Numerical Simulation of Flow in Smectic Liquid Crystals

Merlin Fallahpour\textsuperscript{a}, Sean McKee\textsuperscript{b}, and Ewa B. Weinmüller\textsuperscript{a}

\textsuperscript{a}Department of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, A-1040 Wien, Austria
e-mail: merlin.fallahpour@gmail.com e.weinmueller@tuwien.ac.at

\textsuperscript{b}Department of Mathematics and Statistics, University of Strathclyde, Livingstone Tower, 26 Richmond Street, Glasgow, G1 1XH, United Kingdom
e-mail: sean.mckee@strath.ac.uk

Abstract

Our aim is to simulate a nonlinear system of ODEs describing the flow in smectic liquid crystals. The nonlinear system is first linearized. We present a direct approach to compute the exact analytic solution of this linear system and use this solution as a starting profile in the MATLAB package \texttt{bvpsuite2.0} to obtain the approximate solution to the nonlinear system. Although, the solution of the nonlinear system has steep boundary layers and therefore is difficult to resolve, we demonstrate that \texttt{bvpsuite2.0} can cope with the problem and provide an approximation with reasonable accuracy.

1 Introduction

Nematic liquid crystals consist of rod-like molecules that have preferred local average direction: this direction is usually characterized by a unit vector $\mathbf{n}$, called the director. Smectic liquid crystals (SmA) tend to have more order: they occur when the molecules are arranged in parallel layers with the director parallel to the unit layer normal $\mathbf{a}$ as displayed in Figure 1.

![Figure 1: Undistorted SmA](image-url)
This is somewhat of an idealization and discrepancies between $a$ and $n$ can occur, see Figure 2. This is particularly noticeable in flow problems where the flow is normal to the layers (i.e. in the $x$-direction in Figure 2 where a constant pressure gradient exists in that direction).

Thus, the problem that we wish to consider here is a pressure-driven ‘Poiseuille flow’ between two plates in the $x$-direction of a SmA liquid crystal where $a$ and $n$ are allowed to deviate from one and other. We shall do this by invoking the theory developed by Stewart [13]. Rather than derive the nonlinear equations ab initio, we shall simply write them down (for details of the technical argument, see [14]).

However, before we do this, we should point out that this problem has been discussed by de Gennes [5] and de Gennes and Prost [6] in the case of a linearized model when $a$ and $n$ are constrained to be parallel. Indeed, de Gennes argues that the flow velocity $u$, normal to the layers is given by

$$u(z) = \lambda_p \frac{dp}{dx} \left( \frac{1 - \cosh \kappa z \sinh \kappa d/2}{\sinh \kappa d/2} \right)$$

where $\lambda_p$ is the permeation coefficient, $d$ is the plate separation distance, $\kappa = \frac{1}{\sqrt{\eta \lambda_p}}$ is comparable to the smectic interlayer distance and $\eta$ is the one-constant approximation of the viscosity.

As we shall see our numerical results predict not only a boundary layer for $u$, but also boundary layers (approximately half the length) for both $\theta$ and $\delta$ (angles, respectively, associated with $n$ and $a$ - see Figure 2).

This note may be regarded as an extension of the paper by Stewart et al. [14]. In that paper the nonlinear BVP was studied using asymptotic methods. The purpose of this note is to provide accurate and efficient collocation software for determining the numerical solution of this demanding BVP with three boundary layers.
2 The Nonlinear Differential Equations

We consider the following system of differential equations for the unknown functions \( \theta(z) \), \( \delta(z) \) and \( u(z) \) [14]:

\[
M(\theta, \delta) \sin(\theta - \delta) - K^n_1 \cos(\theta) \frac{d^2 \sin(\theta)}{dz^2} + \frac{du}{dz} \left[ \alpha_3 \cos(\theta)^2 - \alpha_2 \sin(\theta)^2 + \kappa_1 \cos(\theta - \delta) \right] = 0, \tag{1}
\]

\[
a - J_{3,3} - \tilde{t}_{13,3} = 0, \tag{2}
\]

\[
u + \lambda p J_{3,3} = 0, \tag{3}
\]

where

\[
J_{3,3} = \frac{dJ_3}{dz}, \quad \tilde{t}_{13,3} = \frac{d\tilde{t}_{13}}{dz},
\]

and

\[
J_3 = \cos(\delta)^2 \left[ K^n_1 \cos(\delta) \frac{d^2 \sin(\delta)}{dz^2} + M(\theta, \delta) \sin(\theta - \delta) \right]
- B_0 \sin(\delta) \left[ \sec(\delta) + \cos(\theta - \delta) - 2 \right],
\]

