View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Strathclyde Institutional Repository

Strathprints Institutional Repository

Ramage, A. and Newton, C.J.P. (2008) *Adaptive grid methods for Q-tensor theory of liquid crystals : a one-dimensional feasibility study.* Molecular Crystals and Liquid Crystals, 480 (1). pp. 160-181. ISSN 1542-1406

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (http://strathprints.strath.ac.uk/) and the content of this paper for research or study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to Strathprints administrator: mailto:strathprints@strath.ac.uk

http://strathprints.strath.ac.uk/

Ramage, A. and Newton, C.J.P. (2008) Adaptive grid methods for Q-tensor theory of liquid crystals: a one-dimensional feasibility study. Molecular Crystals and Liquid Crystals, 480 . pp. 160-181. ISSN 1542-1406

http://strathprints.strath.ac.uk/13298/

This is an author produced version of a paper published in Molecular Crystals and Liquid Crystals, 480 . pp. 160-181. ISSN 1542-1406. This version has been peer-reviewed but does not include the final publisher proof corrections, published layout or pagination.

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (http://strathprints.strath.ac.uk) and the content of this paper for research or study, educational, or not-for-profit purposes without prior permission or charge. You may freely distribute the url (http://strathprints.strath.ac.uk) of the Strathprints website.

Any correspondence concerning this service should be sent to The Strathprints Administrator: eprints@cis.strath.ac.uk

UNIVERSITY OF STRATHCLYDE
DEPARTMENT OF MATHEMATICS

Adaptive Grid Methods for Q-Tensor Theory of Liquid Crystals: A One-Dimensional Feasibility Study

by

Alison Ramage and Christopher J.P. Newton

Strathclyde Mathematics Research Report No. 13 (2006)
July 2006

Adaptive Grid Methods for **Q**-Tensor Theory of Liquid Crystals: A One-Dimensional Feasibility Study *

Alison Ramage† Christopher J.P. Newton‡

Abstract

This paper illustrates the use of moving mesh methods for solving partial differential equation (PDE) problems in Q-tensor theory of liquid crystals. We present the results of an initial study using a simple one-dimensional test problem which illustrates the feasibility of applying adaptive grid techniques in such situations. We describe how the grids are computed using an equidistribution principle, and investigate the comparative accuracy of adaptive and uniform grid strategies, both theoretically and via numerical examples.

Keywords

Nematic liquid crystals, order reconstruction, adaptive grids, moving meshes.

1 Introduction

The mathematical theory of liquid crystals has been extensively studied for over 75 years (for an introduction to many of the important ideas, see [1]). However, to date there has been much less work done on the numerous interesting and important numerical analysis issues which the study of such materials raises. Often, the underlying physical problems involve characteristic length and time scales which vary by many orders of magnitude, or complex combinations of fluid flow and changes in orientational order within a liquid crystal cell. Such features provide difficult numerical challenges to those trying to simulate the real-life dynamic situations which are of interest in an industrial setting. In this paper, we investigate the use of moving mesh solution techniques for a simple one-dimensional (1D) **Q**-tensor model of a nematic liquid crystal cell, with the aim of establishing the feasibility of these methods in the accurate representation of regions of high elastic distortion (*defects*) in novel liquid crystal display devices. Construction of grids that dynamically adapt to features of the computed solution is currently an area of intense research activity in numerical analysis, but such methods have not yet been used for applications in liquid crystals.

Most moving mesh methods can be classified into two groups: *velocity* based methods which calculate the mesh velocity directly (e.g. moving finite element methods [2], geometric conservation law methods [3]), and *location* based methods which directly control the mesh points themselves (e.g. equidistribution based methods [4, 5], harmonic mapping [6]). A comparison of these two classes is given in [7]. One key feature of velocity based schemes highlighted therein is the tendency for the calculated meshes to become increasingly skew, a problem which can be addressed by remeshing (although this may be expensive). We have therefore chosen to focus on the more established location based methods using equidistribution principles.

*This work was supported by EPSRC grant EP/C53154X/1.

†Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, Scotland. Tel: +441415483801, Fax: +441415483345, A.Ramage@strath.ac.uk

‡Hewlett-Packard Labs, Filton Road, Stoke Gifford, Bristol BS34 8QZ, England. Tel: +441173128727, Fax: +441173129937, chris.newton@hp.com

The structure of the paper is as follows. We begin with a brief description of the basic ideas involved in **Q**-tensor modelling, and introduce the 1D model problem to be studied in this paper. Section 3 contains an explanation of the concept of equidistribution and sets out the moving mesh algorithm used. In Section 4, the convergence rate and accuracy of uniform and adaptive grids is compared via a theoretical example, while Section 5 presents the results of numerical experiments which confirm that these results hold in practice for the 1D nematic liquid crystal cell studied.

2 A simple Q-tensor model in one dimension

2.1 Q-tensor theory

The most general configuration of molecules in a nematic material is a biaxial state. The standard director-based description of this state involves the specification of two directors ($\mathbf{n}$ and $\mathbf{m}$) and two scalar order parameters (S_1 and S_2). This results in a theory based on five degrees of freedom, namely, the Euler angles ϕ, θ, ψ and the order parameters S_1 and S_2 (see for example [8]). In **Q**-tensor theory, the same information is represented by the second rank tensor

$$\mathbf{Q} = S_1(\mathbf{n} \otimes \mathbf{n}) + S_2(\mathbf{m} \otimes \mathbf{m}) - \frac{1}{3}(S_1 + S_2)\mathbf{I}, \tag{1}$$

where $\mathbf{I}$ is the identity [9]. This tensor, which is symmetric and has zero trace (as $|\mathbf{n}| = |\mathbf{m}| = 1$), can be written in general as

$$\mathbf{Q} = \begin{bmatrix} q_1 & q_2 & q_3 \\ q_2 & q_4 & q_5 \\ q_3 & q_5 & -q_1 - q_4 \end{bmatrix} \tag{2}$$

involving the 5 quantities q_i, $i = 1, \ldots, 5$. A detailed description of this model is given in [8].

