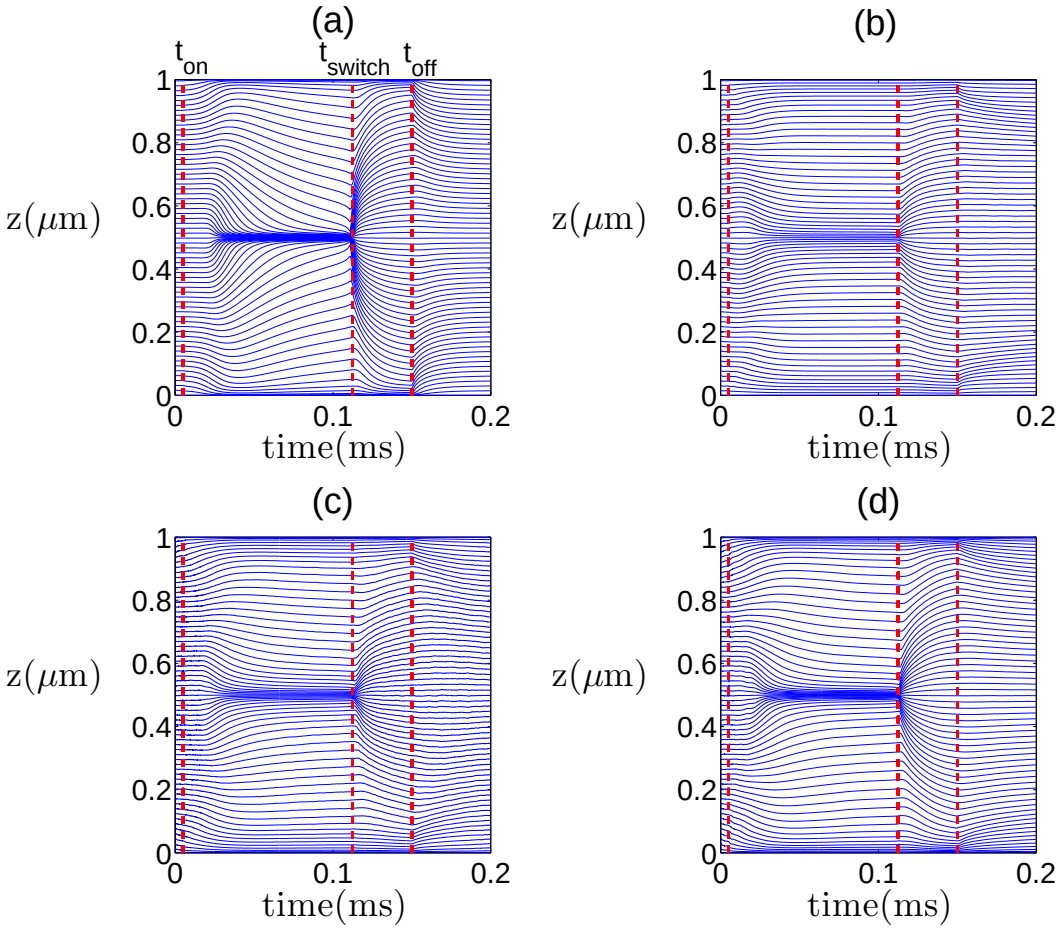

Figure 5.5: Node trajectories with 256 quadratic elements for monitor functions (a) AL, (b) BM1, (c) BM2a, (d) BM2b. The electric field strength is 11.35 $V\ \mu m^{-1}$.

5.4.2.2 Behaviour of adaptive time-stepping

Figure 5.6 shows the variation in time-step size as the calculation proceeds through the first 0.2 milliseconds of the simulation using the four monitor functions and meshes with 256 elements. Vertical lines have again been added to indicate the times when the electric field is switched on, when switching occurs, and when the field is switched off.

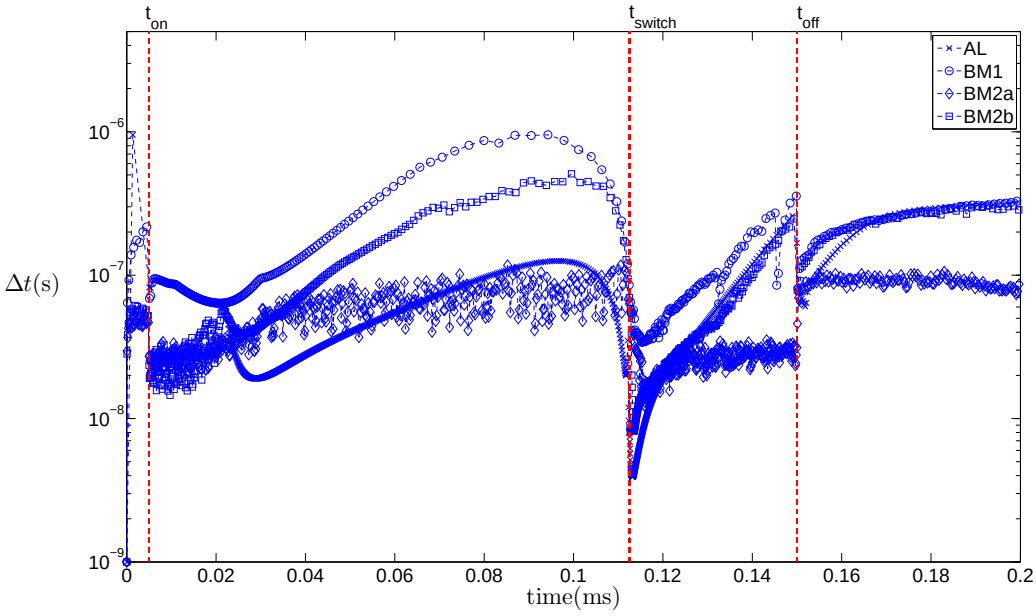

Figure 5.6: Evolution of the adaptive time-step for all four choices of monitor function with electric field strength 11.35 $V\ \mu m^{-1}$ and 256 quadratic elements.

An interpretation of the time-step history is best made in conjunction with the mesh trajectories presented in Figure 5.5. Note that, because time-step adaptivity is based on error indicators of the solution and the mesh, exactly how the mesh

moves will have a significant bearing on the calculation of the time-step size. We first observe that the mesh generated with BM1 evolves much more gradually than those stemming from AL or BM2, and so allows larger time-steps to be taken. However, it will be seen in the next subsection that this is done at the expense of accuracy in that the mesh does not reproduce the features of interest well. As discussed in §5.4.2.1, both AL and BM2b lead to meshes which adapt well to boundary and interior layers but the AL mesh does so much more rapidly, thus requiring smaller time-steps to be taken. As the mesh using BM2b adapts smoothly, larger time-steps can be used. In Table 5.1 we compare the total number of time steps needed using the various monitor functions.

Monitor Function	Time steps
AL	2140
BM1	635
BM2a	2175
BM2b	1409

Table 5.1: Comparison of the number of time steps used with 256 quadratic elements and electric field strength 11.35 $V\ \mu m^{-1}$.

5.4.2.3 Biaxiality

Figure 5.7 and 5.8 show a cross-section of the biaxiality at the cell centre. The approximations shown were computed using 256 quadratic elements with electric field strength 11.35 $V\ \mu m^{-1}$. The results in Figure 5.7 are taken at the switching time observed for each individual monitor function for the case $N = 256$, i.e., after 0.11, 0.1126, 0.1125, 0.1125 milliseconds for AL, BM1, BM2a, BM2b respectively. The results in Figure 5.8 are taken after 0.1125 milliseconds; we see from Table 5.2 that the method is converging towards this switching time for the BM monitor functions (interestingly AL actually appears to not be converging to a particular switching time, and is therefore heavily penalised in the comparison in Figure 5.8). The reference fine grid solution is indicated by a dashed line.

In terms of the first three monitor functions (based on input function $\mathrm{tr}(\boldsymbol{Q}^2)$), it is clear that the monitor function BM1 does a particularly poor job of resolving the rapid change in the biaxiality, with BM2a working most effectively. Furthermore, we observe that BM2b provides the best approximation overall: there is very little difference between BM2b and the reference solution, apart from in the resolution of the planar uniaxial point (where $b = 0$). This illustrates the importance of choosing an appropriate input function. We note also the slight asymmetry of the results obtained using all four monitor functions. This is in fact a physical effect caused by the flexoelectric term in the electric energy term (symmetric solutions

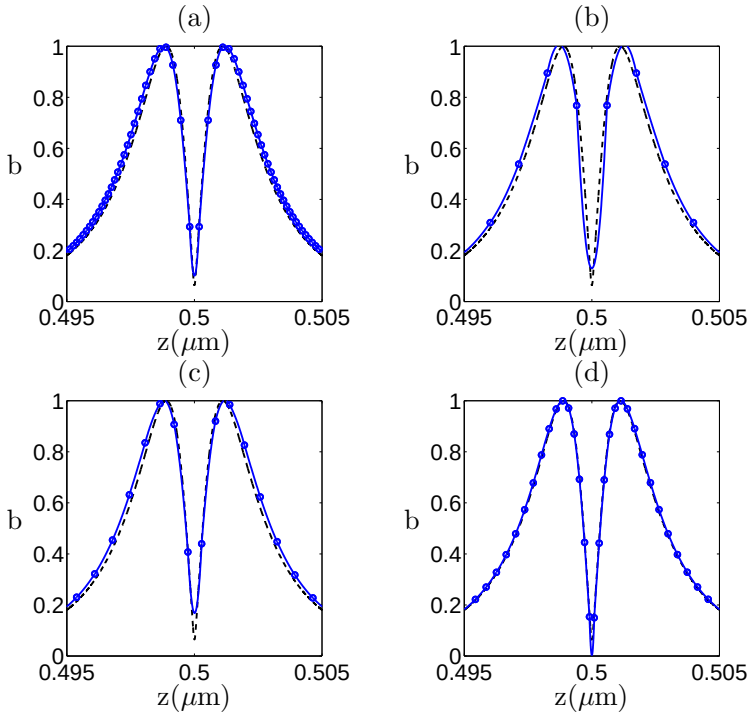

Figure 5.7: Cross-section of biaxiality at the cell centre. Measurements are taken at the switching time observed for each monitor function: (a) AL, (b) BM1, (c) BM2a and (d) BM2b. All grids have 256 quadratic elements and the electric field strength 11.35 $V\ \mu m^{-1}$.

are obtained when $\bar{e} = 0$). As the transition through biaxiality takes places, two eigenvalues of $\boldsymbol{Q}$ at the cell centre are exchanged. This exchange of eigenvalues is illustrated in Figure 5.9 (cf. [9, Figure 8]). This plot was produced using the BM2b monitor function with 256 quadratic elements. Analogous plots using the other three monitor functions tested here (AL, BM1 and BM2a) are indistinguishable from Figure 5.9 on this scale.

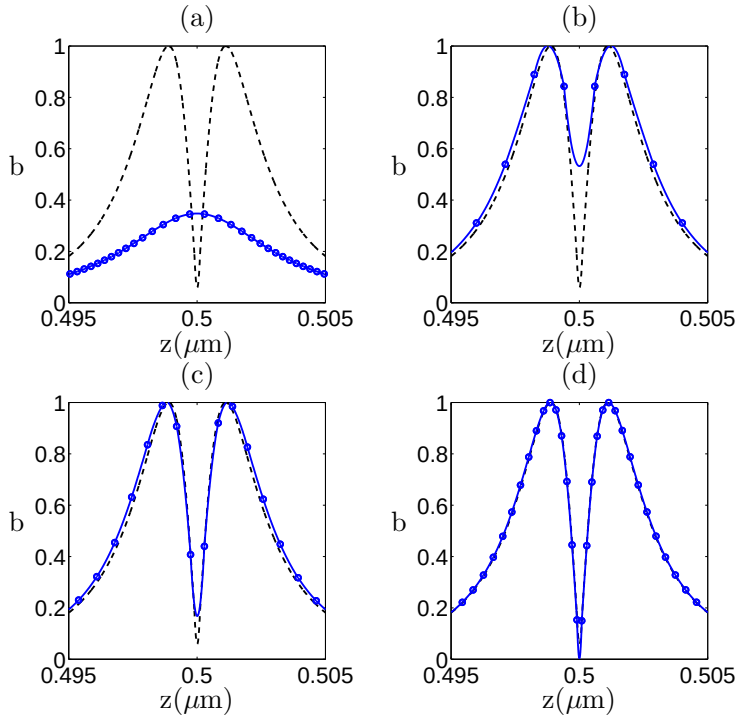

Figure 5.8: Cross-section of biaxiality at the cell centre. Measurements are taken after 0.1125 milliseconds for monitor functions (a) AL, (b) BM1, (c) BM2a and (d) BM2b. All grids have 256 quadratic elements and the electric field strength 11.35 $V\ \mu m^{-1}$.

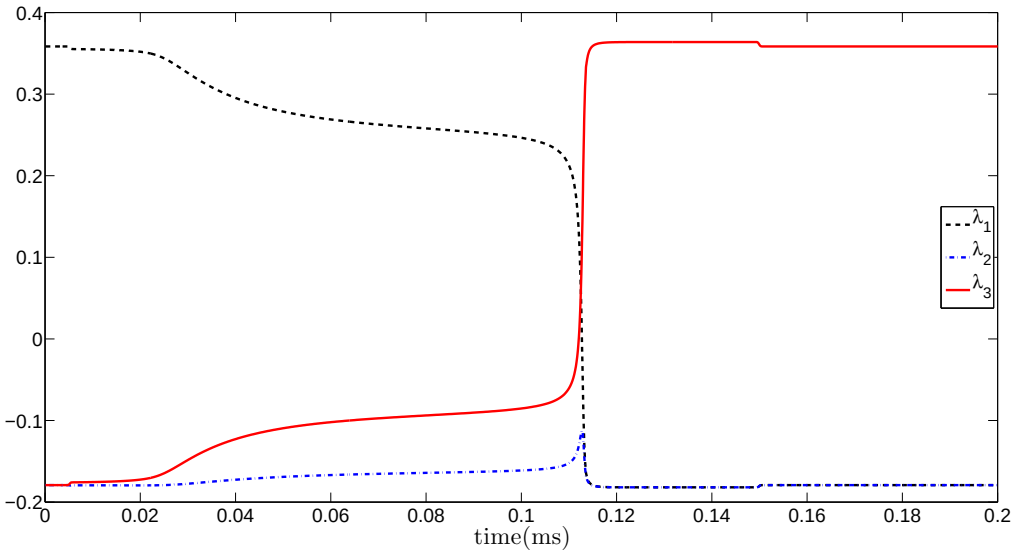

Figure 5.9: Eigenvalues of $\boldsymbol{Q}$ at the cell centre with electric field strength 11.35 $V\ \mu m^{-1}$ using the BM2b monitor function with 256 quadratic elements.

5.4.2.4 Switching time

One of the key challenges in the practical design of liquid crystal cells for displays is the accurate prediction of the switching time. In [61] the authors observe that the use of an over-coarse or poorly adapted grid can lead to poor prediction of switching times, or failure to capture switching altogether. We observe similar behaviour in Table 5.2, which presents the observed switching times using the four monitor functions under investigation on finite element grids with 256 quadratic elements and electric field strength 11.35 $V\ \mu m^{-1}$. It can be seen that for a poorly

	Monitor Function			
N	AL	BM1	BM2a	BM2b
64	0.1109	no switching	0.1248	0.1150
128	0.1108	0.1159	0.1127	0.1126
256	0.1100	0.1126	0.1125	0.1125
512	0.1109	0.1125	0.1125	0.1125

Table 5.2: Switching times (in milliseconds) for an electric field of strength 11.35 V μm^{-1}.

adapted coarse grid, that is, BM1 with 64 elements, switching does not occur at all. For the other coarse grids, switching occurs later than is observed using the reference fine grid solution. As the number of elements is increased, however, the

switching time appears to converge to a fixed value of $t = 0.1125$ ms (for the BM monitor functions), whilst switching with AL occurs slightly earlier at $t = 0.1109$ ms.

5.4.2.5 Efficiency

As well as comparing the accuracy of results obtained with the various monitor functions, it is essential that we also consider the computational cost of each method. Figure 5.10 shows the L_∞ error in $b(z,t)$ (computed in an analogous way to (5.22)) against the total CPU time in seconds required for each method, and where we assume the switching time to be 0.1125 milliseconds. At the switching time we expect to see $b = 0$ (or close to zero) at approximately the cell centre.