\[
\tilde{t}_{13} = \frac{1}{2} \frac{du}{dz} \left( \alpha_4 + \alpha_5 - \alpha_2 + \tau_2 \right) + \frac{1}{4} \frac{du}{dz} \left[ \alpha_1 \sin(2\theta)^2 + \tau_1 \sin(2\delta)^2 \right] + \frac{1}{2} \frac{du}{dz} \left[ \kappa_2 \cos(\theta + \delta) \right]
+ \kappa_6 \cos(\theta - \delta) + (\alpha_2 + \alpha_3) \cos(\theta)^2
+ \frac{1}{2} \frac{du}{dz} \left[ \kappa_2 \sin(\theta + \delta)^2 \right]
+ \kappa_3 \sin(2\theta) \sin(2\delta) \right] + \frac{du}{dz} \left[ \kappa_4 \sin(2\theta) + \kappa_5 \sin(2\delta) \right].
\]

For notational convenience, the function \( M(\theta, \delta) \) has been introduced as

\[
M(\theta, \delta) = B_1 \cos(\theta - \delta) - B_0 \left[ \sec(\delta) + \cos(\theta - \delta) - 2 \right].
\]

Here, \( K^n_1, K_1^n, \lambda_p, \alpha, \alpha_1 \) to \( \alpha_5, B_0, B_1, \tau_1, \tau_2, \) and \( \kappa_1 \) to \( \kappa_6 \) are given constant parameters, see Table 1. Equations (1)–(3) are subject to following boundary conditions:

\[
u(\pm d/2) = 0, \quad \theta(\pm d/2) = \pm \theta_0, \quad \delta(\pm d/2) = \pm \delta_0, \quad \delta''(0) = 0. \tag{4}
\]

We first find the solution to the linearized system of equations derived from (1)–(3) under the assumption that \( u, \theta \) and \( \delta \) and their derivatives are small. In [14], the analytical solution to the linear problem has already been provided, for details see also [4]. However, the reasoning presented here is arguably more straightforward and technically less involved; the solution takes also a more compact form. With the numerical solution of the linear system of equations, we anticipate that we have found a good starting guess for the numerical simulation of the nonlinear system (1)–(3). Respective numerical experiments are carried out using the MATLAB code bvpsuite2.0 [16, 2]. This code is an updated version of the older MATLAB software bvpsuite1.1 [10].
3 Analytical properties of the linear problem

We first assume that $u, \theta, \delta$, and their derivatives are small. Moreover, we use for small values of $x$ the approximations $\sin(x) \approx x$ and $\cos(x) \approx 1$. Then the three linear equations follow:

\begin{align*}
B_1(\theta - \delta) - K_1^n \theta'' + u'(\alpha_3 + \kappa_1) &= 0, \quad (5) \\
K_1^n \delta'' + B_1(\theta' - \delta') + \eta u'' - a &= 0, \quad (6) \\
u + \lambda_p [K_1^n \delta'' + B_1(\theta' - \delta')] &= 0, \quad (7)
\end{align*}

where $2\eta = \alpha_2 + \alpha_4 + \alpha_5 + \tau_2 + 2(\alpha_3 + \kappa_1 + \kappa_6)$.

These differential equations are subject to the boundary conditions (4). Now, we provide the exact solution of the BVP (4)–(7). We first multiply (7) by $\lambda_p^{-1}$ and subtract from (6). This yields a second order BVP for $u$,

\begin{equation}
\frac{u'' - \frac{u}{\eta \lambda_p}}{\frac{u}{\eta}} = \frac{a}{\eta}, \quad u(\pm \frac{d}{2}) = 0.
\end{equation}

The solution to (8),

\begin{equation}
u(z) = \alpha \lambda_p \left( \frac{\cosh(\lambda_1 z)}{\cosh(\lambda_1 d/2)} - 1 \right), \quad \lambda_1 = \kappa = \frac{1}{\sqrt{\eta \lambda_p}},
\end{equation}

is immediate. By differentiating (5) and subtracting from (6), we obtain

\begin{equation}
\theta''' + \delta''' = \frac{1}{K_1^n} (a - \nu u''), \quad \nu = \eta - \alpha_3 - \kappa_1.
\end{equation}

where the reasonable assumption that $K_1^n = K_1^n$ has been made and $\nu = \eta - \alpha_3 - \kappa_1$.