For static problems, 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 $\mathcal{F}$ of the liquid crystal sample. If it is assumed that distortions of $\mathbf{Q}$ are small, $\mathcal{F}$ can be taken to depend only on $\mathbf{Q}$ and its first derivatives, that is, we may write

$$\mathcal{F} = \int_V F_{bulk}(q_i, \nabla q_i)\, dv + \int_{\mathcal{S}} F_{surface}(q_i)\, ds.$$

In this paper, we apply fixed boundary conditions (strong anchoring), so the surface energy term can be ignored in the minimisation. Taking the elastic energy up to second order in the gradient of $\mathbf{Q}$, the bulk energy can be written as $\mathcal{F}_{bulk} = F_{thermotropic} + F_{elastic} \equiv F_t + F_e$, with

$$\begin{aligned} F_t &= \frac{1}{2}A(T - T^*)\,\mathrm{tr}\mathbf{Q}^2 - \frac{\sqrt{6}}{3}B\,\mathrm{tr}\mathbf{Q}^3 + \frac{1}{4}C(\mathrm{tr}\mathbf{Q}^2)^2, \\ F_e &= \frac{1}{2}L_1(\mathrm{div}\mathbf{Q})^2 + \frac{1}{2}L_2|\nabla \times \mathbf{Q}|^2, \end{aligned} \tag{3}$$

where A, B, C, L_1 and L_2 are positive constants, T represents temperature and T^* is the pseudo-critical temperature at which the isotropic phase becomes unstable (see [10]). The values for these material constants used here are

$$L_1 = 9.7 \times 10^{-12}\ \mathrm{N}, \qquad L_2 = 2.4 \times 10^{-12}\ \mathrm{N},$$
$$A = 0.13 \times 10^6\ \frac{J}{Km^3}, \quad B = 1.6 \times 10^6\ \frac{J}{Km^3}, \quad C = 3.9 \times 10^6\ \frac{J}{Km^3},$$

(taken from [11]).

For computational purposes, it is useful to non-dimensionalise the equations (3) with respect to the nematic coherence length $\zeta = \sqrt{9CL_2/(2B^2)} = 4.06 \times 10^{-9}$. Using a value of $T - T^* = -0.38°C$ results in the scaled parameter values (denoted by $_s$) $L_{1s} = 4.0417$, $A_s = -0.33682$, $B_s = 10.969$ and $C_s = 26.736$.

2.2 A uniaxial 1D model

For a uniaxial model, only one director and one order parameter are required, with associated $\mathbf{Q}$-tensor

$$\mathbf{Q} = \sqrt{\frac{3}{2}} S \left(\mathbf{n} \otimes \mathbf{n} - \frac{1}{3} I \right). \tag{4}$$

If in a 1D model of this type we take the z-axis in the direction of $\mathbf{n}$, (4) simplifies further to give

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

that is, $\mathbf{Q}$ depends only on the scalar order parameter S. The free energy density can be therefore be minimised (and hence the equilibrium configuration of S identified) by solving a single Euler-Lagrange equation for S, subject to suitable boundary conditions.

Here we study the variation of S in a sample with homogeneous uniaxial alignment in the one-dimensional domain $\Omega_s \equiv z \in [0, d_s]$. Note that this domain has been scaled in accordance with the above non-dimensionalisation, that is, the true length of the physical domain is d microns where $d = d_s \zeta$. For this model problem, the expressions for the energies in (3) simplify to

$$F_t = \frac{1}{2} A_s S^2 - \frac{1}{3} B_s S^3 + \frac{1}{4} C_s S^4, \qquad F_e = \left(\frac{2L_{1s} + 1}{6} \right) \left(\frac{\partial S}{\partial z} \right)^2. \tag{5}$$

The governing differential equation for S is therefore the Euler-Lagrange equation

$$\frac{\partial F}{\partial S} - \frac{d}{dz} \left(\frac{\partial F}{\partial S'} \right) = 0 \qquad \Rightarrow \qquad S'' = \frac{3A_s}{2L_{1s} + 1} S - \frac{3B_s}{2L_{1s} + 1} S^2 + \frac{3C_s}{2L_{1s} + 1} S^3.$$

Writing

$$\alpha = \frac{3A_s}{2L_{1s} + 1}, \quad \beta = \frac{3B_s}{2L_{1s} + 1}, \quad \gamma = \frac{3C_s}{2L_{1s} + 1},$$

this becomes

$$S'' = \alpha S - \beta S^2 + \gamma S^3. \tag{6}$$

We solve this subject to the boundary conditions $S = 0$ at $z = 0$ and $S = S_{eq}$ at $z = d_s$. 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}, which is the equilibrium nematic scalar order parameter. The value of S_{eq}, which minimises F_t in (5)) over S, is $S_{eq} = (B_s + \sqrt{B_s^2 - 4A_s C_s})/(2C_s) = 4.3896 \times 10^{-1}$.