As before, the errors are obtained by comparing against a fine-grid reference solution. Note that results are omitted for cases where no switching occurred. It is clear that the BM monitor functions are performing significantly better than AL or the uniform grid in this comparison, with BM2b the best choice. It is unsurprising that AL performs so poorly in this comparison, as after 0.1125 milliseconds the order reconstruction has already taken place, and the biaxiality is beginning to relax throughout the cell.

Figure 5.11 shows a similar cost comparison of the monitor functions, but is instead plotted at switching time predicted by each monitor function (presented in Table 5.2). Unsurprisingly AL appears to have improved significantly in this

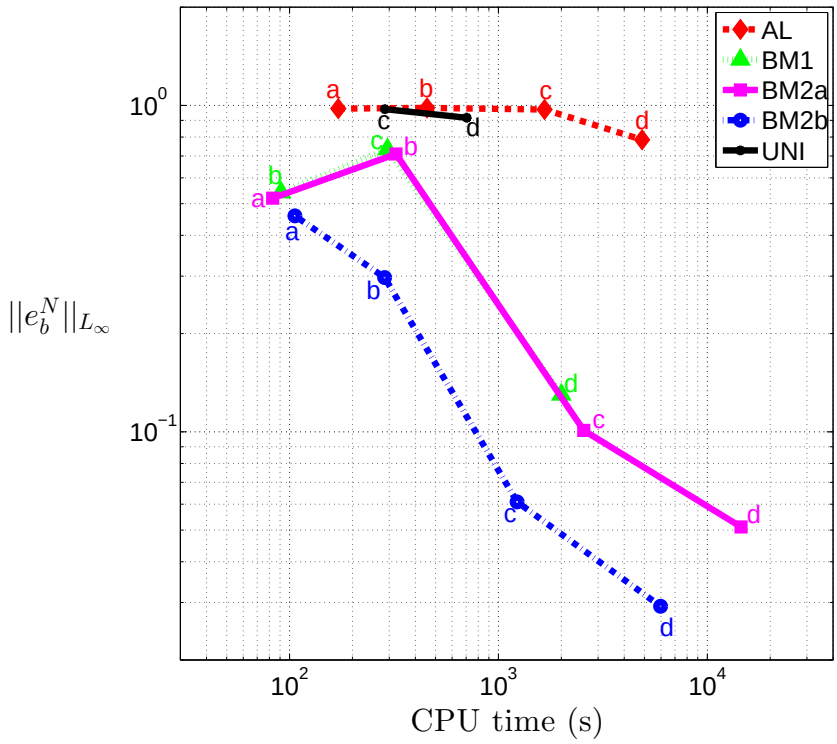

Figure 5.10: Cost comparison for the four choices of monitor function, with results obtained after 0.1125 milliseconds.

plot, as AL does not appear to be converging to the $t = 0.1125$ ms switching time. When the number of elements is sufficiently large, AL and BM2a are the most accurate of the monitor functions based on $\mathrm{tr}(\boldsymbol{Q}^2)$, with AL being the more efficient of the two. However, using BM2b is better still. Overall, it is clear that in order to calculate b to a given degree of accuracy, using the BM2b monitor function leads to the most accurate and efficient method, which is not surprising given that BM2b is based on input function (5.19) so is specifically tailored to

model changes in biaxility.

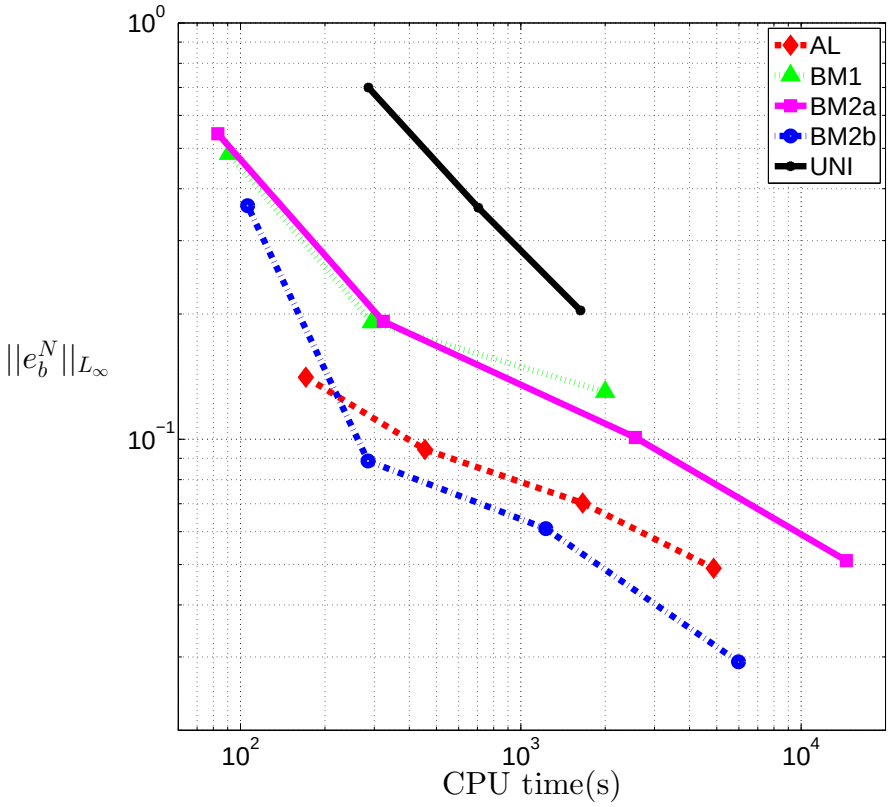

Figure 5.11: Cost comparison for the four choices of monitor function, with results obtained at the switching time of each individual monitor function.

5.5 Conclusions

In this chapter, an adaptive moving mesh method has been developed to tackle one-dimensional problems modelled using $\boldsymbol{Q}$-tensor theory of liquid crystals. An MMPDE approach has been used to generate the moving mesh where the equations

have been discretised using second-order finite differences in space and first-order backward Euler time integration. To capture the highly non-linear nature of the $\boldsymbol{Q}$-tensor equations, a conservative finite element discretisation using quadratic elements has been used to update the solution on the adaptive moving mesh. Time integration of the $\boldsymbol{Q}$-tensor equations has been achieved using a second-order semi-implicit Runge-Kutta scheme and adaptive time-step control. These components have been put together to form an adaptive algorithm that has been carefully tested, and the computed solutions have been shown to converge at optimal rates in both space and time. These experiments confirm our previous findings for a much simpler scalar problem, namely that it is not necessary to approximate the MMPDE equation with the same spatial or temporal degree of accuracy compared to that used to discretise the governing PDEs to ensure optimal rates of convergence [14]. Evidence has also been given to suggest that the computed solutions exibit nodal superconvergence, which is somewhat surprising given the highly non-uniform nature of the adaptive moving meshes. For the first time, a monitor function has been constructed based upon a local measure of biaxiality. This has been shown to lead to higher levels of solution accuracy and a considerable improvement in computational efficiency compared to those monitor functions used previously for liquid crystal problems. The results in this chapter have been published in [49].

Chapter 6

Resolving defects in two dimensions

We have already mentioned that defects in liquid crystals are a subject of great interest, and it is crucial that the behaviour and nature of defects can be accurately represented by our chosen numerical method. Schopohl & Sluckin [68] show that defects tend to occur over length scales of only a few nematic coherence lengths, that is, over length scales of around $10 - 100$ nm. The size of the defect is thus small compared to the size of a liquid crystal cell which is typically of length $1 - 10$ μm. The presence of characteristic lengths with large scale differences poses numerical problems, and to resolve defects accurately and efficiently is non-trivial; considerable care is therefore required when choosing the most appropriate numerical approach. The work presented in previous chapters shows that, with

a sensible choice of monitor function, we can adequately resolve defect-like small-scale structures in one dimension. In this chapter we extend our MMPDE approach and the conservative finite element discretisation of the $\boldsymbol{Q}$-tensor equations to a two-dimensional setting. Adopting a similar strategy to previous chapters, we apply a number of different monitor functions to the problem, and present analysis which indicates that monitor functions based on a local measure of biaxiality perform well. We then apply the biaxiality-based monitor function to a problem first presented by Bos [80]: a two dimensional π-cell problem with a sinusoidal perturbation across the centre of the cell.

6.1 Moving the mesh

6.1.1 Moving mesh partial differential equations

In the absence of an analogous formulation of the equidistribution principle in higher dimensions, a variational formulation is often used to obtain an MMPDE to drive mesh movement. It is again useful to consider the physical domain Ω_p as the image of the computational domain Ω_c under a suitably defined invertible mapping, i.e.,

$$x = x(\xi, \eta, t), \quad y = y(\xi, \eta, t), \quad (\xi, \eta) \in \Omega_c, \tag{6.1}$$

with $\boldsymbol{x}(t) = (x(t), y(t))^T$ and $\boldsymbol{\xi} = (\xi, \eta)^T$. The mapping $\boldsymbol{\xi}(\boldsymbol{x}, t)$ is chosen in order to minimise the functional

$$I[\boldsymbol{\xi}(\boldsymbol{x}, t)] = \frac{1}{2} \int_{\Omega_p} \sum_{i=1}^{2} (\nabla \xi_i)^T G^{-1} (\nabla \xi_i) \, \mathrm{d}\boldsymbol{x}, \tag{6.2}$$

where G is a 2×2 symmetric positive definite monitor matrix. We discuss the choice of G later in the chapter. Following the same process as discussed in §5.2.1, we can obtain a MMPDE from the modified gradient flow equations

$$\frac{\partial \xi}{\partial t} = -\frac{P}{\tau} \frac{\delta I}{\delta \xi} \quad \text{and} \quad \frac{\partial \eta}{\partial t} = -\frac{P}{\tau} \frac{\delta I}{\delta \eta}. \tag{6.3}$$

Recall that $\tau > 0$ is a user-specified temporal smoothing parameter which affects the temporal scale over which the mesh adapts, and P is a positive function of $(\boldsymbol{x}, t)$, chosen such that the mesh movement has a spatially uniform time scale [38]. The Euler-Lagrange equations associated with (6.2) are

$$\nabla \cdot \left(G^{-1} \nabla \xi_i \right) = 0, \quad i = 1, 2, \tag{6.4}$$

and, from (6.3) and (6.4), a MMPDE can be defined as

$$\frac{\partial \xi_i}{\partial t} = \frac{P}{\tau} \nabla \cdot \left(G^{-1} \nabla \xi_i \right), \quad i = 1, 2. \tag{6.5}$$

The selection of an appropriate monitor matrix is crucial to the success of mesh adaption. In this work, we consider the monitor matrix proposed by Winslow [78],

$$G = \begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix}, \tag{6.6}$$

where $\rho(\boldsymbol{x}, t)$ is a strictly positive *monitor function*. The choice of monitor function should ideally be based on a local a posteriori error estimate but if no such estimate exists then the monitor function can be any smooth function designed to adapt the mesh towards important solution features.

In practice, we interchange the roles of the dependent and independent variables in (6.5), since it is the location of the physical mesh points $\{\boldsymbol{x}_i(t)\}_{i=1}^{\mathcal{N}}$ that defines the mapping $\mathcal{A}_t$ (see (3.1)). With a Winslow-type monitor matrix, the resulting MMPDEs take the form

$$\tau \frac{\partial \boldsymbol{x}}{\partial t} = P\left(A\boldsymbol{x}_{\xi\xi} + B\boldsymbol{x}_{\xi\eta} + C\boldsymbol{x}_{\eta\eta} + D\boldsymbol{x}_{\xi} + E\boldsymbol{x}_{\eta}\right), \qquad (\xi, \eta) \in \Omega_c, \tag{6.7}$$

where

$$\begin{aligned}
A &= \frac{x_\eta^2 + y_\eta^2}{\rho J^2}, \\
B &= \frac{-2\left(x_\xi^2 x_\eta^2 + y_\xi^2 y_\eta^2\right)}{\rho J^2}, \\
C &= \frac{x_\xi^2 + y_\xi^2}{\rho J^2}, \\
D &= \frac{\left(x_\xi^2 x_\eta^2 + y_\xi^2 y_\eta^2\right)\left(\frac{\partial}{\partial \eta}\left(\rho^{-1}\right)\right) - \left(x_\eta^2 + y_\eta^2\right)\left(\frac{\partial}{\partial \xi}\left(\rho^{-1}\right)\right)}{J^2}, \\
E &= \frac{\left(x_\xi^2 x_\eta^2 + y_\xi^2 y_\eta^2\right)\left(\frac{\partial}{\partial \xi}\left(\rho^{-1}\right)\right) - \left(x_\xi^2 + y_\xi^2\right)\left(\frac{\partial}{\partial \eta}\left(\rho^{-1}\right)\right)}{J^2},
\end{aligned}$$

and $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian of $\mathcal{A}_t$. To complete the specification of the coordinate transformation, the MMPDE must be supplemented by suitable

boundary conditions $\boldsymbol{g}(\boldsymbol{\xi}, t), \boldsymbol{\xi} \in \partial\Omega_c$; these are obtained using a one dimensional moving mesh approach, using the same one dimensional MMPDE as in Chapter 5.

6.1.2 Spatial balancing

The choice of the spatial balancing coefficient P in (6.5) is essential to a reliable adaptive strategy. The motivation behind choosing a suitable P is to have the entire mesh adapting with a uniform time scale, thereby reducing the stiffness of the MMPDE and making it more robust for a constant choice of temporal smoothing parameter τ. Huang and Russell note in [40] that a heuristic approach to choosing P should be taken in the absence of any clear guidance on how to force a uniform time scale upon the MMPDE. Experimentally we find that a choice of $P = (\rho J)^2$ produces a mesh that adapts smoothly. This choice of P results in the MMPDE

$$\tau \frac{\partial \boldsymbol{x}}{\partial t} = \left(A\boldsymbol{x}_{\xi\xi} + B\boldsymbol{x}_{\xi\eta} + C\boldsymbol{x}_{\eta\eta} + D\boldsymbol{x}_{\xi} + E\boldsymbol{x}_{\eta} \right), \tag{6.8}$$

where

$$\begin{aligned}
A &= \rho(x_\eta^2 + y_\eta^2), \\
B &= -2\rho\left(x_\xi^2 x_\eta^2 + y_\xi^2 y_\eta^2\right), \\
C &= \rho(x_\xi^2 + y_\xi^2), \\
D &= \rho^2\left(x_\xi^2 x_\eta^2 + y_\xi^2 y_\eta^2\right)\left(\frac{\partial}{\partial\eta}\left(\rho^{-1}\right)\right) - \left(x_\eta^2 + y_\eta^2\right)\left(\frac{\partial}{\partial\xi}\left(\rho^{-1}\right)\right), \\
E &= \rho^2\left(x_\xi^2 x_\eta^2 + y_\xi^2 y_\eta^2\right)\left(\frac{\partial}{\partial\xi}\left(\rho^{-1}\right)\right) - \left(x_\xi^2 + y_\xi^2\right)\left(\frac{\partial}{\partial\eta}\left(\rho^{-1}\right)\right).
\end{aligned}$$

6.1.3 Choice of monitor function

Here we consider two-dimensional analogues of the monitor functions presented in [48] and in the previous chapter. These are based on a function $\mathcal{T}(\boldsymbol{x}, t)$ representing a particular physical quantity derived from the finite element approximation of the $\boldsymbol{Q}$-tensor matrix. We consider the commonly-used solution arc-length monitor function

$$\rho(\mathcal{T}(\boldsymbol{x}, t)) = \left(1 + |\nabla\mathcal{T}(\boldsymbol{x}, t)|^2\right)^{\frac{1}{2}}. \tag{6.9}$$

We also consider a generalisation of the Beckett-Mackenzie monitor function based on first order partial derivatives of $\mathcal{T}(\boldsymbol{x}, t)$ which takes the form

$$\rho(\mathcal{T}(\boldsymbol{x}, t)) = \alpha(\boldsymbol{x}, t) + |\nabla\mathcal{T}(\boldsymbol{x}, t)|^{\frac{1}{m}}, \tag{6.10}$$

which we again refer to as the BM1 monitor function. Lastly we consider the monitor function

$$\rho(\mathcal{T}(\boldsymbol{x},t)) = \alpha(\boldsymbol{x},t) + \|H(\mathcal{T}(\boldsymbol{x},t))\|_{\mathrm{F}}^{\frac{1}{m}}, \tag{6.11}$$

where the Hessian matrix

$$H(\mathcal{T}(\boldsymbol{x},t)) = \begin{bmatrix} \frac{\partial^2 \mathcal{T}}{\partial x^2} & \frac{\partial^2 \mathcal{T}}{\partial x \partial y} \\ \\ \frac{\partial^2 \mathcal{T}}{\partial x \partial y} & \frac{\partial^2 \mathcal{T}}{\partial y^2} \end{bmatrix}, \tag{6.12}$$

and the Frobenius norm of the Hessian matrix is

$$\|H(\mathcal{T}(\boldsymbol{x},t))\|_{\mathrm{F}} = \sqrt{\left(\frac{\partial^2 \mathcal{T}}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 \mathcal{T}}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 \mathcal{T}}{\partial y^2}\right)^2}. \tag{6.13}$$

We refer to this as the BM2 monitor function, as it is based on second derivatives of $\mathcal{T}(\boldsymbol{x},t)$. The motivation for a suitable choice of $\alpha(\boldsymbol{x},t)$ in (6.10) and (6.11) is, as in previous chapters, to avoid mesh starvation away from the layers in the solution. In (6.10) we choose

$$\alpha(\boldsymbol{x},t) = \max\left[1, \frac{1}{\mathrm{area}(\Omega_p)} \int_{\Omega_p} (|\nabla \mathcal{T}(\boldsymbol{x},t)|)^{\frac{1}{m}} \, \mathrm{d}\boldsymbol{x}\right], \tag{6.14}$$

and in (6.11),

$$\alpha(\boldsymbol{x},t) = \max\left[1, \frac{1}{\mathrm{area}(\Omega_p)} \int_{\Omega_p} \|H(\mathcal{T}(\boldsymbol{x},t))\|_{\mathrm{F}}^{\frac{1}{m}} \mathrm{d}\boldsymbol{x}\right]. \tag{6.15}$$

With no specific guidance on the choice of m in higher-dimensional settings we choose $m = 3$.