The latter equation is now integrated three times. This yields

\begin{equation}
\theta + \delta = \frac{1}{K_1^n} \left[ \frac{a z^3}{6} - \frac{\nu \alpha \lambda_p}{\lambda_1} \left( \frac{\sinh(\lambda_1 z)}{\cosh(\lambda_1 d/2)} \right) \right] + c_0 + c_1 z + c_2 z^2.
\end{equation}

Finally, we substitute $\theta - \delta = - (\theta + \delta) + 2 \theta$ into (5) with $\theta + \delta$ specified in (9). This gives a second order ODE for $\theta$ and introduces two further unknown constants. Using (9), we can now calculate $\delta$. Both, $\theta$ and $\delta$ depend on five constants which are computed using

\begin{equation}
\theta(\pm d/2) = \pm \theta_0, \quad \delta(\pm d/2) = \pm \delta_0, \quad \delta''(0) = 0.
\end{equation}

For details see [4].

To summarize, the solution of (4)–(7) reads:

\begin{align*}
\theta(z) &= \frac{a \lambda_p}{2 \lambda_1} \left[ (2\eta - 2\nu) \lambda_1^3 + \nu \lambda_2^3 \right] \frac{\sinh(\lambda_1 z)}{\cosh(\lambda_1 d/2)} + f(z) - g(z), \quad (10) \\
\delta(z) &= \frac{a \lambda_p}{2 \lambda_1} \left[ \nu \lambda_2^3 - 2\eta \lambda_2^3 \right] \frac{\sinh(\lambda_1 z)}{\cosh(\lambda_1 d/2)} + f(z) + g(z), \quad (11) \\
u(z) &= a \lambda_p \left[ \frac{\cosh(\lambda_1 z)}{\cosh(\lambda_1 d/2)} - 1 \right], \quad (12)
\end{align*}

For details see [4].
where

\[ f(z) = \frac{a}{12K_1^2} z^3 + \left[ \frac{1}{d} (\delta_0 + \theta_0) - \frac{1}{dK_1^2} \left( \frac{a}{48} d^3 - \frac{a\lambda_p\nu}{\lambda_1} \sinh(\lambda_1 d/2) \right) \right] z, \]

\[ g(z) = \left[ \frac{\delta_0 - \theta_0}{2} + \frac{a\lambda_p\lambda_1^2 (2\eta - \nu)}{2K_1^2 \lambda_1 (\lambda_1^2 - \lambda_2^2) \cosh(\lambda_1 d/2)} \right] \sinh(\lambda_2 z) \sinh(\lambda_2 d/2) - \frac{a}{4B_1} z, \]

and \( \lambda_2 = \sqrt{\frac{2B_1 K_1}{K_1^2}} \). Note that \( \lambda_1^{-1} \) and \( \lambda_2^{-1} \) provide a measure of the two boundary layers in the BVP (4)–(7). Using Maple, we have verified that the above functions solve the system of equations (5)–(7), satisfy boundary conditions (4) and are identical with the solution presented in [14, p.1828].

4 Numerical simulation of the problem

We use the package \texttt{bvpsuite2.0} to approximate solutions to the linear BVP (4)–(7). This code is a new version\(^1\) of the \texttt{bvpsuite1.1} which was originally designed to solve implicit nonlinear BVPs in ODEs and differential algebraic equations. The software can also cope with ODEs with singular points which are typically located at the boundaries of the interval. Clearly, ODEs without singular points can also be simulated [10].

The solver routine in \texttt{bvpsuite1.1} is based on a class of polynomial collocation methods whose orders may vary from 2 to 8. Collocation has been investigated for the class of singular differential equations in [8, 11, 15]. It turns out that this method is robust with respect to singularities in the independent variable and, furthermore, it retains its high convergence order in the case when the analytical solution is appropriately smooth. The code also provides an asymptotically correct estimate for the global error of the numerical approximation. To enhance the efficiency of the method, a mesh adaptation strategy is implemented, which attempts to choose grids related to the solution behaviour, in such a way that the tolerance is satisfied with the least possible computational effort. The error estimation procedure and the mesh adaptation work dependably provided that the solution of the problem and its global error are appropriately smooth\(^2\). For further information see [10]. The software proved useful for the approximation of numerous singular BVPs arising in applications from science and technology, see e.g. [3, 7, 9, 12, 1].

To describe the collocation approach used in \texttt{bvpsuite1.1}, we consider the problem

\[ f(x, y(x), y'(x), y''(x), y'''(x)) = 0, \quad x \in [0, 1]. \quad (13) \]

subject to three boundary conditions,

\[ b(y(0), y'(0), y''(0), y(1), y'(1), y''(1)) = 0. \quad (14) \]

---

\(^1\)Main difference between the two versions of the software is that in the new code, we removed the GUI and restructured the package to become modular. We plan to publish \texttt{bvpsuite2.0} soon.