We can obtain a numerical approximation to the solution of (6) as follows. Multiplying both sides of (6) by S' and integrating gives

$$S' S'' = S'(\alpha S - \beta S^2 + \gamma S^3) \qquad \Rightarrow \qquad S' = \sqrt{\alpha S^2 - \frac{2\beta}{3} S^3 + \frac{\gamma}{2} S^4 + 2c_1}$$

where c_1 is the constant of integration. To evaluate this constant, we apply the given boundary condition $S = S_{eq}$ at $z = d_s$ and make the additional assumption that $S' = 0$ at $z = d_s$. This will be the case if d_s is large enough, as S varies rapidly near $z = 0$ then quickly approaches the equilibrium value. With this assumption, we have

$$c_1 = -\frac{1}{2} \left(\alpha S_{eq}^2 - \frac{2\beta}{3} S_{eq}^3 + \frac{\gamma}{2} S_{eq}^4 \right)$$

so, writing

$$G(S) = \alpha S^2 - \frac{2\beta}{3} S^3 + \frac{\gamma}{2} S^4,$$

we obtain

$$S' = \sqrt{G(S) - G(S_{eq})}$$

and integration gives

$$\int_0^{S(z)} \frac{ds}{\sqrt{G(S) - G(S_{eq})}} = \int_0^z dz = z. \tag{7}$$

An approximation to the solution can therefore be obtained using a numerical evaluation of (7) for various values of S. These approximations for regions of length $d = 0.1, 1, 10$ microns are plotted using a solid line in Figures 1(b), (c) and (d). Note that, for reasons of clarity, these plots show $z \in [0, d/2]$ only. In each case, the solution features a boundary layer of width approximately 25 nanometres, which occupies a smaller proportion of the total cell length as d increases.

3 Moving mesh methods

3.1 Mappings and equidistribution

It is well known in the numerical analysis community that adaptive moving mesh methods are capable of resolving solutions with sharp transitions such as boundary layers to acceptable degrees of accuracy without using an excessive number of grid points. The idea is that instead of selectively refining and derefining the mesh, existing points are moved to regions of high error while maintaining the same grid connectivity.

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. The idea underlying many such mappings is an *equidistribution principle*, originally introduced by de Boor [12] for solving boundary value problems for ordinary differential equations. The aim is to select grid points in order to limit some measure of the solution error (defined using a *monitor function*) by distributing it equally across each subinterval. As an example, suppose we are solving on a physical domain $\Omega_p \equiv [0, 1]$ for an unknown function $u(z, t)$. To generate a grid, we look for a mapping from computational space $\Omega_c \times (0, T]$ say (where Ω_c is usually taken to be the same as Ω_p) to physical space $\Omega_p \times (0, T]$ such that, in the transformed variables, the variation of the selected monitor function is reduced. For example, if z and ξ denote the physical and computational coordinates, respectively, a one-to-one coordinate transformation between the domains is denoted by

$$\begin{aligned} z = z(\xi, t) \quad &, \qquad \xi \in \Omega_c = (0, 1), \qquad t \in (0, T] \\ z(0) = 0 \quad &, \qquad z(1) = 1. \end{aligned}$$

We impose a uniform mesh on the computational domain, given by

$$\xi_i = \frac{i}{N}, \qquad i = 0, 1, \ldots, N,$$

where N is a positive integer, and denote the corresponding mesh in Ω_p by

$$0 = z_0(t) < z_1(t) < \ldots < z_{N-1}(t) < z_N(t) = 1.$$

For a chosen monitor function $M(u(z, t))$, the one-dimensional *equidistribution principle* (EP) can be expressed as

$$\int_0^{z(\xi,t)} M(u(s, t))\, ds = \xi \int_0^1 M(u(z, t))\, dz. \tag{8}$$

This EP can be expressed equivalently in discrete form as

$$\int_{z_i(t)}^{z_{i+1}(t)} M(u(s, t))\, ds = \int_{z_{i-1}(t)}^{z_i(t)} M(u(s, t))\, ds, \qquad i = 1, \ldots, N-1$$

or

$$\int_{z_{i-1}(t)}^{z_i(t)} M(u(s,t))\,ds = \frac{1}{N}\int_0^1 M(u(z,t))\,dz, \qquad i = 1,\ldots,N. \tag{9}$$

One major issue associated with the generation of adaptive grids based on an equidistribution principle is the choice of a suitable monitor function. This is particularly true for complicated PDEs and in higher dimensions. For this paper, we restrict our attention to one of the standard methods and choose M to be a measure of the arc-length of a specified quantity, here the solution $u(z)$. That is, we take

$$M(u(z,t)) = \sqrt{\mu + \left(\frac{\partial u}{\partial z}(z,t)\right)^2} \tag{10}$$

where μ is a user-prescribed scaling parameter. Examples of the use of other monitor functions for 1D problems can be found in [13, 14, 15].

3.2 Adaptive solution procedure

The moving mesh algorithm has two key components, namely, the generation of a grid and the solution of the physical differential equation (equation (6)) on that grid. As (6) models a steady problem, an artificial time scale is introduced and these steps are carried out alternately until the mesh positions are no longer significantly changing.

In practice, the adaptive solution process was implemented using MATLAB [16] and the COMSOL Multiphysics finite element package [17]. The grid generation follows the method presented by Sanz-Serna and Christie in [18] which is equivalent to equidistributing the arc-length monitor function given by (10) over the interval $[0, d]$. The solution of (6) on the new grid and the advancement of the solution in time is carried out within COMSOL. Specifically, at each time step (6) is solved using finite element approximation with quadratic elements. The solution S is then updated in time via one step of the COMSOL time-dependent solver, `femtime`. Note that the initial grid points are uniformly distributed on $[0, d]$.

4 Analysis of mesh convergence

4.1 Analytic test problem

The aim of this section is to assess the level of accuracy obtained using the moving mesh method described above and a traditional uniform grid. To this end, we study a problem closely related to the model problem in §2.2 but whose Euler-Lagrange equation has an exact solution.