6.1.3.1 Smoothing the monitor function

The monitor function often has large spatial and temporal variations, and so we employ both spatial and temporal smoothing. The benefits of doing so are that a smoother monitor function makes for a MMPDE that is easier to integrate and a smoother monitor function will also generate a smoother mesh. Temporal smoothing is done by relaxing the monitor function so that the monitor function at the current time level n is given by

$$\rho^n = (1-\omega)\rho^n + \omega\rho^{n-1}, \tag{6.16}$$

where $0 < \omega < 1$ is a relaxation parameter.

Spatial smoothing is achieved by taking a local average of the monitor function across elements within a certain radius of a mesh point. That is, the smoothed monitor function $\tilde{\rho}$ is defined as

$$\tilde{\rho}(\boldsymbol{x}_m) = \frac{\int_{C(\boldsymbol{\xi}_m, r)} \rho(\boldsymbol{x}(\boldsymbol{\xi}))\mathrm{d}\boldsymbol{\xi}}{\int_{C(\boldsymbol{\xi}_m, r)} \mathrm{d}\boldsymbol{\xi}}, \tag{6.17}$$

where $\boldsymbol{x}_m \in \Omega_p$ is a mesh point in the physical domain, $\boldsymbol{\xi}_m \in \Omega_c$ is the corresponding mesh point in the computational domain, and $C(\boldsymbol{\xi}_m, r) \subset \Omega_c$ represents the area covered by all the cells that lie fully, or partially, within a circle of radius r around the mesh point $\boldsymbol{\xi}_m$. If required, spatial smoothing can be repeated in an iterative fashion to further smooth the monitor function. We note that in [38] the authors consider a smoothing strategy where, instead of smoothing over a circle about a given vertex, they smooth over all the neighbouring cells of a vertex.

We find, for structured triangular grids, that this strategy results in the monitor function being smoothed excessively in the direction of the triangle diagonals.

6.1.4 Discretisation of the MMPDE

The numerical solution of the chosen MMPDE requires spatial and temporal discretisation. In space, we discretise using standard linear Galerkin finite elements. In time, we use a backward Euler integration scheme to update the solution at $t = t^{n+1}$ and, to avoid solving nonlinear algebraic systems, we evaluate the coefficients $A, B, \ldots, E$ at the time $t = t^n$. We therefore seek $\boldsymbol{x}_h^{n+1} \in \left(\mathcal{L}^1\left(\Omega_c\right)\right)^2$ such that

$$\begin{aligned}\tau \int_{\Omega_c} \left(\frac{\boldsymbol{x}_h^{n+1} - \boldsymbol{x}_h^n}{\Delta t} \right) \cdot \hat{\boldsymbol{v}}_h \mathrm{d}\boldsymbol{\xi} + \int_{\Omega_c} \Big[\left(\boldsymbol{x}_h^{n+1}\right)_\xi \cdot \left(A^n \hat{\boldsymbol{v}}_h\right)_\xi + \left(\boldsymbol{x}_h^{n+1}\right)_\eta \cdot \left(C^n \hat{\boldsymbol{v}}_h\right)_\eta \\ + \tfrac{1}{2} \Big[\left(\boldsymbol{x}_h^{n+1}\right)_\xi \cdot \left(B^n \hat{\boldsymbol{v}}_h\right)_\eta + \left(\boldsymbol{x}_h^{n+1}\right)_\eta \cdot \left(B^n \hat{\boldsymbol{v}}_h\right)_\xi \Big] \\ - \Big[D^n \left(\boldsymbol{x}_h^{n+1}\right)_\xi + E^n \left(\boldsymbol{x}_h^{n+1}\right)_\eta \Big] \cdot \hat{\boldsymbol{v}}_h \Big] \mathrm{d}\boldsymbol{\xi} = 0, \quad (6.18)\end{aligned}$$

for all $\hat{\boldsymbol{v}}_h \in \left(\mathcal{L}_0^1\left(\Omega_c\right)\right)^2$. The resulting linear systems are solved using the iterative method BiCGSTAB ([10],[76]) and an incomplete LU (ILU) [65] factorisation as a preconditioner. An analysis of the performance of this iterative solver for the discretised MMPDE equations can be found in [15].

6.1.5 Integration of PPDE

To integrate the physical equations (3.13a and 3.13b) forward in time, we employ a second-order singly diagonally implicit Runge-Kutta (SDIRK2) method similar to that used in the previous chapter. Due to the high cost of solving the physical PDEs in two dimensions we simply set MAXPASS = 1 in the iterative solution algorithm described in §5.3.

6.2 Stationary defect test problem

We first consider the resolution of stationary defects: this problem is ideal for examining the ability of selected monitor functions to adapt to the small scale structures present in liquid crystals, and it provides a setting for us to conduct rigorous convergence tests of our numerical method.

The defect we consider is the same wedge disclination line studied in some detail by Schopohl & Sluckin [68]. The disclination line is in the z direction, however the director $\boldsymbol{n}$ lies in the $x-y$ plane. We will therefore seek solutions to the $\boldsymbol{Q}$-tensor equations (2.18) in the $x-y$ plane. We generate the defect by imposing an inital condition on the director

$$\boldsymbol{n}(\theta, 0) = (\cos(m\theta), \sin(m\theta), 0)\,, \tag{6.19}$$

where m is integral or half integral, and is referred to as the disclination index.

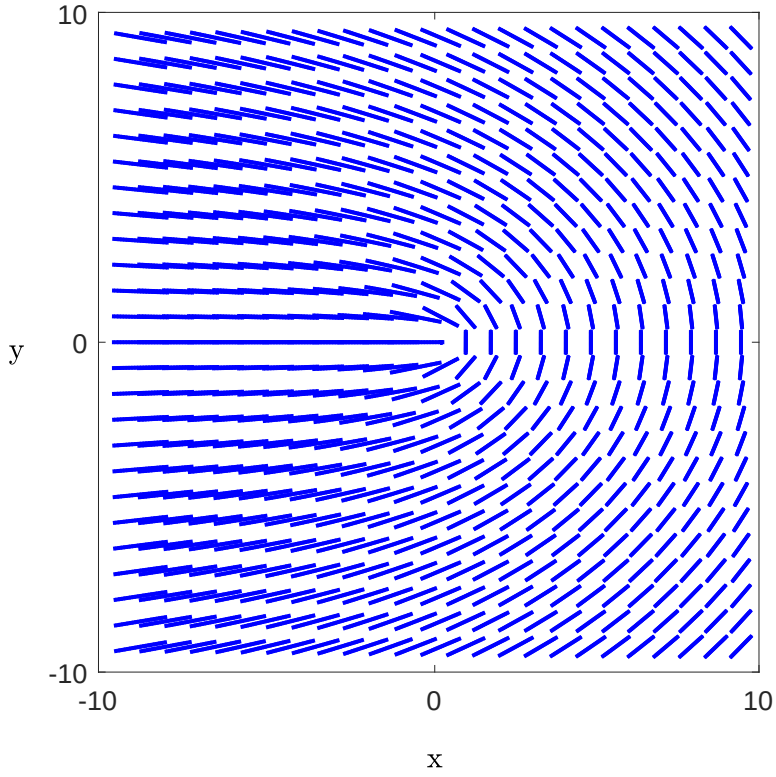

Figure 6.1: Director plot for +1/2 defect. Length scale measured in units of the nematic coherence length, ζ.

Travelling around the defect core rotates the director through the angle $2\pi m$: we restrict our attention to the case $m = +\frac{1}{2}$. Studies were also carried out for the case $m = -\frac{1}{2}$, but the results are quantitatively so similar that they have been omitted for brevity. In all experiments we consider a liquid crystal cell of width 20ζ and height 20ζ, where $\zeta \approx 4.06$ nm is the nematic coherence length.

We will attempt to resolve the stationary defect with the monitor functions

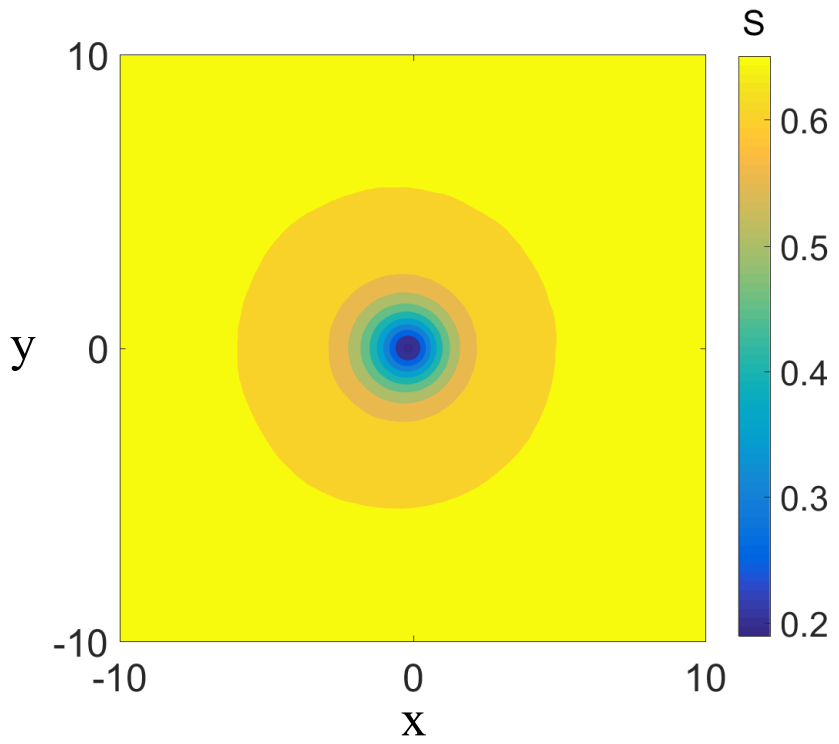

Figure 6.2: Contour plot of the order parameter, $S = (\mathrm{tr}(\boldsymbol{Q}^2))^{\frac{1}{2}}$, profile for the $+1/2$ defect. Length scale measured in units of the nematic coherence length, ζ.

presented in §6.1.3. Essential to the success of our adaptive method is a suitable choice of $\mathcal{T}(\boldsymbol{x}, t)$ in two dimensions on which we base our monitor functions. This represents a particular physical quantity derived from the finite element approximation of $\boldsymbol{Q}$. In Chapter 5, we presented detailed numerical results in one dimension for choices of $\mathcal{T}(\boldsymbol{x}, t) = \mathrm{tr}(\boldsymbol{Q}^2)$ and $\mathcal{T}(\boldsymbol{x}, t) = b(\boldsymbol{x}, t)$, where b is the invariant measure of biaxiality and

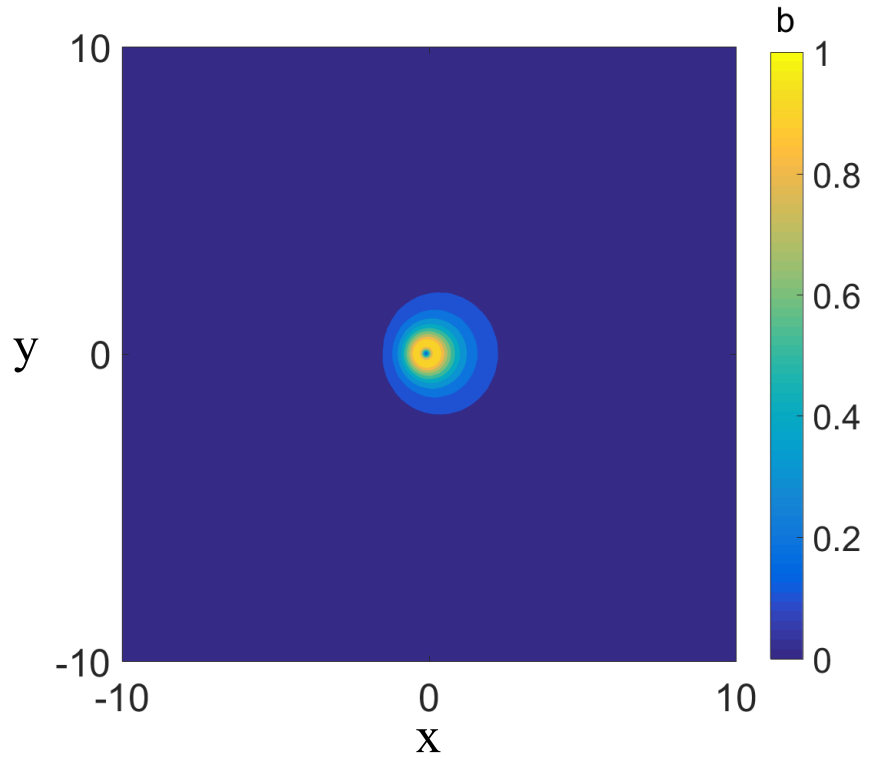

Figure 6.3: Contour plot of the biaxiality, b, for the $+1/2$ defect. Length scale measured in units of the nematic coherence length, ζ.

$$b(\boldsymbol{x}, t) = \left[1 - \frac{6\,\mathrm{tr}(\boldsymbol{Q}^3)^2}{\mathrm{tr}(\boldsymbol{Q}^2)^3}\right]^{\frac{1}{2}}. \tag{6.20}$$

As previously noted $b \in [0, 1]$, with uniaxial states corresponding to $b = 0$ and totally biaxial states corresponding to $b = 1$. In Figure 6.2 and Figure 6.3 we plot the quantities S and b respectively, for the $+\frac{1}{2}$ defect, and observe the length scales over which both quantities vary. Both figures were obtained using an adaptive mesh of 1388 quadratic triangular elements and the BM2b monitor

function, i.e., (6.11) with $\mathcal{T}(x, y, t) = b(x, y, t)$. The adaptive algorithm was run until time $t = 0.05$ ms with a uniform timestep of $\Delta t = 10^{-8}$s: by this point the solution has entered a quasi-steady state. We observe that the order parameter takes its equilibrium value $S \approx 0.65$ outside a central circular region of diameter 10ζ. Within this region the order parameter varies significantly within a core with diameter approximately 2ζ. Outside a core of diameter of approximately 4ζ the biaxiality is zero, and inside it has a volcano-like structure with a rim at $b = 1$ representing the purely biaxial state, and a base at $b = 0$ representing the uniaxial state: this sudden variation takes place over a core only a few nanometres in diameter. We note the similarities to the structure of the biaxiality observed in Chapter 5 in a one dimensional setting.