\(^2\)The required smoothness of higher derivatives is related to the order of the collocation method employed.
First, the interval of integration \([0, 1]\) is partitioned,
\[
\Delta := \{0 = x_0 < x_1 < \ldots < x_{l-1} < x_l = 1, \ x_j = jh, \ j = 0, \ldots, I = 1/h\},
\]
and in each subinterval \([x_j, x_{j+1}]\), we introduce \(k \in \mathbb{N}\) collocation nodes \(x_{jl} := x_j + u_lh, \ j = 0, \ldots, I - 1, \ l = 1, \ldots, k\), where \(0 < u_1 < \ldots < u_k < 1^3\).

By \(P_{k,h}\), we denote the class of piecewise polynomial functions which are *globally* in \(C^2[0, 1]\) and reduce in each subinterval \([x_j, x_{j+1}]\) to a polynomial of degree less than or equal to \(k+2\). We now approximate the analytical solution \(y\) by a polynomial function \(p \in P_{k,h}\), such that \(p\) satisfies the system (13) at the collocation points,
\[
f(x_{jl}, p(x_{jl}), p'(x_{jl}), p''(x_{jl})) = 0, \ l = 1, \ldots, k, \ j = 0, \ldots, I - 1,
\]
the boundary conditions
\[
b(p(0), p'(0), p''(0), p(1), p'(1), p''(1)) = 0,
\]
and the continuity relations,
\[
p_{j-1}(x_j) = p_j(x_j), \ p'_{j-1}(x_j) = p'_j(x_j), \ p''_{j-1}(x_j) = p''_j(x_j), \ j = 1, \ldots, I - 1,
\]
where \(p(x) := p_j(x), \ x \in [x_j, x_{j+1}]\). Inspecting the number of unknowns and the number of equations, we immediately see that the discrete system is closed.

In the numerical simulation of the linear and nonlinear problem considered here, we face the challenge that all three dependent variables have steep boundary layers. This means that we need to work with a low order collocation and anticipate dense meshes in case of strict tolerances. Typical values of the material parameters involved can be found in [14, pp.1829-1830], see Table 1.

We denote the approximations to \(\theta, \delta, \text{ and } u\) by \(\hat{\theta}, \hat{\Delta}, \text{ and } \hat{U}\), respectively. For the simulation \(u\) was scaled by the factor \(10^{13}\) in order to make \(\hat{U} \approx 10^{13} u\) an \(O(1)\) function, similar in size to \(\Theta\) and \(\Delta\). Consequently, \(\Theta \approx \theta, \Delta \approx \delta, \text{ and } \hat{U} \approx 10^{13} u\). The resulting solution plots for the linear problem (4)–(7) can be found in Figures 3 and 4. The error estimates, automatically generated by *bvpsuite2.0* are shown in Figure 5. We used collocation with two Gaussian points and set the absolute and relative tolerance requirements to \(Tol_a = Tol_r = 10^{-6}\). The initial grid contained 1,000 points, the final grid around 28,000 points.

Again, the package *bvpsuite2.0* was used to obtain approximations of the solution to the nonlinear system (1)–(3) subject to boundary conditions (4). The final absolute and relative tolerance requirements were set to \(Tol_a = Tol_r = 10^{-3}\). In the first step, the required tolerances were \(Tol_a = Tol_r = 10^{-1}\). Here, as a starting profile, the numerical solution to the linear problem was used. The numerical solution from the run with \(Tol_a = Tol_r = 10^{-1}\) became then a starting profile for the tolerances \(Tol_a = Tol_r = 10^{-2}\) and so on. In all three steps, collocation with one collocation point on an initial grid with 10,000 mesh points was applied. The automatically chosen final grids contained around 10,000, 14,000 and 41,000 points to satisfy the tolerances \(10^{-1}, 10^{-2}\) and \(10^{-3}\), respectively. The results of the numerical calculations for the third step are shown in Figures 6, 7 and 8.