Specifically, we replace the quartic energy polynomial F_t in (5) by a simpler quadratic one, which we will call F_q. We choose F_q to be the quadratic which passes through the origin and has a minimum at the same point as F_t (that is, at (S_{eq}, F_{eq}) where $F_{eq} = F_t(S_{eq})$). This is the quadratic polynomial

$$F_q(S) = \frac{F_{eq}}{S_{eq}} S \left(2 - \frac{1}{S_{eq}} S\right).$$

Plots of F_q (dashed line) and F_t (solid line) in shown in Figure 1(a). The Euler-Lagrange equation associated with the new total energy $\hat{\mathcal{F}} = F_q + F_e$ is

$$\frac{\partial \hat{\mathcal{F}}}{\partial S} - \frac{d}{dz}\left(\frac{\partial \hat{F}}{\partial S'}\right) = 0 \qquad \Rightarrow \qquad S'' = \frac{6F_{eq}}{S_{eq}(2L_{1s}+1)}\left(1 - \frac{1}{S_{eq}} S\right).$$

Writing

$$\hat{\alpha} = \frac{6F_{eq}}{S_{eq}(2L_{1s}+1)}, \qquad \hat{\beta} = \frac{1}{S_{eq}},$$

this becomes

$$S'' = \hat{\alpha}(1 - \hat{\beta}S). \tag{11}$$

With boundary conditions $S = 0$ at $z = 0$ and $S = S_{eq}$ at $z = d_s$, this equation can be solved exactly to obtain the closed form solution

$$S(z) = S_{eq}\left(\frac{\sinh \rho z}{\tanh \rho d_s} - \cosh \rho z + 1\right) \tag{12}$$

where $\rho = \sqrt{|\hat{\alpha}\hat{\beta}|}$. Figures 1(b), (c) and (d) show a comparison of the solutions to (6) and (11) given by numerical integration of (7) (solid line) and evaluation of (12) for values of $d = 0.1, 1, 10$ microns. It can be seen that the shape of the two curves is very similar in each case, with a boundary layer of width ρ at $z = 0$. Note that for the physically realistic values of d considered here, $\tanh(\rho d_s) \simeq 1$ so in practice we use

$$S_{ex}(z) = S_{eq}\left[\sinh(\rho z) - \cosh(\rho z) + 1\right]. \tag{13}$$

to represent the 'exact' solution of (11).

4.2 Mesh convergence

We now analyse the convergence of the adaptive grid obtained by equidistributing the solution arc-length in the above problem. Specifically, we use the adaptive algorithm described in §3.2 with monitor function

$$M(S(z,t)) = \sqrt{1 + \left(\frac{dS}{dz}(z,t)\right)^2}.$$

The resulting mesh trajectories (plotted against time) with $N = 64$ and $d = 1$ micron are shown in Figure 2(a). Figure 2(b) shows a magnification of the trajectories over the first ten time steps (the time step size is $\delta t = 1e-7$). It can be seen that the node points move very quickly in response to the presence of the boundary layer.

The behaviour of the error as a grid is refined will clearly depend on the choice of norm used to measure that error. When considering problems with boundary layers, the appropriate norm is the maximum norm: other norms involve averaging techniques which smooth out the error and prevent the capture of local behaviour within the layer. To measure the error e in the linear interpolant, S_I say, of the solution (13) to (11), we use the L_∞ norm defined by

$$\|e\|_{L_\infty(0,d_s)} = \max_{z\in[0,d_s]} |S(z) - S_I(z)|. \tag{14}$$

and consider the L_∞ error incurred on a uniform grid and on an adaptive grid obtained as described above.

Suppose that $S_{ex}(z)$ is approximated by its piecewise linear interpolant, $S_I(z)$. From standard interpolation theory, there exists a positive constant c_1 independent of $S(z)$ such that

$$\|e\|_{L_\infty(z_{i-1},z_i)} \leq c_1(z_i - z_{i-1})^2 |S''_{ex}(\eta_i)|, \tag{15}$$

where $\eta_i \in (z_{i-1}, z_i)$. As

$$|S''_{ex}(z)| = |\rho^2 S_{eq}\left[\sinh(\rho z) - \cosh(\rho z)\right]| \leq \rho^2 S_{eq} \qquad \forall z \in [0, d_s],$$

on a uniform grid with $z_i - z_{i-1} = 1/N$ we have from (15) that

$$\|e\|_{L_\infty(0,d_s)} \leq \frac{C}{N^2} \tag{16}$$

where C is independent of N. For the adaptive grid, (15) can be used together with the equidistribution principle (9) to get

$$\|e\|_{L_\infty(z_{i-1},z_i)} \le c_2 \left(\int_{z_{i-1}}^{z_i} dz \right)^2 \le c_2 \left(\int_{z_{i-1}}^{z_i} \sqrt{1+(S'_{ex}(z))^2}\, dz \right)^2 = c_2 \left(\frac{1}{N} \int_0^{d_s} \sqrt{1+(S'_{ex}(z))^2}\, dz \right)^2$$

where $c_2 = c_1 \rho^2 S_{eq}$. Hence, there is also a bound of the form (16) in this case (with a different constant C).

Tables 1 and 2 show the values of a numerical approximation to the L_∞ error $L_\infty(N)$ for regions of $d = 0.1, 1, 10$ microns. The approximation was obtained by computing (14) on a refined grid with 10 evenly spaced points in each interval (z_{i-1}, z_i), $i = 1, \ldots, N$. Also shown is the convergence rate κ calculated by assuming a relationship of the form $L_\infty(N) = C/N^\kappa$ and comparing neighbouring lines in the table to obtain

$$\kappa = \log_2 \left(\frac{L_\infty(N)}{L_\infty(2N)} \right)$$

(as in [19]). It is clear from these results that, on both grids, the bound on the interpolation error (16) is sharp, that is, the error decays in practice like N^{-2} as predicted. Furthermore, for a given N, the adaptive grid error is always smaller than the uniform one. This implies that the constant C in (16) is smaller in the adaptive grid case so, in practice, the same specified order of accuracy can be reached using an adaptive grid with a smaller number of points.