We plot the eigenvalues of $\boldsymbol{Q}$ along the line $x = 0$ in Figure 6.4. As we would expect, we see that two of the eigenvalues of $\boldsymbol{Q}$ are exchanged at the centre of the defect core.

From the configuration shown in Figure 6.1, we expect that the solution will be symmetric about the x-axis, yet not necessarily symmetric about the y-axis. From the results presented in Figures 6.2 and 6.3, we observe that the solution is symmetric about the x-axis; however, it is clearly not radially symmetric. We have conducted additional numerical experiments where we have rotated the $+\frac{1}{2}$ defect structure, and concluded that the solution is not radially symmetric (the results obtained when rotating the defect structure are symmetric about a single

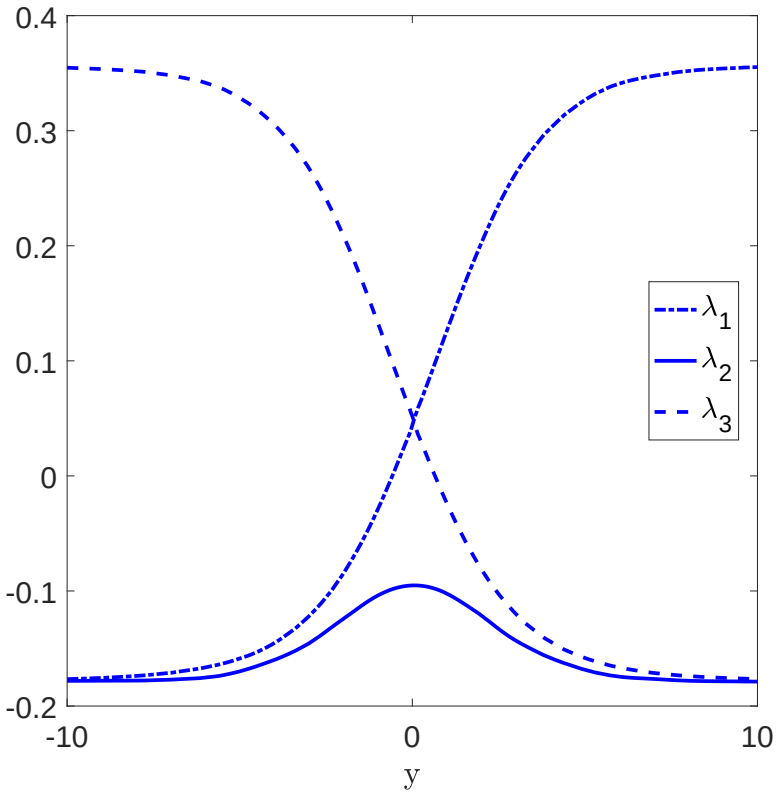

Figure 6.4: Eigenvalues of $\boldsymbol{Q}$ along the line $x = 0$ for the $+1/2$ defect. Length scale measured in units of the nematic coherence length, ζ.

axis only). This is an interesting observation worthy of further study.

6.2.1 Numerical Results

6.2.1.1 Spatial convergence rate

In Chapter 4 we presented convergence results for a scalar model of a one-dimensional uniaxial problem. We also presented convergence rates of spatial errors for the much more complicated π-cell order reconstruction problem considered in Chapter 5. In a similar vein, we now consider convergence rates of spatial errors in a two dimensional setting. To our knowledge there is no suitable problem with an analytic solution on which to test the spatial convergence rates of our adaptive method. Here we again compare solutions with a reference solution obtained using a very fine mesh. We calculated our reference solution on an adaptive mesh of $N = 5334$ quadratic triangular elements, using the BM2b monitor function, and a uniform time-step $\Delta t = 10^{-8}$ seconds. We will use $q_{i*}(\boldsymbol{x}, t)$ to denote this reference approximation to $q_i(\boldsymbol{x}, t)$, and assume throughout that

$$|q_{i*}(\boldsymbol{x}, t) - q_i(\boldsymbol{x}, t)| \ll |q_{i*}(\boldsymbol{x}, t) - q_{iN}(\boldsymbol{x}, t)|\,,$$

where q_{iN} denotes the finite element approximation calculated on a grid with N quadratic elements. The error in the approximation q_{iN} will be denoted by $e^N_{q_i}$.

Since the approximate solution grid points will not in general coincide with the reference grid points, the solution $q_{i*}(\boldsymbol{x}, t^*)$ is interpolated using the *MATLAB* function *scatteredInterp*, and the solution is defined on the coarse grid points. We estimate the spatial error in the l_∞ norm using the maximum error computed at

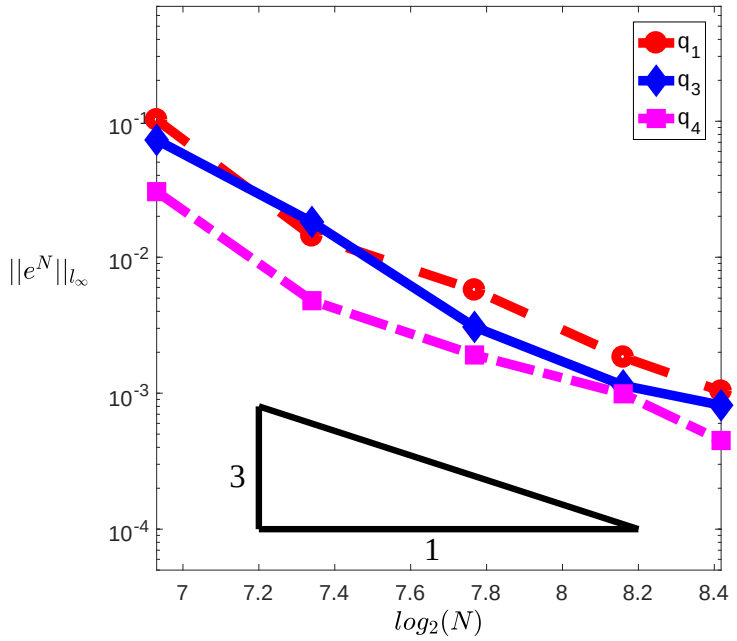

Figure 6.5: Spatial error in the approximation of the +1/2 defect.

the grid nodes (analogous to the one-dimensional procedure used in Chapters 4 and 5), that is,

$$\|\boldsymbol{e}_{q_i}^N\|_{l_\infty} = \max_{j=0,\ldots,N} |q_{i*}(\boldsymbol{x}_j, t^*) - q_{iN}(\boldsymbol{x}_j, t^*)|. \tag{6.21}$$

We examine the error at time $t^* = 0.2$ ms as by this time the solution has entered a quasi-steady state. The error norm (6.21) for the non-zero components of $\boldsymbol{Q}$ (components q_2 and q_5 are exactly zero for this problem) is presented in Figure 6.5 for various values of N for the +1/2 defect. All approximate solutions are obtained using the BM2b monitor function. We observe that $\|\mathbf{e}_{q_i}^N\|_{l_\infty}$ appears to converge at the rate $\mathcal{O}(N^{-3})$, i.e., the optimal rate expected using quadratic

triangular elements.

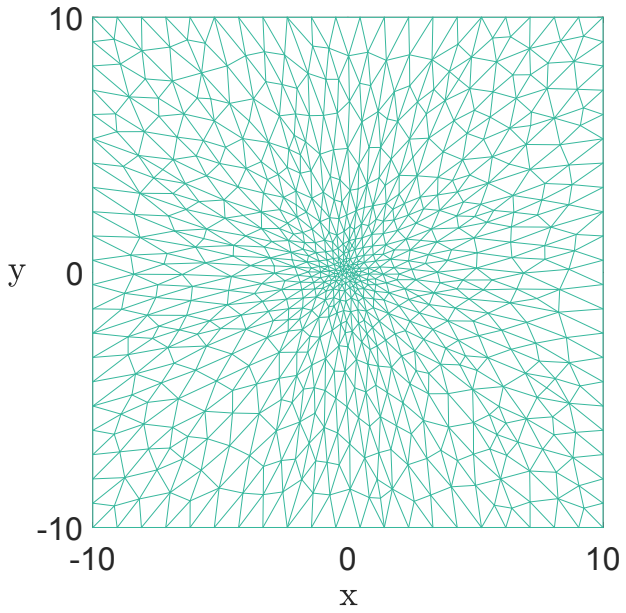

Figure 6.6: Mesh of 1388 quadratic elements for the +1/2 defect, obtained using the BM2b monitor function. Length scale measured in units of the nematic coherence length, ζ.

6.2.1.2 Resolving the defect core

A sample adaptive mesh is presented in Figure 6.6. At this scale it is difficult to observe how the mesh has adapted to resolve the structure of the order parameter and biaxiality. Figures 6.7 and 6.8 show, respectively, the order parameter and the biaxiality along the cross section $y = 0$; we also plot the location of grid nodes and observe how the monitor functions cope with adapting to the small scale structure

of the defect core. The mesh is clearly adapting to resolve the core structure of both the order parameter and the biaxiality. What is particularly pleasing is the ability of the monitor function to place a significant number of nodes in the inside of the volcano structure described by the biaxiality. We know from our experience in one dimension that it is particularly difficult to resolve this structure, and the BM2b monitor function performs well.

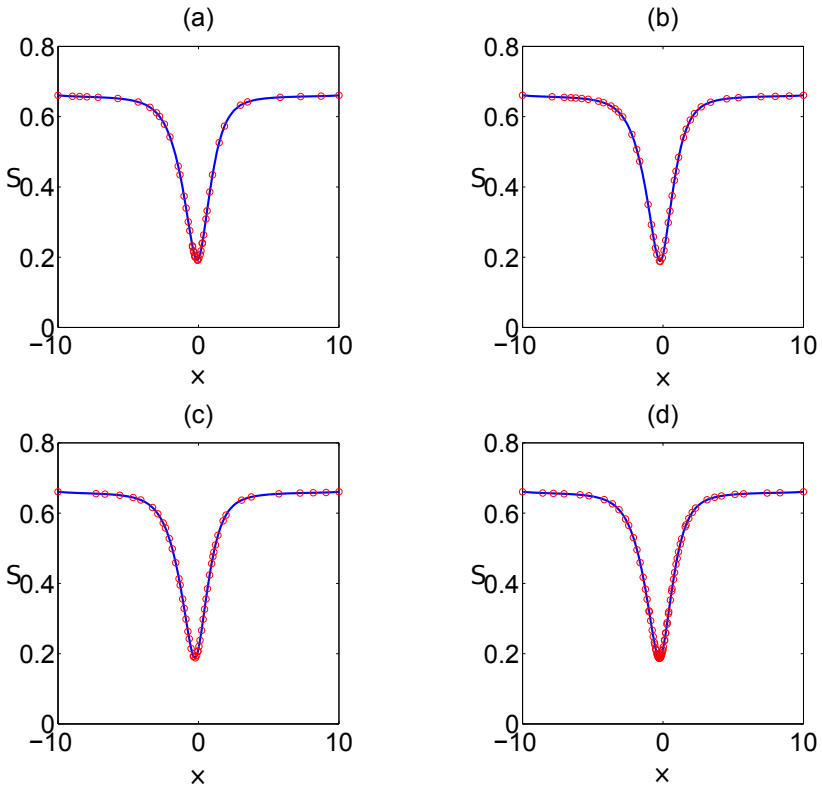

Figure 6.7: Order parameter along the cross section $y = 0$ for the $+1/2$ defect, obtained using 1946 quadratic elements using the monitor functions: (a) AL; (b) BM1a; (c) BM1b; (d) BM2b. Length scale measured in units of the nematic coherence length, ζ.

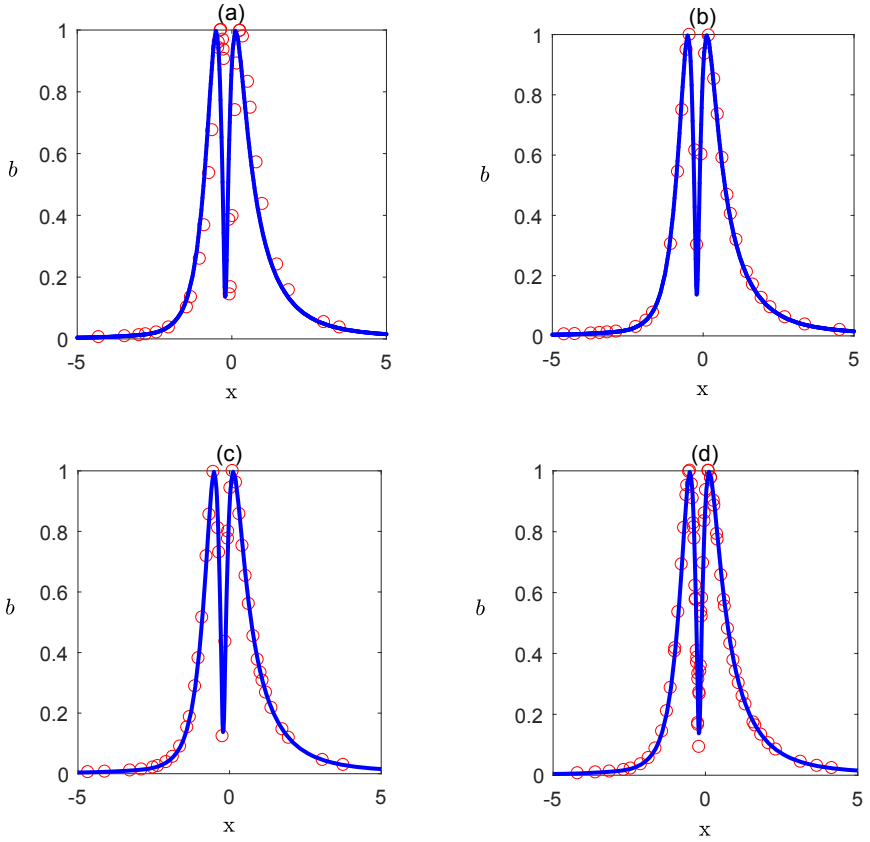

Figure 6.8: Biaxiality along the cross section $y = 0$ for the $+1/2$ defect, obtained using 1946 quadratic elements using the monitor functions: (a) AL; (b) BM1a; (c) BM1b; (d) BM2b. Length scale measured in units of the nematic coherence length, ζ.

6.2.1.3 Efficiency and cost

As well as comparing the grids obtained with each monitor function, it is important to consider the computational cost of each method. Figure 6.8 shows a plot of the l_∞ error in the three non-zero components of the $\boldsymbol{Q}$-tensor, i.e., q_1, q_3 and q_4, against the total CPU time in seconds required for each method. Analysis of the resulting computational cost calculations give us some insight into what monitor function performs the best. For high tolerances, i.e., where the error in the approximate solution is large, we observe that the BM1a and BM1b monitor functions outperform both the AL and BM2b monitor functions; however we note that the magnitude of the error here is undesirably large, so the results at high tolerances are of less significance. More significant are the cost comparisons at low tolerances. As the magnitude of the error decreases, the results clearly show that the BM2b monitor function is the most efficient. For the grid densities chosen, neither BM1a or BM1b is able to reach the level of accuracy obtained by the BM2b monitor function. The AL monitor function comes closest to matching the accuracy of BM2b, but it does so at a far greater cost. The uniform mesh, as would be expected, performs worst of all.