---

\(^3\)The values \(u_1 = 0\) and \(u_k = 1\) are excluded to avoid the evaluation at possibly singular points located at the boundaries of the interval \([0, 1]\)
### Table 1: Typical material parameters specified in [14]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>$10^{-5}$ m</td>
<td>plate separation distance</td>
</tr>
<tr>
<td>$K_1^a = K_1^n$</td>
<td>$5 \times 10^{-12}$ N</td>
<td>elastic constant</td>
</tr>
<tr>
<td>$B_0$</td>
<td>$8.95 \times 10^7$ N m$^{-2}$</td>
<td>layer compression constant</td>
</tr>
<tr>
<td>$B_1$</td>
<td>$4 \times 10^7$ N m$^{-2}$</td>
<td>constant accounting for the strength between n and a</td>
</tr>
<tr>
<td>$\lambda_p$</td>
<td>$10^{-16}$ m$^2$ Pa$^{-1}$ s$^{-1}$</td>
<td>permeation coefficient</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>$-0.0060$ Pa s</td>
<td>nematic-like viscosity coefficient</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>$-0.0812$ Pa s</td>
<td>nematic-like viscosity coefficient</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>$-0.0036$ Pa s</td>
<td>nematic-like viscosity coefficient</td>
</tr>
<tr>
<td>$\alpha_4, \tau_1, \tau_2$</td>
<td>0.0652 Pa s</td>
<td>analogous to the classical incompressible SmA viscosities</td>
</tr>
<tr>
<td>$\alpha_5$</td>
<td>0.0640 Pa s</td>
<td>nematic-like viscosity coefficient</td>
</tr>
<tr>
<td>$\kappa_1, \kappa_2, \ldots, \kappa_6$</td>
<td>0.0020 Pa s</td>
<td>“coupling” viscosity coefficients</td>
</tr>
<tr>
<td>$a$</td>
<td>$0.2$ rad</td>
<td>constant pressure gradient</td>
</tr>
<tr>
<td>$\theta_0$</td>
<td>0.2 rad</td>
<td>constant angle at the boundary for $\theta$</td>
</tr>
<tr>
<td>$\delta_0$</td>
<td>0.15 rad</td>
<td>constant angle at the boundary for $\delta$</td>
</tr>
</tbody>
</table>

### 5 Conclusions

We have considered constant pressure-driven ‘Poiseuille flow’ in a SmA liquid crystal in standard book-shelf configuration, see Figure 1 or [13, p.276], within two parallel plates where both the director and the layer normal are anchored on the plates, but are free to deviate from one another throughout the layers.

An analytic solution for the linearized equation was developed to act as an initial guess for solving the nonlinear BVP using the MATLAB collocation routine `bvp suite2.0`. This routine successfully provided the solution of the nonlinear problem with a prescribed accuracy, regardless of the steep boundary layers which are known to be difficult to resolve.

It is interesting to note that the boundary layer associated with $u$ is approximately $5 \times 10^{-8}$ m for both the linear and nonlinear case. However, $\delta$ and $\theta$ have boundary layers of the order of $10^{-9}$ m in the linear case, while for the nonlinear problem they are considerably larger, approximately $2.5 \times 10^{-8}$ m. This difference is possibly attributable to the linearising approximations, $\sin(x) \approx x, \cos(x) \approx 1$, which are certainly not true in the boundary layers. This strengthens the case for a numerical solution since the ‘simple’ linear problem can be somewhat misleading.
Figure 3: BVP (4)–(7): Approximations $\tilde{\Theta}(z) \approx \theta(z)$ and $\tilde{\Delta}(z) \approx \delta(z)$. In the lower graphs, the areas around the interval boundaries are zoomed in, to enhance the visibility of the solution behaviour.

Figure 4: BVP (4)–(7): Approximation $\tilde{U}(z) \approx 10^{13} u(z)$. In the lower graph, the area around the right boundary has been zoomed in.
Figure 5: BVP (4)–(7): Absolute global errors of the approximations $\tilde{\Theta}(z) \approx \theta(z)$, $\tilde{\Delta}(z) \approx \delta(z)$ and $U(z) \approx 10^{13} u(z)$.

Figure 6: BVP (1)–(4): $Tol_a = Tol_r = 10^{-3}$: Approximations $\tilde{\Theta}(z) \approx \theta(z)$ and $\tilde{\Delta}(z) \approx \delta(z)$. In the lower graphs, the areas around the interval boundaries are zoomed in.
Figure 7: BVP (1)–(4): $Tol_a = Tol_r = 10^{-3}$: Approximation $\tilde{U}(z) \approx 10^{13} u(z)$. In the lower graph, the area around the right boundary has been zoomed in.

Figure 8: BVP (1)–(4): $Tol_a = Tol_r = 10^{-3}$: Absolute global errors of the approximations $\tilde{\Theta}(z) \approx \theta(z)$, $\tilde{\Delta}(z) \approx \delta(z)$ and $\tilde{U}(z) \approx 10^{13} u(z)$. 
References