5 Accuracy and efficiency in practice

For general situations (including the model problem described in §2.2), the exact solution will not be available to use in error computations. Accepted practice in the field suggests instead comparing computed solutions with a numerical solution calculated using a very fine uniform grid, using linear interpolation to obtain the values of this fine grid solution at the current grid points. Here we use a fine grid with $N = 131072$ and denote its interpolant by S_f. We also need to replace the L_∞ norm with a practical way of measuring the error based only on computed solution values at node points: we will use the l_∞ norm defined by

$$l_\infty(N) = \max_{i=0,\ldots,N/2} |S_f(z_i) - S_N(z_i)| \tag{17}$$

where S_f is as above and S_N is the computed solution for a particular value of N. Note that we only consider the left half of the domain $z \in [0, d/2]$ as that is where the solution is varying rapidly in this example. Evaluating (17) leads to the values in Tables 3 and 4, which are also plotted in Figure 3(a). Also shown is the convergence rate κ calculated by assuming a relationship of the form $l_\infty(N) = C/N^\kappa$ and comparing neighbouring lines in the table to obtain

$$\kappa = \log_2 \left(\frac{l_\infty(N)}{l_\infty(2N)} \right).$$

The convergence rate here appears to be $O(N^{-4})$ rather than $O(N^{-2})$: this is not surprising, as 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 [20].

It can be seen in Figure 3(a) that for two out of three cases, the error with the adaptive grid is mostly smaller than the error with a uniform grid with the same number of points. That is, a smaller number of points are required to achieve a fixed order of accuracy with an adaptive grid compared with a uniform one. The example where the reverse is true is the one where the width of the boundary layer is relatively large, that is, where the need for adaptivity is not clear.

The plateau effect in the error curves for the adaptive grids can be explained by considering an important caveat associated with this error estimator. When performing such a comparison with adaptive grids, care must be taken to monitor the minimum grid size. The smallest element size decreases very rapidly in the region of the boundary layer, so that even for grids with fairly small N, some elements may be smaller than those in the fine uniform grid (with $N = 131072$). This means that the l_∞ error (17) will be 'polluted' by interpolation issues in the boundary layer region. For this problem, the length of the smallest element (in microns) for each of the grids used is given in Table 5. For a region of size $d = 0.1$ microns, there is no problem. However, for $d = 1$ micron, the grids with $N = 2048, 4096, 8192$ have elements smaller than the smallest fine grid element. Similarly, for $d = 10$ microns, the grids with $N = 256, \ldots, 8192$ have the same problem.

Having established that an adaptive grid can improve accuracy (for a fixed number of grid points), we now consider the overhead associated with the computation of the adaptive grid points which should be taken into account when assessing the overall efficiency of the approach. To illustrate this, we focus on a region of length $d = 1$ micron. The number of points and CPU time in seconds required for the solution of (6) on various grids are listed in Table 6. The specific grids listed are those with smallest N which give accuracy within a specified tolerance, measured using l_∞ with S_f and S_N as before. The cost of calculating the adaptive grid is listed separately. The % speedup is calculated by taking the difference of the total uniform and adaptive grid times as a percentage of the uniform grid time. A graphical comparison of the total times for computation is given in Figure 3(b) where the horizontal axis represents the required tolerance (on a logarithmic scale) and the vertical axis shows the total CPU time in seconds for each grid. We observe that the relative efficiency of the adaptive method improves as higher accuracy (smaller tolerance) is required.

It is important that we do not place too much emphasis on these results as it is difficult to quantify efficiency in a meaningful way for these model 1D problems. They involve only the solution of a single ODE, which is relatively cheap to solve numerically. We anticipate, however, that for higher-dimensional problems with full Q-tensor theory, this will no longer be the case and the efficiency saving from adaptive grids will become more pronounced.

6 Observations and conclusions

In this paper we have presented the results of a feasibility study of the use of moving mesh methods for solving one-dimensional problems in Q-tensor theory. The specific method used, that is, that of equidistributing the arc-length of some property of the solution, proved to be very effective. For a model problem with an exact solution, it was shown that the adaptive grids provided the same level of accuracy as uniform grids, but using far fewer points, resulting in a computational saving. The results of further experiments on a more realistic 1D order reconstruction problem are presented in [21]. In that case the material is no longer uniaxial so a system of 6 coupled PDEs must be solved in conjunction with the moving mesh. The results stated there support the observations made here, that significant computational savings can be made with the moving mesh method. Finally, we anticipate that adaptive grid methods have the potential for similar gains in two- and three-dimensional problems, and have begun work to verify this.

References

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

[2] K. Miller and R. Miller, "Moving finite elements i," *SIAM J. Numer. Anal.*, vol. 18, p. 1019, 1981.

[3] R. R. W. Cao, W. Huang, "A moving mesh method based on the geometric conservation law," *SIAM J. Sci. Comput.*, vol. 124, p. 118, 2002.

[4] P. Knupp, "Mesh generation using vector-fields," *J. Comput. Phys.*, vol. 119, p. 142, 1995.

[5] A. Winslow, "Numerical solution of the quasi-linear poisson equations in a nonuniform triangle mesh," *J. Comput. Phys.*, vol. 1, p. 149, 1967.