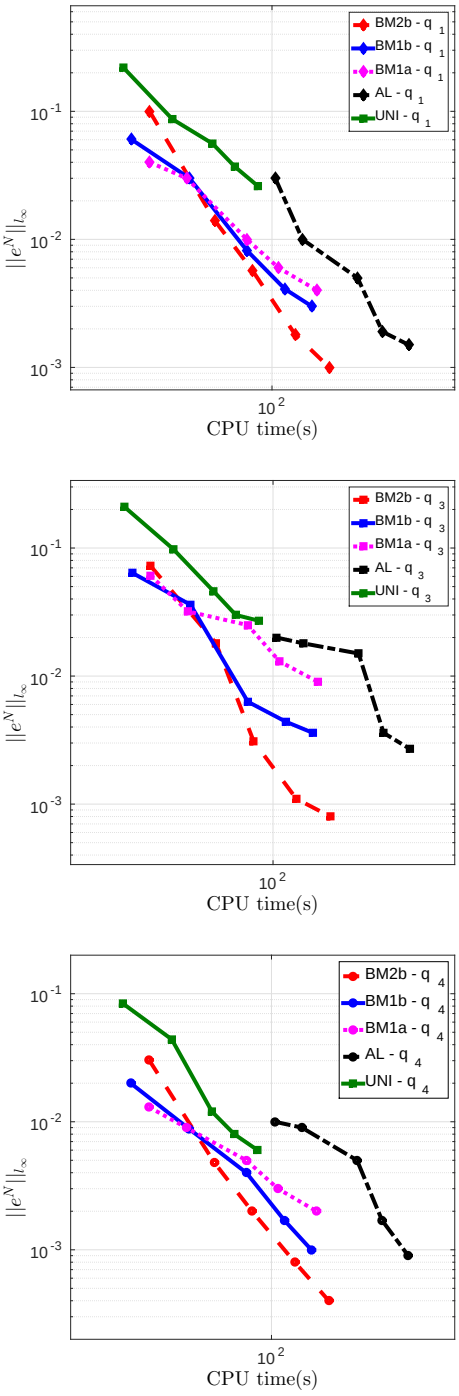

Figure 6.9: The l_∞ error in q_1, q_3 and q_4 plotted against the total CPU time in seconds for each method, measured at time $t^* = 0.2$ ms. The data points correspond to grids using 122, 162, 218, 286, and 342 quadratic triangular elements.

6.3 Perturbed two-dimensional π-cell problem

Having established a good choice of monitor function for resolving defects in two-dimensional problems, we turn our attention again to the π-cell problem previously considered, in one dimension, in Chapter 5. The two-dimensional setup of this problem is shown in Figure 6.10.

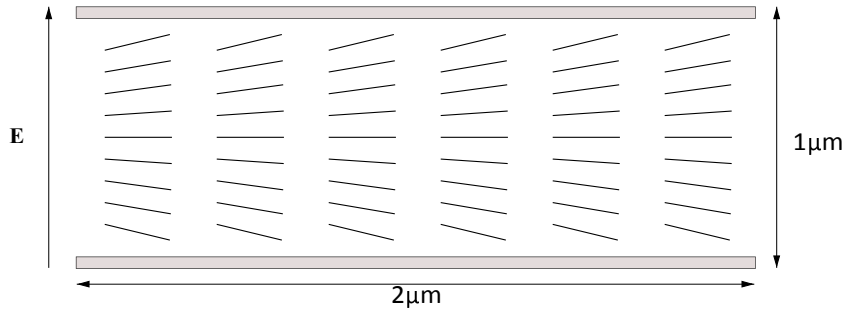

Figure 6.10: Unperturbed cell configuration in splay state under the influence of an electric field **E**.

Strong anchoring is again applied at the upper and lower cell boundaries; that is, we assume that the cell surface has been treated so as to induce a fixed pretilt angle at each boundary, with the director varying linearly throughout the cell. At the left and right boundaries we enforce periodic conditions by mapping the left boundary nodes to the right boundary nodes in the finite element mesh connectivity array. In this model the director angle will be exactly $\theta = 0^\circ$ through the centre of the cell. In reality, however, it is unrealistic for this to be achieved exactly due to small variations in the pretilt angles or thermal fluctuations [80].

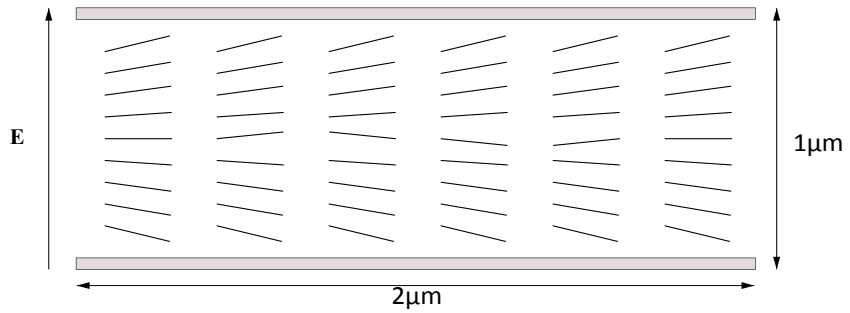

Figure 6.11: Perturbed cell configuration in splay state under the influence of an electric field **E**.

It is suggested in [80] that a more realistic problem would have the central layer of directors perturbed slightly so that a perfect $\theta = 0°$ degree angle is not achieved. Instead the angle across the centre of the cell is modified so that it follows the sinusoidal function $A\sin(2\pi x/p)$, where A is the amplitude ($A = 1°$ here), x is the spatial coordinate in the horizontal plane, and p is the cell width. A schematic of this setup is shown in Figure 6.11. This perturbation is fixed only at $t = 0$ for one time step. An electric field of strength $18V\mu\text{m}^{-1}$ is applied parallel to the cell thickness at time $t = t_{\text{on}} = 0$. In all experiments we measure time from the moment the electric field is applied to the cell. We consider a cell of width 2μm and thickness 1μm, with a pre tilt of $\theta = \pm 6°$. Based on the evidence in the previous chapter, we will apply the best choice of monitor function to this problem, i.e., the BM2b monitor function.

6.3.1 Numerical results

Initially, immediately before the application of the electric field, the cell is in an equilibrium state. At this point the order parameter and biaxiality take constant values of $S \approx 0.65$ and $b = 0$ respectively. The mesh at this stage is quasi-uniform as no adaption has yet taken place. As time evolves, the combined effect of the perturbation and the applied electric field become apparent.

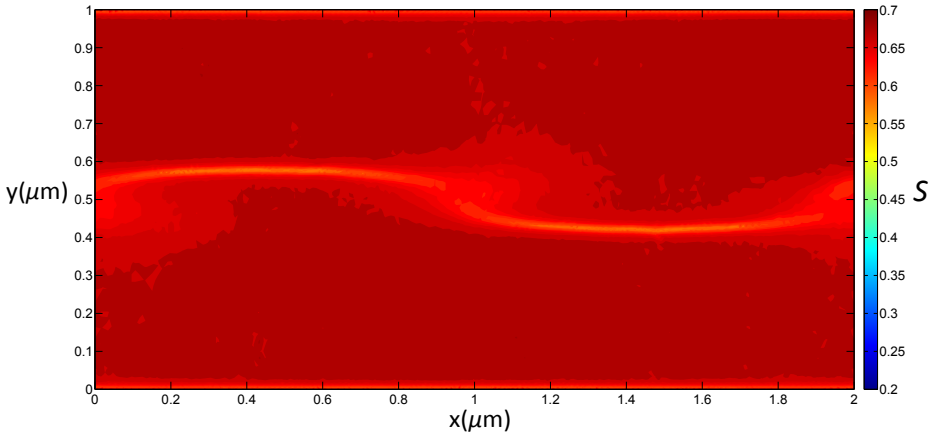

Figure 6.12: Order parameter after 12μs.

Figures 6.12 and 6.13 show the cell 12μs after the application of the electric field; at this time there is a region of concentrated splay distortion at the centre of the cell. We observe that in this area of high elastic distortion, the order parameter and biaxiality are no longer at their constant equilibrium values. With the measure of biaxiality now varying in time and space, and our choice of monitor function

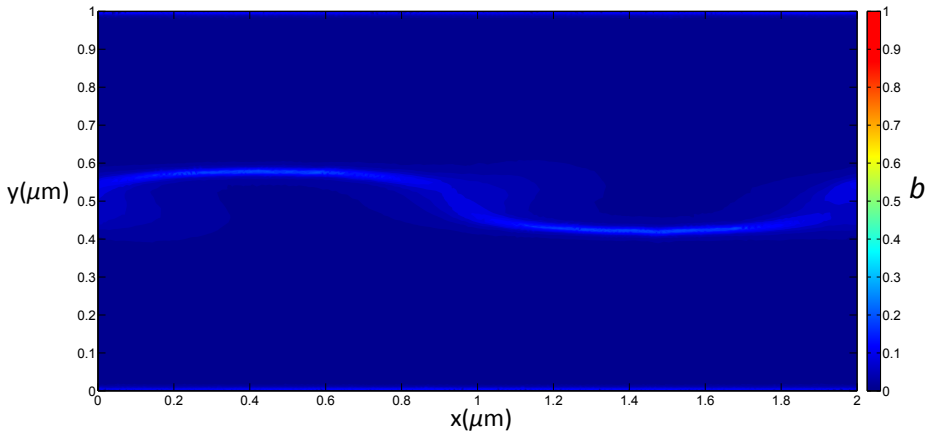

Figure 6.13: Biaxiality after 12μs.

being based on second derivatives of biaxiality, the mesh, as expected, has started to adapt to the area of distortion as depicted in Figure 6.14.

From [80] and the evolution of the biaxiality profile in Figure 6.18, we predict that pairs of +1/2 and -1/2 defects appear in the cell; these are depicted in Figure 6.15, which is plotted after 15.5μs, by which time the distortion at the centre of the cell has become more pronounced. Outside this area of distortion the cell is largely in an equilibrium state, with the order parameter and biaxiality still at their constant equilibrium values. However, the cores of the defects are now completely biaxial, and the measure of biaxiality approaches its maximal value of 1. With the large spatial variation in the biaxiality we observe in Figure 6.16 that the mesh has now adapted significantly from its quasi-uniform initial state, and has started to adapt well to resolve the defects. As time evolves further, the oppositely signed

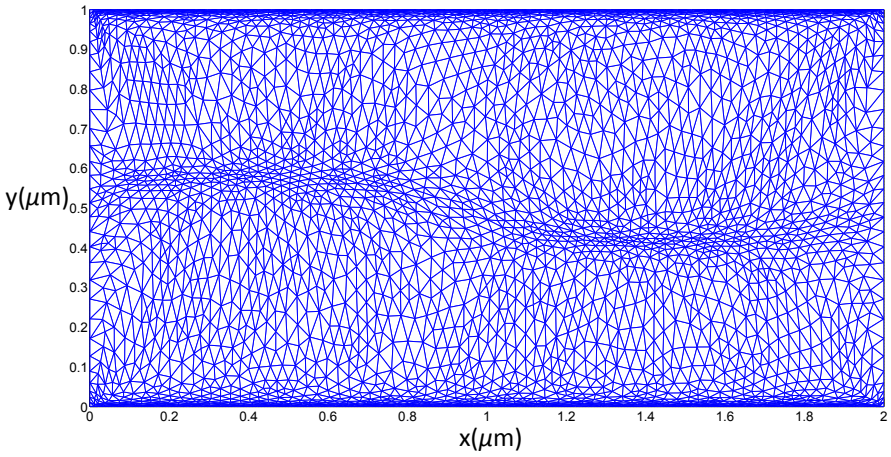

Figure 6.14: Adapted finite element mesh after 12μs.

defects are attracted to each other, moving ever closer until they ultimately meet and annihilate each other. Figures 6.17 and 6.18 show snapshots of the order parameter and biaxiality respectively, measured after 15.5μs, 16μs and 17μs. It is pleasing to note how good a job the mesh does of capturing the movement of the defects. In Figure 6.16, after 16μs, the mesh is still well adapted to the sinusodial shape of the initial perturbation, consistent with the presence of large variations in the biaxiality throughout the central area of the cell. However after 17μs, by which point the defects have almost coalesced, the mesh has relaxed in areas where the biaxiality is now back to its equilibrium value, and instead is completely focused on resolving the defects. After the defects meet and annihilate, the biaxiality and order parameter again relax towards their equilibrium value everywhere in the cell, and the mesh also relaxes back to a quasi-uniform state.

At the scale plotted in Figure 6.15, which is taken after 15.5μs, the predicted defects are not immediately obvious (although there is clearly some distortion across the center of the cell). We also plot the director profile at a slightly later time, after 19μs, by which point two of the defects have coalesced roughly at the cell centre, and the other two defects have moved towards the left and right boundaries. We plot the director profile at the left boundary, the cell center, and the right boundary in Figures 6.19, 6.20, and 6.21 respectively, and observe that there is clearly significant distortion of the director in each area.

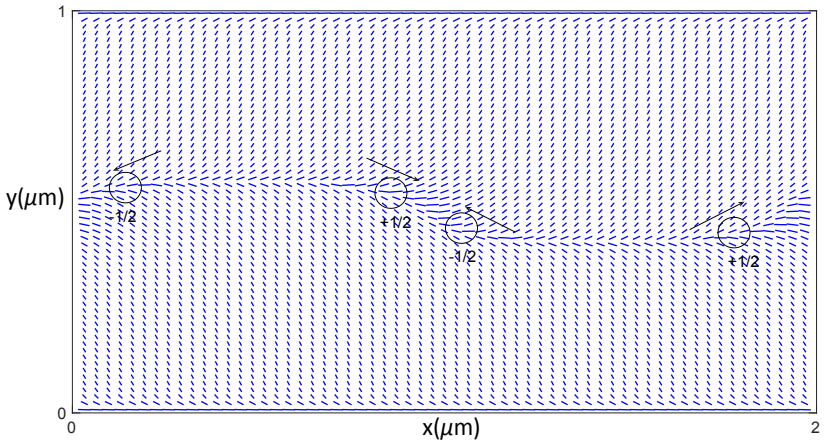

Figure 6.15: Director field after 15.5μs.

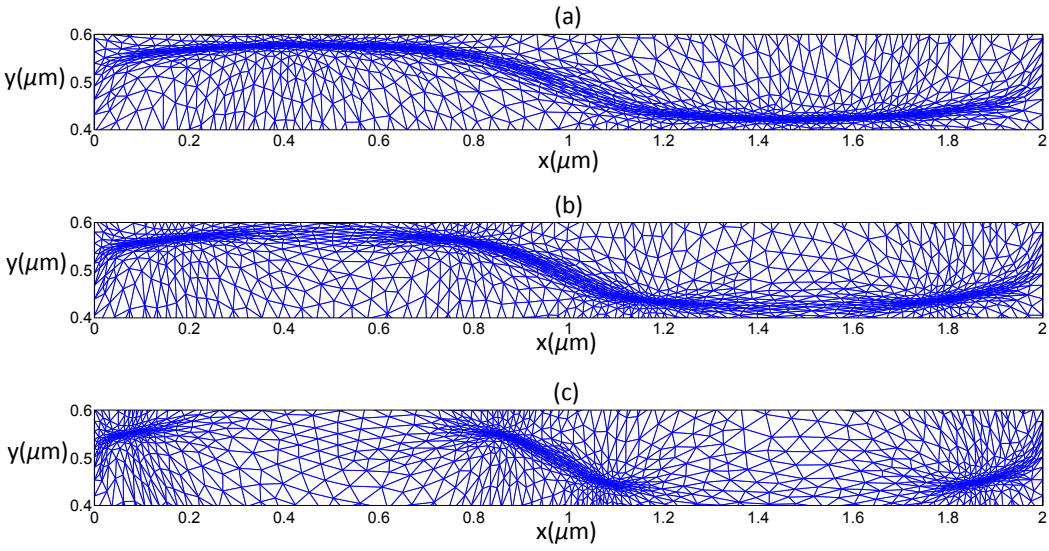

Figure 6.16: Adapted finite element mesh after (a) 15.5μs (b) 16μs and (c)17μs.