[6] A. Dvinsky, "Adaptive grid generation from harmonic maps on riemannian manifolds," *J. Comput. Phys.*, vol. 95, p. 450, 1991.

[7] R. R. W. Cao, W. Huang, "Approaches for generating moving adaptive meshes: location versus velocity," *Appl. Numer. Math.*, vol. 47, p. 121, 2003.

[8] N. J. Mottram and C. J. Newton, "Introduction to Q-tensor theory," Tech. Rep. 10/04, University of Strathclyde, 2004.

[9] P. de Gennes, "An analogy between superconductors and smectics a," *Solid State Commun.*, vol. 10, p. 753, 1972.

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

[11] R. Barberi, F. Ciuchi, G. 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*, vol. 13, pp. 61–71, 2004.

[12] C. de Boor, "Good approximation by splines with variable knots ii," in *Conference on the Applications of Numerical Analysis, Dundee 1973* (J. Morris, ed.), vol. 363 of *Lecture Notes in Mathematics*, Berlin: Springer-Verlag, 1974.

[13] M. G. Beckett, J. A. Mackenzie, A. Ramage, and D. M. Sloan, "On the numerical solution of one-dimensioanl pdes using adaptive methods based on equidistibution," *J. Comput. Phys.*, vol. 167, pp. 372–392, 2001.

[14] L. S. Mulholland, Y. Qiu, and D. M. Sloan, "Solution of evolutionary partialdifferential equations using adaptive finite differences with pseudospectral post-processing," *J. Comput. Phys.*, vol. 131, pp. 280–298, 1997.

[15] W. Ren and X.-P. Wang, "An iterative grid redistribution method for singular problems in mulitple dimensions," *J. Comput. Phys.*, vol. 159, pp. 246–273, 2000.

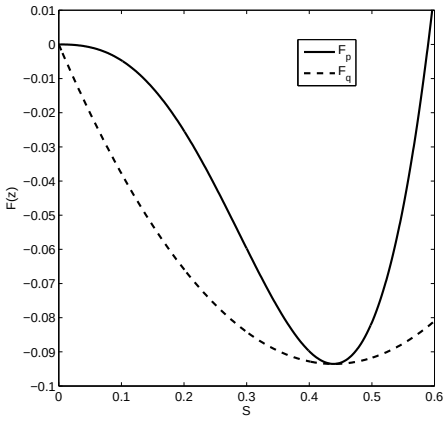
[16] Mathworks, Inc., "MATLAB™," *http://www.mathworks.com/*, vol. version 7.2.0.232, 2006.

[17] COMSOL Group, "COMSOL multiphysics™," *http://www.comsol.com/*, vol. version 3.2, 2006.

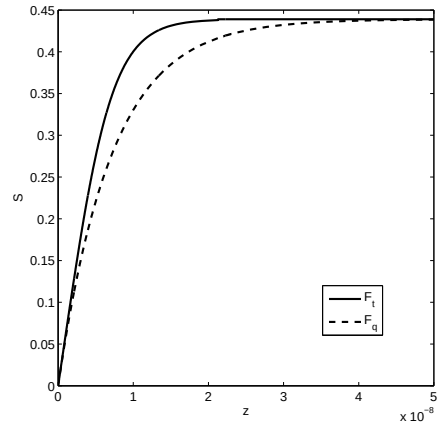
[18] J. M. Sanz-Serna and I. Christie, "A simple adaptive technique for nonlinear wave problems," *J. Comput. Phys.*, vol. 67, pp. 348–360, 1986.

[19] A. F. Hegarty, E. O'Riordan, and M. Stynes, "A comparison of uniformly convergent difference schemes for two-dimensional convection-diffusion problems," *J. Comput. Phys.*, vol. 105, pp. 24–32, 1993.

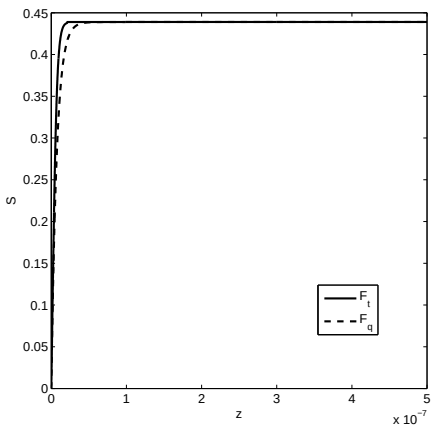
[20] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems.* London: Academic Press, 1984.



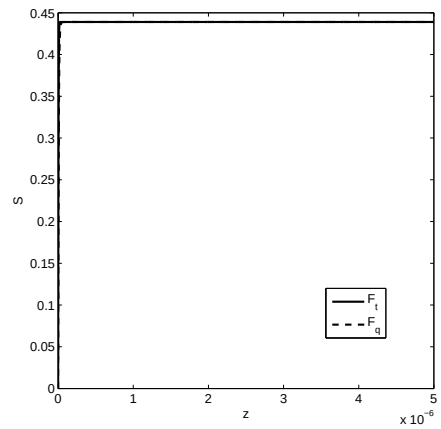
(a) Energy polynomials F_t and F_q plotted against S.



(b) Solutions computed with F_t and F_q for $d = 0.1$ microns.



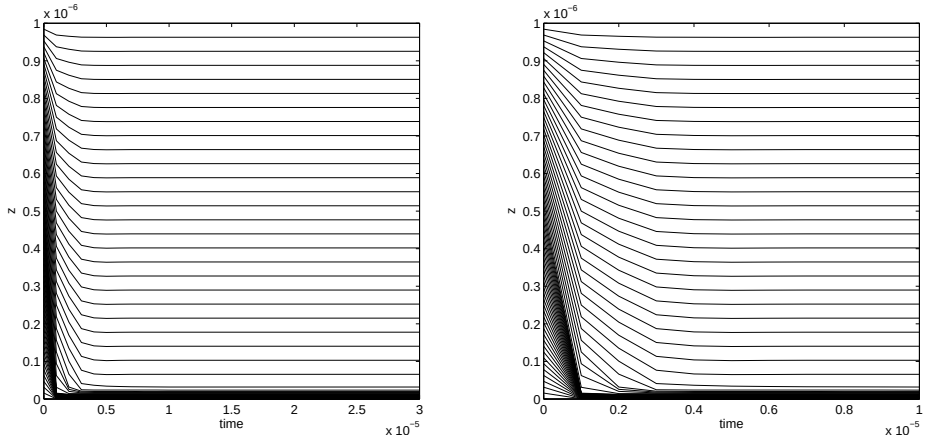
(c) Solutions computed with F_t and F_q for $d = 1$ micron.



(d) Solutions computed with F_t and F_q for $d = 10$ microns.

Figure 1: Comparison of energy expressions and solutions for F_t (solid) and F_q (dashed).

[21] A. Ramage and C. J. Newton, "Adaptive solution of a one-dimensional order reconstruction problem in Q-tensor theory of liquid crystals," Tech. Rep. 14/06, University of Strathclyde, 2006.



(a) Node trajectories against time. (b) Magnification up to time $1e-6$.

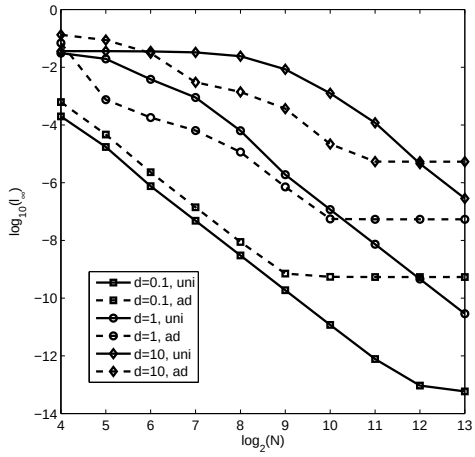
Figure 2: Mesh trajectories for $N = 64$ and $d = 1$ micron.

	$d = 0.1$		$d = 1$		$d = 10$	
N	$\|e\|_{L_\infty}$	κ	$\|e\|_{L_\infty}$	κ	$\|e\|_{L_\infty}$	κ
16	2.7434e-02	-	2.7526e-01	-	3.9499e-01	-
32	8.4297e-03	1.7024e+00	1.9036e-01	5.3201e-01	3.8947e-01	2.0316e-02
64	2.3433e-03	1.8469e+00	9.9757e-02	9.3226e-01	3.4557e-01	1.7252e-01
128	6.1820e-04	1.9224e+00	3.8780e-02	1.3631e+00	3.0161e-01	1.9629e-01
256	1.5879e-04	1.9609e+00	1.2500e-02	1.6334e+00	2.2234e-01	4.3991e-01
512	4.0241e-05	1.9804e+00	3.5649e-03	1.8100e+00	1.2738e-01	8.0369e-01
1024	1.0129e-05	1.9902e+00	9.5298e-04	1.9033e+00	5.3847e-02	1.2421e+00
2048	2.5409e-06	1.9951e+00	2.4643e-04	1.9512e+00	1.8308e-02	1.5564e+00
4096	6.3630e-07	1.9975e+00	6.2663e-05	1.9755e+00	5.3882e-03	1.7646e+00
8192	1.5921e-07	1.9988e+00	1.5800e-05	1.9877e+00	1.4642e-03	1.8797e+00

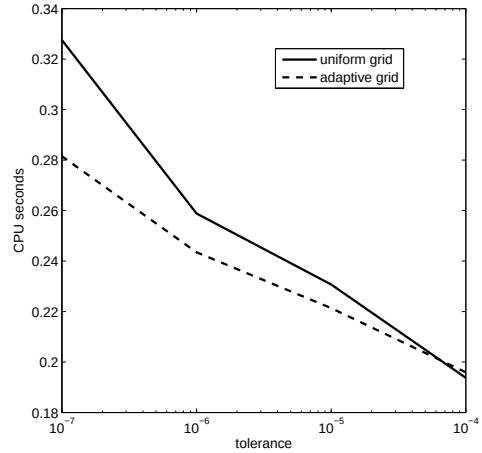
Table 1: L_∞ errors and convergence rates on uniform grid.

	$d = 0.1$		$d = 1$		$d = 10$	
N	$\|e\|_{L_\infty}$	κ	$\|e\|_{L_\infty}$	κ	$\|e\|_{L_\infty}$	κ
16	4.9004e-03	-	3.8964e-02	-	9.9690e-02	-
32	1.2428e-03	1.9793e+00	5.4055e-03	2.8497e+00	3.9438e-02	1.3379e+00
64	3.1778e-04	1.9675e+00	2.4165e-03	1.1615e+00	1.9156e-02	1.0418e+00
128	7.9331e-05	2.0021e+00	8.6752e-04	1.4779e+00	3.7886e-03	2.3381e+00
256	1.9886e-05	1.9961e+00	2.2710e-04	1.9336e+00	1.6648e-03	1.1863e+00
512	4.9733e-06	1.9995e+00	5.7810e-05	1.9739e+00	4.6253e-04	1.8478e+00
1024	1.2432e-06	2.0001e+00	1.4566e-05	1.9888e+00	1.3971e-04	1.7271e+00
2048	3.1080e-07	2.0000e+00	3.6389e-06	2.0010e+00	3.4720e-05	2.0086e+00
4096	7.7700e-08	2.0000e+00	9.1104e-07	1.9979e+00	9.2870e-06	1.9025e+00
8192	1.9425e-08	2.0000e+00	2.2782e-07	1.9996e+00	2.3245e-06	1.9983e+00

Table 2: L_∞ errors and convergence rates on adaptive grid.