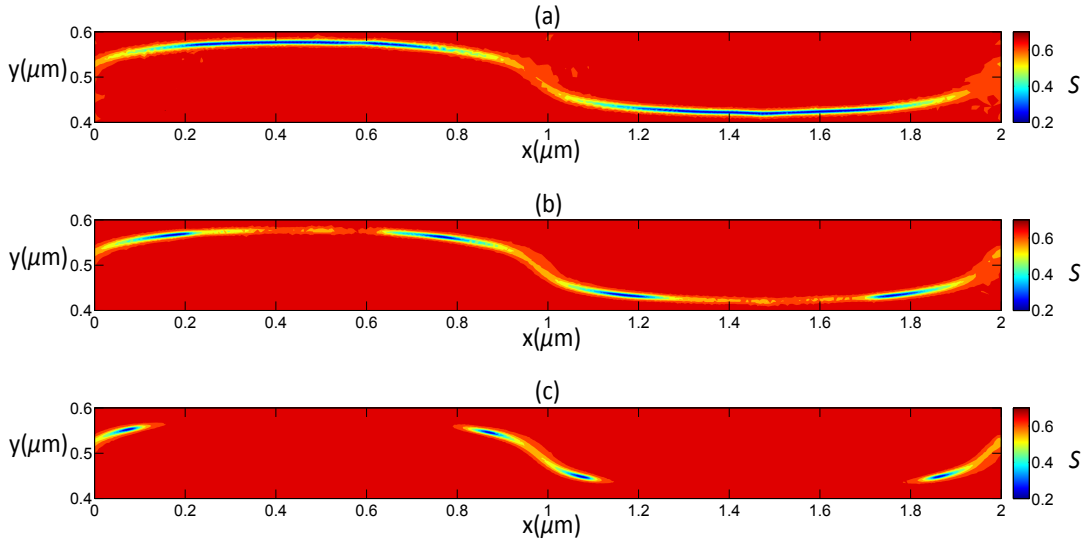

Figure 6.17: Order parameter profile after (a) 15.5μs (b) 16μs and (c)17μs

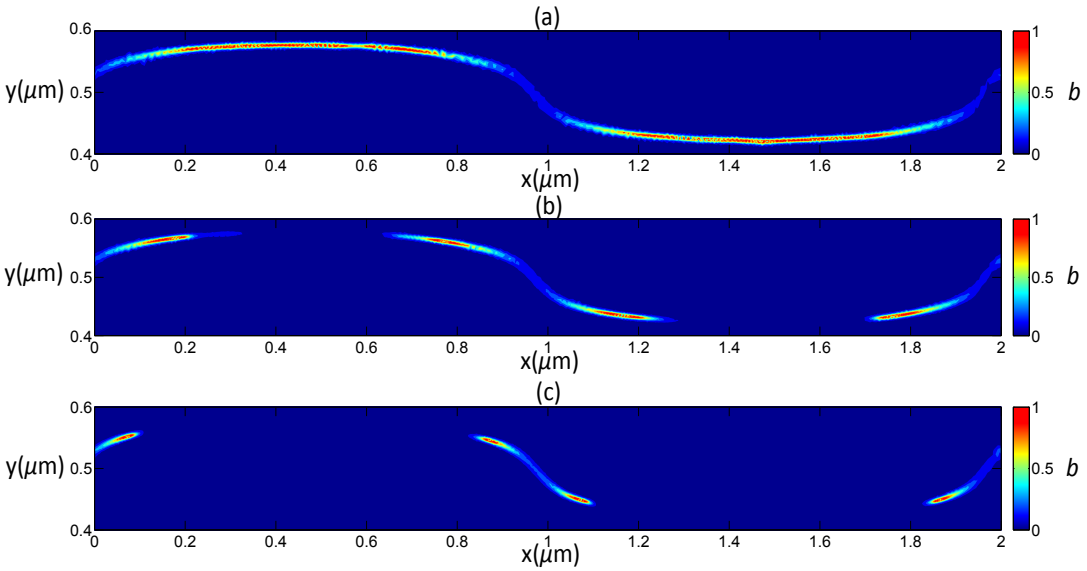

Figure 6.18: Biaxiality profile after (a) 15.5μs (b) 16μs and (c)17μs

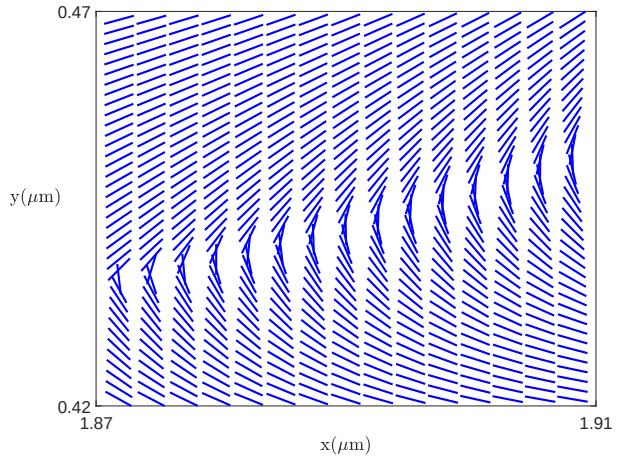

Figure 6.19: Director field at the left boundary after 19μs.

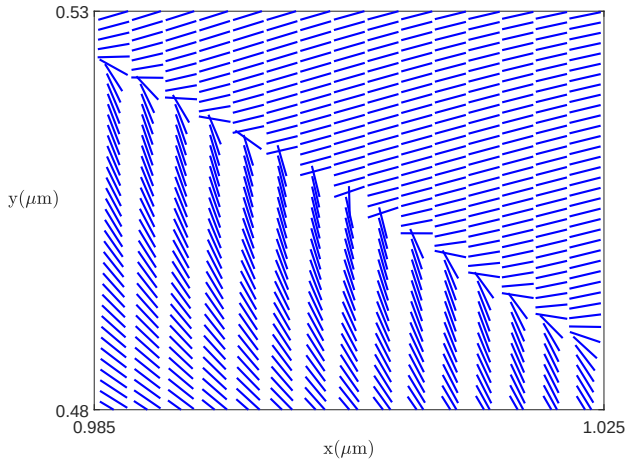

Figure 6.20: Director field in the cell center after 19μs.

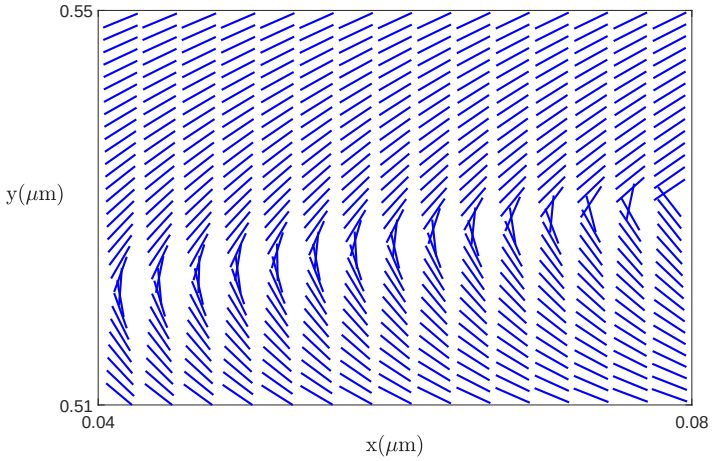

Figure 6.21: Director field at the right boundary after 19μs.

6.4 Conclusions

In this chapter an adaptive moving mesh method was developed to tackle two-dimensional problems modelled using $\boldsymbol{Q}$-tensor theory of liquid crystals. A stationary defect was considered as an ideal two-dimensional test problem for the adaptive moving mesh method. It was found that the adaptive method coped well with resolving the small scale structure of the defect core. A study of a number of different monitor functions was then performed, with the selection of monitor functions being the two-dimensional analogues to those considered in earlier chapters. A cost/benefit analysis was performed and it was found that the monitor function based on second derivatives of the biaxiality was computationally the most efficient. Having established the best choice of monitor function, a perturbed π-cell problem was then considered. It was found that the adaptive moving mesh

method did a particularly good job of tracking the development, movement, and annihilation of the defects that were present in the liquid crystal cell. The method was able to cope well with the small-scale structure of the defect core, and the short timescales associated with the establishment and annihilation of defects.

Chapter 7

Conclusions

The focus of this thesis has been on the development and application of an efficient moving mesh method to $\boldsymbol{Q}$-tensor liquid crystal models. The novelty contained within the thesis can be seen in three areas: an adaptive moving mesh method was applied to a one-dimensional uniaxial model problem; an adaptive moving mesh method was then applied to the biaxial order reconstruction π-cell problem in one-dimension; finally the one-dimensional work was extended to two dimensions and an adaptive moving mesh method was applied to a perturbed two-dimensional π-cell problem.

With regards to the one-dimensional uniaxial model problem, we considered the solution of a non-linear boundary value problem arising from a simplified $\mathbf{Q}$-tensor model of a nematic liquid crystal. The solution-adaptive mesh was obtained by equidistribution of the BM monitor function, which has previously been used

to solve linear reaction-diffusion problems. Numerical experiments were carried out which show that the computed errors are robust to variations in the size of the liquid crystal cell – a desirable property which is not realised when a uniform grid or an adaptive mesh based on equidistribution of the AL monitor function is used. An iterative algorithm was used to find approximately equidistributed grids, and to obtain solutions to a given degree of accuracy. It was shown that the use of the BM monitor function can result in over a thousand-fold decrease in CPU time compared to the use of the AL monitor function.

Motivated by the positive results obtained when applying the BM monitor function to the uniaxial model problem, the next problem tackled was the physically more realistic, and challenging, biaxial order reconstruction π-cell problem. An MMPDE approach was been used to generate the moving mesh where the equations had been discretised using second-order finite differences in space and first-order backward Euler time integration. To capture the highly non-linear nature of the $\boldsymbol{Q}$-tensor equations, a conservative finite element discretisation using quadratic elements was used to update the solution on the adaptive moving mesh. Time integration of the $\boldsymbol{Q}$-tensor equations was achieved using a second-order semi-implicit Runge-Kutta scheme and adaptive time-step control. These components were put together to form an adaptive algorithm that has been carefully tested and the computed solutions have been shown to converge at optimal rates in both space and time. These experiments confirm our previous findings for the

uniaxial model problem, namely, that it is not necessary to approximate the MM-PDE equation with the same spatial or temporal degree of accuracy as that used to discretise the governing PDEs to ensure optimal rates of convergence. Evidence has also been presented to suggest that the computed solutions exibit nodal super-convergence, which is somewhat surprising given the highly non-uniform nature of the adaptive moving meshes. For the first time, a monitor function has been constructed based upon a local measure of biaxiality. This has been shown to lead to higher levels of solution accuracy and a considerable improvement in computational efficiency compared to those monitor functions used previously for liquid crystal problems.

Finally, this work was then extended to two dimensions, and an adaptive moving mesh method was developed to solve two-dimensional problems. A stationary $+1/2$ defect was considered to explore the ability of a variety of monitor functions, analogous to those considered in earlier one-dimensional work, to resolve the small scale structure of the defect core. A cost/benefit analysis was performed and it was found that the monitor function based on second derivatives of the biaxiality was computationally the most efficient. As was the case with the one-dimensional work, this was the first time such a monitor function has been considered for two-dimensional liquid crystal problems. Having established the best choice of monitor function, a perturbed π-cell problem was then considered. It was found that the adaptive moving mesh method did a particularly good job of tracking the devel-

opment, movement, and annihilation of the defects that were present in the liquid crystal cell. The method was able to cope well with the small-scale structure of the defect core, and the short timescales associated with the establishment and annihilation of defects.

Future work in the area could follow a number of directions. First there are options in terms of exploring multi-dimensional problems with irregular geometries. This would pose a further challenge to the adaptive moving mesh method as it would potentially have to resolve defects present around the areas where the cell geometry is most complex. A prime candidate would be a Zenithally Bistable Device (ZBD) such as the one presented in [53], where the liquid crystal cell has an alignment layer on the upper surface and a periodic grating structure on the lower surface. Furthermore, there is plenty of scope for extending the adaptive moving mesh method to solve liquid crystal problems in three dimensions. Any such work would benefit from the MMPDE approach and the conservative finite element discretisation of the $\boldsymbol{Q}$-tensor equations extending naturally to higher dimensions [15]. Challenges would be continuing to identify the correct adaptivity criteria for problems with moving singularities such as defects, and the efficient solution of the large systems of highly non-linear algebraic equations arising after discretisation.

Appendix A

Governing equations

The full three dimensional governing equations can be derived from (2.18) and the energy terms described in §2.2. We have, as in §2.5,

$$\nu \frac{\partial \mathcal{D}}{\partial \dot{q}_i} = \nabla \cdot \hat{\mathbf{\Gamma}}_i - \hat{f}_i \qquad i = 1, \ldots, 5, \tag{A.1}$$

where the vector $\mathbf{\Gamma}_i$ has entries

$$\mathbf{\Gamma}_{i,j} = \frac{\partial \mathcal{F}_b}{\partial q_{i,j}}, \qquad q_{i,j} = \frac{\partial q_i}{\partial x_j}, \qquad j = 1, 2, 3,$$

$\hat{f}_i$ is given by

$$\hat{f}_i = \frac{\partial \mathcal{F}_b}{\partial q_i}.$$

A.1 1D governing equations: z-direction

In one dimension the components f_i and $\mathbf{\Gamma}_{i,j}$ in the z-direction are:

$$
\begin{aligned}
f_1 &= -\sqrt{6}B_s q_5^2 - 2\sqrt{6}B_s q_1 q_4 - \sqrt{6}B_s q_4^2 - 2C_s q_2^2 q_4 - 2C_s q_5^2 q_4 - 2C_s q_3^2 q_4 + \sqrt{6}B_s q_2^2 \\
&\quad -2A_s q_1 - 4C_s q_1^3 - 6C_s q_1^2 q_4 - 4C_s q_1 q_2^2 - 6C_s q_1 q_4^2 - 4C_s q_1 q_3^2 \\
&\quad -4C_s q_1 q_5^2 - A_s q_4 - 2C_s q_4^3 - U_z^2 \Delta\epsilon^*/\sqrt{6} \\
f_2 &= 2\sqrt{6}B_s q_2 q_4 - 4C_s q_2 q_1 q_4 + 2\sqrt{6}B_s q_1 q_2 - 4C_s q_2 q_4^2 + 2\sqrt{6}B_s q_3 q_5 - 2A_s q_2 - 4C_s q_2^3 \\
&\quad -4C_s q_1^2 q_2 - 4C_s q_2 q_5^2 - 4C_s q_2 q_3^2 \\
f_3 &= -2\sqrt{6}B_s q_3 q_4 - 4C_s q_3 q_1 q_4 + 2\sqrt{6}B_s q_5 q_2 - 4C_s q_3 q_4^2 - 2A_s q_3 - 4C_s q_3^3 \\
&\quad -4C_s q_1^2 q_3 - 4C_s q_3 q_2^2 - 4C_s q_3 q_5^2 \\
f_4 &= \sqrt{6}B_s q_2^2 - 2\sqrt{6}B_s q_1 q_4 - \sqrt{6}B_s q_3^2 - 2C_s q_2^2 q_1 - 2C_s q_5^2 q_1 - 2C_s q_3^2 q_1 - \sqrt{6}B_s q_1^2 \\
&\quad -2A_s q_4 - 4C_s q_4^3 - 6C_s q_1^2 q_4 - 4C_s q_4 q_2^2 - 6C_s q_1 q_4^2 - 4C_s q_4 q_3^2 \\
&\quad -4C_s q_4 q_5^2 - A_s q_1 - 2C_s q_1^3 - U_z^2 \Delta\epsilon^*/\sqrt{6} \\
f_5 &= -2\sqrt{6}B_s q_5 q_1 - 4C_s q_5 q_1 q_4 + 2\sqrt{6}B_s q_3 q_2 - 4C_s q_5 q_4^2 - 2A_s q_5 - 4C_s q_5^3 \\
&\quad -4C_s q_1^2 q_5 - 4C_s q_5 q_2^2 - 4C_s q_5 q_3^2.
\end{aligned}
$$

$$\begin{aligned}
\Gamma_{1,3} &= -(L_{1s}+1)q_{1,z} - L_{1s}q_{4,z} + U_z\bar{e}, \\
\Gamma_{2,3} &= -2q_{2,z}, \\
\Gamma_{3,3} &= -(L_{1s}+1)q_{3,z}, \\
\Gamma_{4,3} &= -(L_{1s}+1)q_{4,z} - L_{1s}q_{1,z} + U_z\bar{e}, \\
\Gamma_{5,3} &= -(L_{1s}+1)q_{5,z}.
\end{aligned}$$

The component of the displacement field that is used in Maxwell's equation $\nabla\cdot\boldsymbol{D} = 0$ are:

$$D_3 = \frac{\sqrt{6}}{3}U_z\Delta\epsilon^*(q_1+q_4) - \bar{\epsilon}U_z - \bar{e}(q_{1,z}+q_{4,z}).$$