(a) Comparison of uniform grid (uni) and adaptive grid (ad) l_∞ errors for various region sizes.



(b) CPU times (in seconds) required to solve (6) on various adaptive and uniform grids.

Figure 3: Data on relative accuracy and efficiency of uniform and adaptive grids.

	$d = 0.1$		$d = 1$		$d = 10$	
N	$\|e\|_{l_\infty}$	κ rate	$\|e\|_{l_\infty}$	κ	$\|e\|_{l_\infty}$	κ
16	1.9922e-04	-	3.1000e-02	-	3.6447e-02	-
32	1.7347e-05	3.5216e+00	1.9687e-02	6.5506e-01	3.6265e-02	7.2194e-03
64	7.5938e-07	4.5138e+00	3.8344e-03	2.3602e+00	3.5548e-02	2.8783e-02
128	4.8189e-08	3.9781e+00	9.0050e-04	2.0902e+00	3.2855e-02	1.1368e-01
256	3.0142e-09	3.9989e+00	6.3546e-05	3.8248e+00	2.4325e-02	4.3365e-01
512	1.8846e-10	3.9994e+00	1.9001e-06	5.0637e+00	8.5313e-03	1.5116e+00
1024	1.1816e-11	3.9955e+00	1.1759e-07	4.0142e+00	1.2626e-03	2.7563e+00
2048	7.7471e-13	3.9309e+00	7.3575e-09	3.9984e+00	1.1928e-04	3.4040e+00
4096	9.2815e-14	3.0612e+00	4.5999e-10	3.9995e+00	4.6087e-06	4.6938e+00
8192	5.8287e-14	6.7119e-01	2.8748e-11	4.0001e+00	2.8524e-07	4.0141e+00

Table 3: l_∞ errors and convergence rates on uniform grid.

	$d = 0.1$		$d = 1$		$d = 10$	
N	l_∞	κ	l_∞	κ	l_∞	κ
16	6.2678e-04	-	6.9459e-02	-	1.3445e-01	-
32	4.6690e-05	3.7468e+00	7.5498e-04	6.5236e+00	8.8414e-02	6.0477e-01
64	2.3167e-06	4.3330e+00	1.8173e-04	2.0547e+00	3.1026e-02	1.5108e+00
128	1.4105e-07	4.0378e+00	6.4331e-05	1.4982e+00	3.0290e-03	3.3565e+00
256	8.8425e-09	3.9956e+00	1.1572e-05	2.4749e+00	1.4057e-03	1.1076e+00
512	7.0892e-10	3.6408e+00	7.1329e-07	4.0200e+00	3.6785e-04	1.9341e+00
1024	5.4601e-10	3.7670e-01	5.5444e-08	3.6854e+00	2.1998e-05	4.0637e+00
2048	5.3823e-10	2.0694e-02	5.3839e-08	4.2356e-02	5.3920e-06	2.0285e+00
4096	5.3765e-10	1.5469e-03	5.3769e-08	1.8882e-03	5.3757e-06	4.3632e-03
8192	5.3751e-10	3.8510e-04	5.3758e-08	2.9362e-04	5.3755e-06	4.7443e-05

Table 4: l_∞ errors and convergence rates on adaptive grid.

	$d = 0.1$		$d = 1$		$d = 10$	
N	uniform	adaptive	uniform	adaptive	uniform	adaptive
16	6.2499e-03	6.9780e-04	6.2499e-02	6.1150e-04	6.2499e-01	5.2374e-04
32	3.1250e-03	3.5107e-04	3.1250e-02	3.7623e-04	3.1250e-01	3.2209e-04
64	1.5625e-03	1.7602e-04	1.5625e-02	1.8842e-04	1.5625e-01	1.8182e-04
128	7.8124e-04	8.8110e-05	7.8124e-03	9.4317e-05	7.8124e-02	9.5079e-05
256	3.9062e-04	4.4078e-05	3.9062e-03	4.7198e-05	3.9062e-02	4.7607e-05
512	1.9531e-04	2.2045e-05	1.9531e-03	2.3609e-05	1.9531e-02	2.3826e-05
1024	9.7655e-05	1.1024e-05	9.7655e-04	1.1807e-05	9.7655e-03	1.1921e-05
2048	4.8828e-05	5.5122e-06	4.8828e-04	5.9039e-06	4.8828e-03	5.9614e-06
4096	2.4414e-05	2.7562e-06	2.4414e-04	2.9520e-06	2.4414e-03	2.9809e-06
8192	1.2207e-05	1.3781e-06	1.2207e-04	1.4760e-06	1.2207e-03	1.4905e-06
131072	7.6293e-07	-	7.6293e-06	-	7.6293e-05	-

Table 5: Minimum grid sizes in microns.

	Uniform Grid		Adaptive Grid				
tolerance	N	total	N	solve	grid	total	% speedup
1×10^{-4}	174	1.9371e-1	115	1.9491e-1	7.6075e-4	1.9590e-1	-1.13
1×10^{-5}	338	2.3071e-1	258	2.2016e-1	9.7614e-4	2.2137e-1	4.05
1×10^{-6}	568	2.5882e-1	476	2.4822e-1	1.3882e-3	2.4344e-1	5.94
1×10^{-7}	1051	3.2753e-1	817	2.7932e-1	1.9825e-3	2.8150e-1	14.05

Table 6: CPU times (in seconds) required to solve (6) on various adaptive and uniform grids.