A.2 2D governing equations: xy-plane

In two dimensions the components f_i and $\mathbf{\Gamma}_{i,j}$ in the xy-plane are:

$$
\begin{aligned}
f_1 &= -\sqrt{6}B_s q_5^2 - 2\sqrt{6}B_s q_1 q_4 - \sqrt{6}B_s q_4^2 - 2C_s q_2^2 q_4 - 2C_s q_5^2 q_4 - 2C_s q_3^2 q_4 + \sqrt{6}B_s q_2^2 \\
&\quad -2A_s q_1 - 4C_s q_1^3 - 6C_s q_1^2 q_4 - 4C_s q_1 q_2^2 - 6C_s q_1 q_4^2 - 4C_s q_1 q_3^2 \\
&\quad -4C_s q_1 q_5^2 - A_s q_4 - 2C_s q_4^3 - U_z^2 \Delta\epsilon^*/\sqrt{6} \\
f_2 &= 2\sqrt{6}B_s q_2 q_4 - 4C_s q_2 q_1 q_4 + 2\sqrt{6}B_s q_1 q_2 - 4C_s q_2 q_4^2 + 2\sqrt{6}B_s q_3 q_5 - 2A_s q_2 - 4C_s q_2^3 \\
&\quad -4C_s q_1^2 q_2 - 4C_s q_2 q_5^2 - 4C_s q_2 q_3^2 \\
f_3 &= -2\sqrt{6}B_s q_3 q_4 - 4C_s q_3 q_1 q_4 + 2\sqrt{6}B_s q_5 q_2 - 4C_s q_3 q_4^2 - 2A_s q_3 - 4C_s q_3^3 \\
&\quad -4C_s q_1^2 q_3 - 4C_s q_3 q_2^2 - 4C_s q_3 q_5^2 \\
f_4 &= \sqrt{6}B_s q_2^2 - 2\sqrt{6}B_s q_1 q_4 - \sqrt{6}B_s q_3^2 - 2C_s q_2^2 q_1 - 2C_s q_5^2 q_1 - 2C_s q_3^2 q_1 - \sqrt{6}B_s q_1^2 \\
&\quad -2A_s q_4 - 4C_s q_4^3 - 6C_s q_1^2 q_4 - 4C_s q_4 q_2^2 - 6C_s q_1 q_4^2 - 4C_s q_4 q_3^2 \\
&\quad -4C_s q_4 q_5^2 - A_s q_1 - 2C_s q_1^3 - U_z^2 \Delta\epsilon^*/\sqrt{6} \\
f_5 &= -2\sqrt{6}B_s q_5 q_1 - 4C_s q_5 q_1 q_4 + 2\sqrt{6}B_s q_3 q_2 - 4C_s q_5 q_4^2 - 2A_s q_5 - 4C_s q_5^3 \\
&\quad -4C_s q_1^2 q_5 - 4C_s q_5 q_2^2 - 4C_s q_5 q_3^2.
\end{aligned}
$$

$$
\begin{aligned}
\Gamma_{1,1} &= -(L_{1s}+1)q_{1,x} - q_{4,x} - U_x\bar{e} - L_{1s}q_{2,y}, \\
\Gamma_{2,1} &= -(L_{1s}+1)q_{2,x} + q_{4,y} - U_y\bar{e} - L_{1s}q_{4,y}, \\
\Gamma_{3,1} &= -(L_{1s}+1)q_{3,x} - L_{1s}q_{5,y}, \\
\Gamma_{4,1} &= -2q_{4,x} - q_{1,x} + q_{2,y}, \\
\Gamma_{5,1} &= -2q_{5,x} + q_{3,y}. \\
\Gamma_{1,2} &= -2q_{1,y} - q_{4,y} + q_{2,x}, \\
\Gamma_{2,2} &= -(L_{1s}+1)q_{2,y} + q_{4,x} - U_x\bar{e} - L_{1s}q_{1,x}, \\
\Gamma_{3,2} &= -2q_{3,y} + q_{5,x}, \\
\Gamma_{4,2} &= -(L_{1s}+1)q_{4,y} - q_{1,y} - U_y\bar{e} - L_{1s}q_{2,x}, \\
\Gamma_{5,2} &= -(L_{1s}+1)q_{5,y} - L_{1s}q_{3,x}.
\end{aligned}
$$

The components of the displacement field that are used in Maxwell's equation $\nabla \cdot \boldsymbol{D} = 0$ are:

$$
\begin{aligned}
D_1 &= -\frac{\sqrt{6}}{3}\Delta\epsilon^*(U_x q_1 + U_y q_4) - \bar{\epsilon}U_x + \bar{e}(q_{1,x} + q_{2,y}), \\
D_2 &= -\frac{\sqrt{6}}{3}\Delta\epsilon^*(U_x q_2 + U_y q_4) - \bar{\epsilon}U_y + \bar{e}(q_{2,x} + q_{4,y}).
\end{aligned}
$$

Appendix B

Finite element basis functions

Many of the terms in the weak formulation presented in Chapter 3 cannot be integrated exactly and must be approximated using numerical integration; we employ a suitable Gaussian quadrature formula to achieve this. In order to use Gaussian quadrature to integrate a function over an arbitrary interval we map from the global Cartesian coordinates (Figure B.1a) to the canonical element $\xi \in [-1, 1]$ (Figure B.1b). In one-dimension the transformation is given by

$$x(\xi) = \frac{1}{2}(1-\xi)x_{j-1} + \frac{1}{2}(1+\xi)x_j, \tag{B.1}$$

where we are mapping from the element $e = [x_{j-1}, x_j]$. The local linear shape functions we consider are

$$N_1(\xi) = \frac{1}{2}(1-\xi), \tag{B.2}$$

$$N_2(\xi) = \frac{1}{2}(1+\xi), \tag{B.3}$$

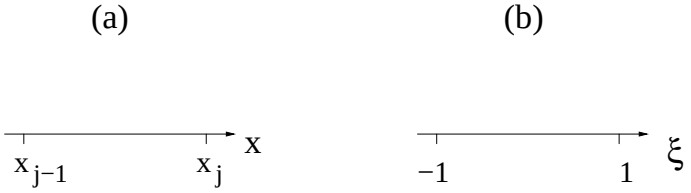

Figure B.1: (a) Global coordinates and (b) local coordinates for linear element in one dimension.

and on an element e the value u_h is approximated by

$$u_h = U_{j-1}N_1(\xi) + U_j N_2(\xi). \tag{B.4}$$

These shape functions N_i, $i = 1, 2$ have the the key property that they take unit value at node i, zero value at every other node, and vary linearly in an element. The local quadratic shape functions we consider are constructed by adding an extra degree of freedom at the cell centre. They are given by

$$N_1(\xi) = \frac{1}{2}(1-\xi), \tag{B.5}$$

$$N_2(\xi) = 1-\xi^2, \tag{B.6}$$

$$N_3(\xi) = \frac{1}{2}(1+\xi), \tag{B.7}$$

and on an element e the value u_h is approximated by

$$u_h = U_{j-1}N_1(\xi) + U_{j-1/2}N_2(\xi) + U_j N_3(\xi). \tag{B.8}$$

The quadratic functions N_i, $i = 1, 2, 3$ have the property that they take unit value at node i, zero value at every other node, and vary quadratically in an element. The two sets of functions are shown pictorially in Figure B.2.

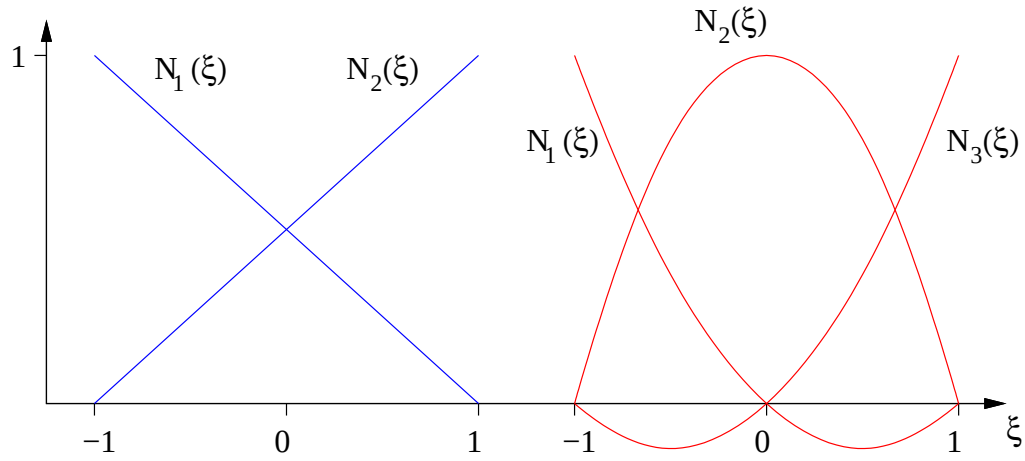

Figure B.2: Piecewise linear and quadratic local element shape functions.

In two dimensions we must again choose a suitable discretisation for our domain. A standard method is to discretise the domain using triangles such that the entire domain is covered exactly (assuming a polygonal domain). That is, a partition $\mathcal{T}_h$ of the domain Ω_h into triangular elements is admissable if and only if a set of triangles covers Ω_h exactly, and that pairs of triangles intersect only at a common vertex or along a complete edge. Pictorially such a mesh may look like the one presented in Figure B.3. We again wish to use numerical quadrature to integrate terms in our finite element formulation so construct local shape functions under a suitable mapping from global Cartesian coordinates to a local canonical triangle. There are a number of possible choices for a canonical triangle; we consider that which is shown in Figure B.4b. The transformation from global to local coordinates is given by

$$\boldsymbol{r} = (1 - \xi - \eta)\boldsymbol{r}_1 + \xi\boldsymbol{r}_2 + \xi\boldsymbol{r}_3, \tag{B.9}$$

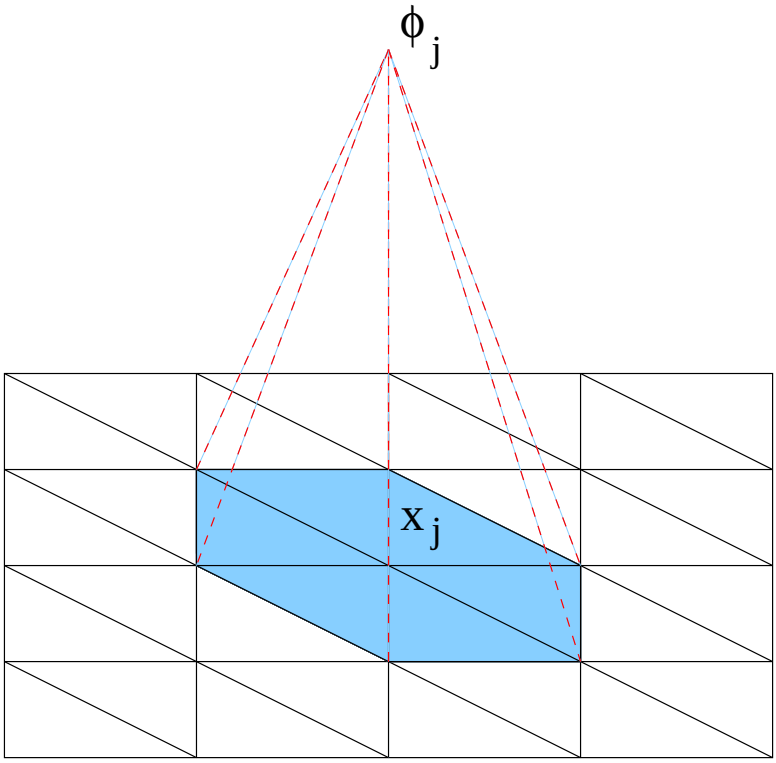

Figure B.3: A combination of linear basis functions giving a piecewise linear solution.

where $\boldsymbol{r}_i = (x_i, y_i), \quad i = 1, 2, 3$ are the local triangle vertices and the mapping defines the local (linear) shape functions

$$N_1(\xi, \eta) = 1 - \xi - \eta, \tag{B.10}$$

$$N_2(\xi, \eta) = \xi, \tag{B.11}$$

$$N_3(\xi, \eta) = \eta. \tag{B.12}$$

Each shape function N_i, $i = 1, \ldots, 3$ is linear in form, takes unit value at node i and is zero at all other nodes. We will, however, restrict our attention in this thesis to quadratic basis functions on triangular elements. The justification for

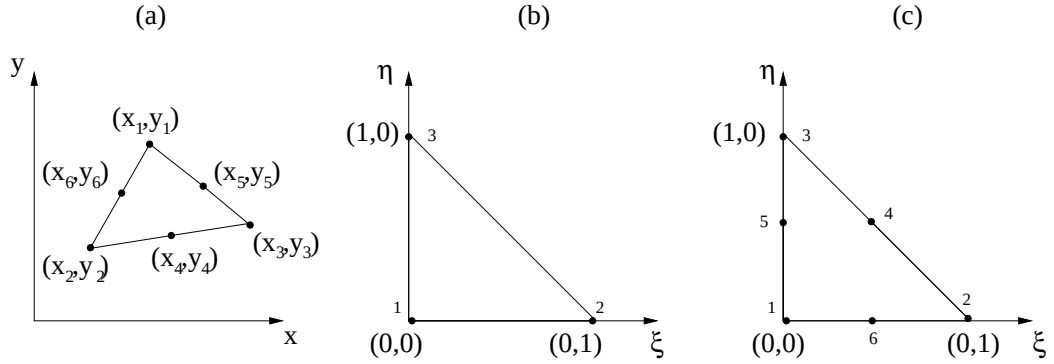

Figure B.4: (a) Global coordinates,(b) local coordinate for linear triangular elements, and (c) local coordinates for quadratic triangular elements.

this is based on our work in Chapter 4 where we show quadratic functions to be significantly better than linears. To ensure continuity between elements the quadratic variation along the common edge is shared, i.e., all three nodal values on an edge are common nodal values. The quadratic triangle is therefore constructed by the addition of degrees of freedom at the mid-point of a triangle's edge, such that each triangle now has six degrees of freedom as opposed to three, as in Figure B.4c. It is sufficient to use the same linear transformation (B.9) from global to local coordinates to construct the quadratic shape functions. Each shape function N_i, $i = 1, \ldots, 6$ must be quadratic in form, take unit value at node i and be zero

at all other nodes. We thus arrive at the set of shape functions

$$N_1(\xi,\eta) = (1-\xi-\eta)(1-2\xi-2\eta), \tag{B.13}$$

$$N_2(\xi,\eta) = \xi(2\xi-1), \tag{B.14}$$

$$N_3(\xi,\eta) = \eta(2\eta-1), \tag{B.15}$$

$$N_4(\xi,\eta) = 4\xi\eta, \tag{B.16}$$

$$N_5(\xi,\eta) = 4\eta(1-\xi-\eta), \tag{B.17}$$

$$N_6(\xi,\eta) = 4\xi(1-\xi-\eta), \tag{B.18}$$

which are graphically represented in Figure B.5. Locally any quadratic function v is approximated by

$$v(\boldsymbol{r}(\xi,\eta)) = \sum_{i=1}^{6} V_i N_i(\xi,\eta), \tag{B.19}$$

where $V_i = V(\boldsymbol{r}_i)$, $i = 1,\ldots,6$.

In solving our system of equations we are required to evaluate spatial derivative terms. To achieve this in terms of our local coordinates, we require the Jacobian of the coordinate transformation (B.9) between global and local coordinates. The Jacobian matrix is

$$J = \frac{\partial(x,y)}{\partial(x,y)} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix}. \tag{B.20}$$

Applying the chain rule to the local element shape functions we have

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} \\ \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \eta} \end{bmatrix} = J \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix}, \tag{B.21}$$

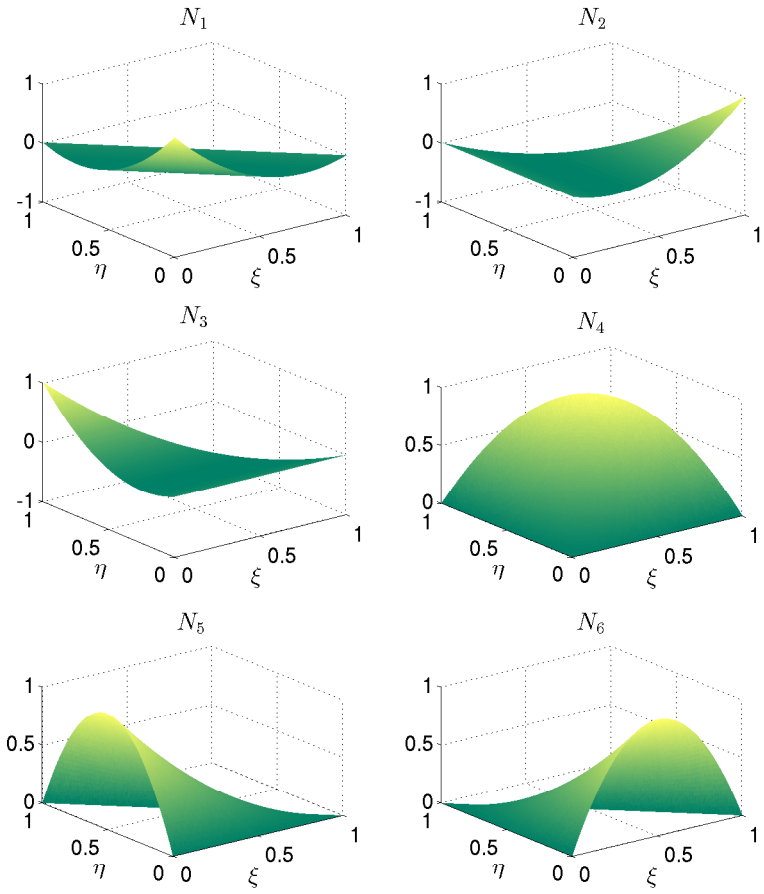

Figure B.5: Graphical representation of the local quadratic shape functions.

and hence

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{1}{|J|}\left[(y_3 - y_1)\frac{\partial N_i}{\partial \xi} - (y_2 - y_1)\frac{\partial N_i}{\partial \eta}\right] \\ \\ \frac{1}{|J|}\left[-(x_3 - y_1)\frac{\partial N_i}{\partial \xi} - (x_2 - x_1)\frac{\partial N_i}{\partial \eta}\right] \end{bmatrix}, \tag{B.22}$$

where

$$|J| = \det \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} = 2\mathrm{A}_{123}, \tag{B.23}$$

and A_{123} is the area of the triangle with vertices $\boldsymbol{r}_1, \boldsymbol{r}_2$ and $\boldsymbol{r}_3$.

Bibliography

[1] A. Amoddeo. Concurrence of bulk and surface order reconstruction to the relaxation of frustrated nematics. In *Journal of Physics: Conference Series*, volume 738, pages 12–89. IOP Publishing, 2016.

[2] A. Amoddeo, R. Barberi, and G. Lombardo. Moving mesh partial differential equations to describe nematic order dynamics. *Comput. Math. Appl.*, 60:2239–2252, 2010.

[3] A. Amoddeo, R. Barberi, and G. Lombardo. Electric field-induced fast nematic order dynamics. *Liq. Cryst.*, 38(1):93–103, 2011.

[4] A. Amoddeo, R. Barberi, and G. Lombardo. Nematic order and phase transition dynamics under intense electric fields. *Liq. Cryst.*, 40(6):799–809, 2013.

[5] I. Babuska, J. E. Flaherty, W. D. Henshaw, J. E. Hopcroft, J. E. Oliger, and T. Tezduyar. *Modeling, mesh generation, and adaptive numerical methods for*

partial differential equations, volume 75. Springer Science & Business Media, 2012.

[6] I. Babuska, A. Miller, and M. Vogelius. Adaptive methods and error estimation for elliptic problems of structural mechanics. Technical report, DTIC Document, 1983.

[7] M. J. Baines, M. E. Hubbard, and P. K. Jimack. Velocity-based moving mesh methods for nonlinear partial differential equations. *Commun. Comput. Phys.*, 10(3):509–576, 2011.

[8] V. M. Balasubramaniam and S. K. Sastry. Use of liquid crystals as temperature sensors in food processing research. *J. Food Eng.*, 26(2):219–230, 1995.

[9] R. Barberi, F. Ciuchi, G. E. Durand M. Iovane, D. Sikharulidze, A. M. Sonnet, and E. G. Virga. Electric field induced order reconstruction in a nematic cell. *Eur. Phys. J. E*, 13:61–71, 2004.

[10] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*. SIAM, 1994.

[11] G. Beckett and J. A. Mackenzie. Convergence analysis of finite-difference approximations on equidistributed grids to a singularly perturbed boundary value problem. *Appl. Numer. Math.*, 35:87–109, 2000.

[12] G. Beckett and J. A. Mackenzie. On a uniformly accurate finite difference approximation of a singularly perturbed reaction-diffusion problem using grid equidistribution. *J. Comp. Appl. Math.*, 131:381–405, 2001.

[13] G. Beckett and J. A. Mackenzie. Uniformly convergent high order finite element solutions of a singularly perturbed reaction-diffusion equation using mesh equidistribution. *Appl. Numer. Math.*, 39:31–45, 2001.

[14] G. Beckett, J. A. Mackenzie, A. Ramage, and D. M. Sloan. On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution. *J. Comput. Phys.*, 167:372–392, 2001.

[15] G. Beckett, J. A. Mackenzie, A. Ramage, and D. M. Sloan. Computational solution of two-dimensional unsteady PDEs using moving mesh methods. *J. Comput. Phys.*, 182:478–495, 2002.

[16] M. W. Bern, J. E. Flaherty, and M. Luskin. *Grid generation and adaptive algorithms*, volume 113. Springer Science & Business Media, 1999.

[17] J. G. Blom, J. M. Sanz-Serna, and J. G. Verwer. On simple moving grid methods for one-dimensional evolutionary partial differential equations. *J. Comput. Phys.*, 74:191–213, 1988.

[18] G. D. Boyd, J. Cheng, and P. D. T. Ngo. Liquid-crystal orientational bistability and nematic storage effects. *Appl. Phys. Lett.*, 36:556–558, 1980.

[19] P. A. Browne, C. J. Budd, C. Piccolo, and M. Cullen. Fast three dimensional r-adaptive mesh redistribution. *J. Comput. Phys.*, 275:174–196, 2014.

[20] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numer.*, 18:111–241, 2009.

[21] N. M. Chadha and N. Kopteva. A robust grid equidistribution method for a one-dimensional singularly perturbed semilinear reaction-diffusion problem. *IMA J. Numer. Anal.*, 31:188–211, 2011.

[22] W. C. Cheng, Y. Hou, and M. Pedram. Power minimization in a backlit TFT-LCD display by concurrent brightness and contrast scaling. In *Proceedings of the conference on Design, automation and test in Europe-Volume 1*, page 10252. IEEE Computer Society, 2004.

[23] S. Cornford and C. J. P. Newton. An adaptive hierarchical finite element method for modelling liquid crystal devices. Technical Report HPL-2011-143R1, Hewlett-Packard Laboratories, 2011.

[24] D. Dabiri. Digital particle image thermometry/velocimetry: a review. *Exp. Fluids*, 46(2):191–241, 2009.

[25] C. de Boor. Good approximation by splines with variables knots II. In *Lecture Notes in Mathematics*, volume 363, pages 12–20. Springer-Verlag, 1974.

[26] P. G. de Gennes. Short range order effects in the isotropic phase of nematics and cholesterics. *Mol. Cryst. Liq. Cryst.*, 129:193–214, 1971.

[27] P. G. de Gennes. An analogy between superconductors and smectics A. *Solid State Commun.*, 10:753–756, 1972.

[28] P. G. de Gennes and J. Prost. *The Physics of Liquid Crystals*. Clarendon Press, Oxford, 2. edition, 1993.

[29] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy. Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure. *Comput. Methods Appl. Mech. Eng.*, 77(12):79 – 112, 1989.

[30] J. Douglas and T. Dupont. Galerkin approximations for two point boundary value problem using continuous, piece-wise polynomial spaces. *Numer. Math.*, 22:99–109, 1974.

[31] J. L. Fergason. Liquid crystals in nondestructive testing. *Appl. Opt.*, 7(9):1729–1737, 1968.

[32] G. Friedel. The mesomorphic states of matter. In *Crystals That Flow: Classic Papers from the History of Liquid Crystals*, pages 162–211. CRC Press, 2004.

[33] J. Fukuda and H. Yokoyama. Director configuration and dynamics of a nematic liquid crystal around a two-dimensional spherical particle: Numerical analysis using adaptive grids. *Eur. Phys. J. E*, 4:389–396, 2001.

[34] J. Fukuda, M. Yoneya, and H. Yokoyama. Defect structure of a nematic liquid crystal around a spherical particle: adaptive mesh refinement approach. *Phys. Rev. E*, 65(4), 2002.

[35] B. Guo and I. Babuska. The h-p version of the finite element method. *Comput. Mech.*, 1(1):21–41, 1986.

[36] B. Guo and I. Babuska. The h-p version of the finite element method. ii: General results and applications. *Comput. Mech.*, 1(3):203–220, 1986.

[37] B. Guo and I. Babuska. The h, p and h-p version of the finite element method: Basis theory and applications. *Adv. Eng. Softw.*, 15(3-4):159–174, 1992.

[38] W. Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *J. Comput. Phys.*, 171(2):753 – 775, 2001.

[39] W. Huang, Y. Ren, and R. D. Russell. Moving mesh methods based on moving mesh partial differential equations. *J. Comput. Phys.*, 113:279–290, 1994.

[40] W. Huang and R.D. Russell. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *SIAM J. Sci. Comput.*, 20:998–1015, 1999.

[41] W. Huang and R.D. Russell. *Adaptive Moving Mesh Methods*. Springer, New-York, 2011.

[42] Liquid Crystal Institute. Science and technology center for advanced liquid crystalline optical materials (alcom), 1993. National Science Foundation grant DMR 89-20147.

[43] P. T. Ireland and T. V. Jones. Liquid crystal measurements of heat transfer and surface shear stress. *Meas. Sci. Technol.*, 11(7):969, 2000.

[44] R. James, E. Willman, F. A. Fernandez, and S. E. Day. Finite-element modeling of liquid-crystal hydrodynamics with a variable degree of order. *IEEE Trans. Electron Devices*, 53(7):1575–1582, 2006.

[45] N. Kopteva and M. Stynes. A robust adaptive method for a quasi-linear one-dimensional convection-diffusion problem. *SIAM J. Numer. Anal.*, 39:1446–1467, 2001.

[46] N. Kopteva and M. Stynes. Numerical analysis of a singularly perturbed nonlinear reaction-diffusion problem with multiple solutions. *Comput. Math. Appl.*, 51:857–864, 2006.

[47] Shin-Nan Li. Automobile rearview mirror with lcd display, February 2007. US Patent 7,175,291.

[48] C. S. MacDonald, J. A. Mackenzie, A. Ramage, and C. J. P. Newton. Robust adaptive computation of a one-dimensional Q-tensor model of nematic liquid crystals. *Comput. Math. Appl.*, 64(11):3627–3640, 2012.

[49] C. S. MacDonald, J. A. Mackenzie, A. Ramage, and C. J. P. Newton. Efficient moving mesh methods for Q-tensor models of liquid crystals. *SIAM J. Sci. Comput.*, 37(2):B215–B238, 2015.

[50] R. Marlow, M. E. Hubbard, and P. K. Jimack. Moving mesh methods for solving parabolic partial differential equations. *Comput. Fluids*, 46(1):353–361, 2011.

[51] A. McRae, C. J. Cotter, and C. J. Budd. Optimal-transport-based mesh adaptivity on the plane and sphere using finite elements. *arXiv preprint arXiv:1612.08077*, 2016.

[52] H. Mori, E. C. Gartland, J. R. Kelly, and P. J. Bos. Multidimensional director modeling using the Q-tensor representation in a liquid crystal cell and its application to the Pi cell with patterned electrodes. *Jpn. J. Appl. Phys.*, 38:135–146, 1999.

[53] N.J. Mottram and C. J. P. Newton. Introduction to Q-tensor theory. Technical Report 10/04, University of Strathclyde, Department of Mathematics, 2004.

[54] L. S. Mulholland, Y. Qiu, and D. M. Sloan. Solution of evolutionary partial differential equations using adaptive finite differences with pseudospectral post-processing. *J. Comput. Phys.*, 131:280–298, 1997.

[55] J. T. Oden, L. Demkowicz, W. Rachowicz, and T. A. Westermann. Toward a universal h-p adaptive finite element strategy, part 2. A posteriori error estimation. *Comput. Methods Appl. Mech. Eng.*, 77(12):113 – 180, 1989.

[56] R. E. O'Malley. *Singular perturbation methods for ordinary differential equations*. Springer-Verlag, New-York, 1991.

[57] S. K. Krishnababu H. P. Hodson W. N. Dawes J. Hannis P. J. Newton, G. D. Lock and C. Whitney. Heat transfer and aerodynamics of turbine blade tips in a linear cascade. *J Turbomach.*, 128(2):300–309, 2006.

[58] R. Parker. Transient surface temperature response of liquid crystal films. *Mol. Cryst. Liq. Cryst.*, 20(2):99–106, 1973.

[59] P. Patricio, M. Tasinkevych, and M. M. Telo da Gama. Colloidal diopolar interaction in 2D smectic-C films. *Eur. Phys. J. E*, 7:117–122, 2002.

[60] Y. Qiu, D. M. Sloan, and T. Tang. Numerical solution of a singularly perturbed two-point boundary value problem using equidistribution: analysis of convergence. *J. Comput. Appl. Math.*, 116:121–143, 2000.

[61] A. Ramage and C. J. P. Newton. Adaptive solution of a one-dimensional order reconstruction problem in Q-tensor theory of liquid crystals. *Liq. Cryst.*, 34(4):479–487, 2007.

[62] A. Ramage and C. J. P. Newton. Adaptive grid methods for Q-tensor theory of liquid crystals: A one-dimensional feasibility study. *Mol. Cryst. Liq. Cryst.*, 480(1):160–181, 2008.

[63] D. C. Reda and D. P. Aeschliman. Liquid crystal coatings for surface shear-stress visualization in hypersonic flows. *J. Spacecr. Rockets*, 29(2):155–158, 1992.

[64] F. Reinitzer. Contributions to the knowledge of cholesterol. *Liq. Cryst.*, 5(1):7–18, 1989.

[65] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[66] J. E. Sargison, S. M. Guo, M. L. G. Oldfield, G. D. Lock, and A. J. Rawlinson. A converging slot-hole film-cooling geometry: Part 2 – transonic nozzle guide vane heat transfer and loss. In *ASME Turbo Expo 2001: Power for*

Land, Sea, and Air, pages V003T01A013–V003T01A013. American Society of Mechanical Engineers, 2001.

[67] T. J. Scheffer and J. Nehring. A new, highly multiplexable liquid crystal display. *Appl. Phys. Lett.*, 45(10):1021–1023, 1984.

[68] N. Schopohl and T. J. Sluckin. Hedgehog structure in nematic and magnetic systems. *J. Phys. France*, 49(7):1097–1101, July 1988.

[69] C. Schwab. *p-and hp-finite element methods: Theory and applications in solid and fluid mechanics*. Oxford University Press, 1998.

[70] N. M. Silvestre, P. Patricio, and M. M. Telo da Gama. Elliptical soft colloids in smectic-C films. *Phys. Rev. E*, 74(2):021706, 2006.

[71] C. A. Smith. A review of liquid crystal display technologies, electronic interconnection and failure analysis. *Circuit World*, 34(1):35–41, 2008.

[72] A. M. Sonnet, A. Kilian, and S. Hess. Alignment tensor versus director: Description of defects in nematic liquid crystals. *Phys. Rev. E*, 52(1):718–722, July 1995.

[73] I. W. Stewart. *The Static and Dynamic Continuum Theory of Liquid Crystals*. Taylor & Francis, London, 2004.

[74] B. A. Szabó. Mesh design for the p-version of the finite element method. *Comput. Methods Appl. Mech. Eng.*, 55(1-2):181–197, 1986.

[75] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. John Wiley & Sons Inc, 1996.

[76] H. Van Der Vorst. A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *Stat. Comput*, 13:631–644, 1992.

[77] G. Wagner, M. Kotulla, P. Ott, B. Weigand, and J. Von Wolfersdorf. The transient liquid crystal technique: influence of surface curvature and finite wall thickness. In *ASME Turbo Expo 2004: Power for Land, Sea, and Air*, pages 521–531. American Society of Mechanical Engineers, 2004.

[78] A. Winslow. Numerical solution of the quasi-linear Poisson equations in a nonuniform triangle mesh. *J. Comput. Phys.*, 1:149, 1967.

[79] F. Zhang, W. Huang, X. Li, and S. Zhang. Moving mesh finite element simulation for phase-field modeling of brittle fracture and convergence of Newton's iteration. *arXiv preprint arXiv:1706.05449*, 2017.

[80] Y. Zhang, D. B. Chung, B. Wang, and P. J. Bos. Alternative dynamics for the symmetric splay to bend transition in a nematic liquid crystal layer. *Liq. Cryst.*, 34(2):143–152, 2007.