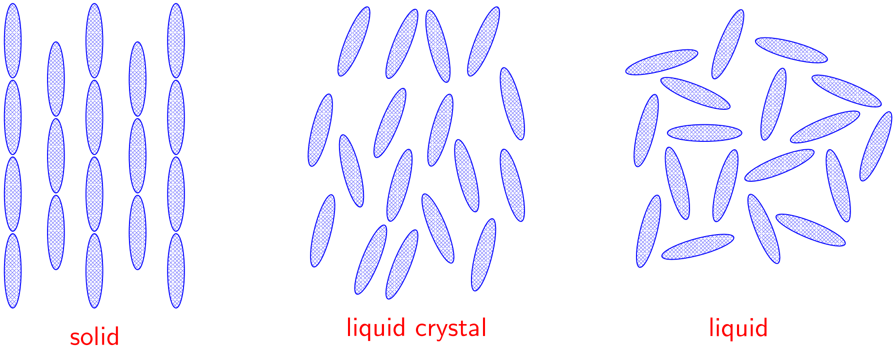
A Moving Mesh Finite Element Method for Modelling Defects in Liquid Crystals

Alison Ramage

Department of Mathematics and Statistics, University of Strathclyde

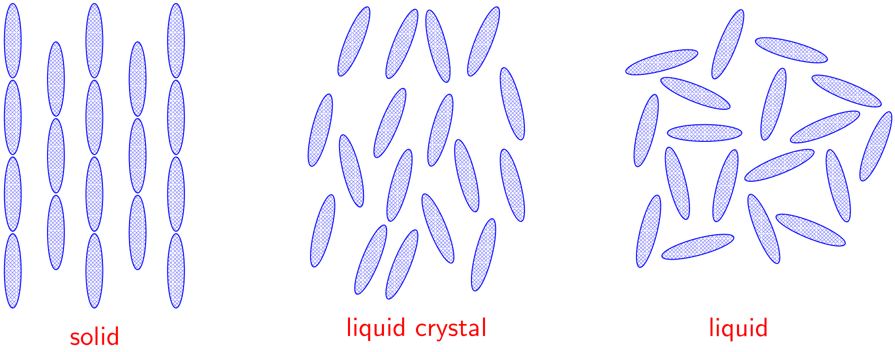
Joint work with Craig MacDonald and John Mackenzie

Liquid crystals



- Liquid crystals occur between solid crystal and isotropic liquid states.

Liquid crystals



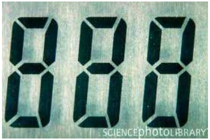
- Liquid crystals occur between solid crystal and isotropic liquid states.
- They may have different equilibrium configurations, but naturally prefer states with minimum energy.

Liquid Crystal Displays

- IDEA: force switching between stable states by altering applied voltage, magnetic field, boundary conditions, ...

Liquid Crystal Displays

- IDEA: force switching between stable states by altering applied voltage, magnetic field, boundary conditions, . . .
- Used in a wide range of LCDs.






Motivation

- Defects in a liquid crystal can arise due to external factors such as applied electric or magnetic fields, or the constraining geometry of the liquid crystal cell.

Motivation

- Defects in a liquid crystal can arise due to external factors such as applied electric or magnetic fields, or the constraining geometry of the liquid crystal cell.
- Understanding the formation and dynamics of defects is important in the design and control of liquid crystal devices.

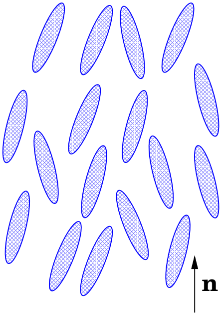
Motivation

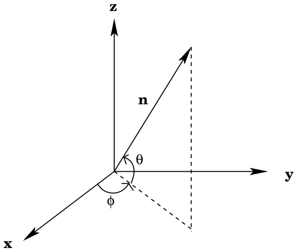
- Defects in a liquid crystal can arise due to external factors such as applied electric or magnetic fields, or the constraining geometry of the liquid crystal cell.
- Understanding the formation and dynamics of defects is important in the design and control of liquid crystal devices.
- Defects typically induce distortion over very small length scales as compared to the size of the cell: this poses significant challenges for standard numerical modelling techniques.

Motivation

- Defects in a liquid crystal can arise due to external factors such as applied electric or magnetic fields, or the constraining geometry of the liquid crystal cell.
- Understanding the formation and dynamics of defects is important in the design and control of liquid crystal devices.
- Defects typically induce distortion over very small length scales as compared to the size of the cell: this poses significant challenges for standard numerical modelling techniques.
- In this talk we present a finite-element based adaptive moving mesh model for tracking defect movement.

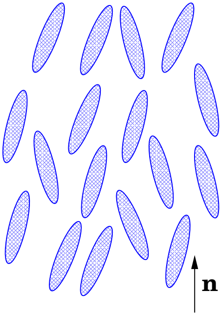
Director-based model

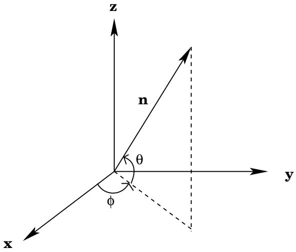




- Director: represents average direction of molecular alignment

Director-based model

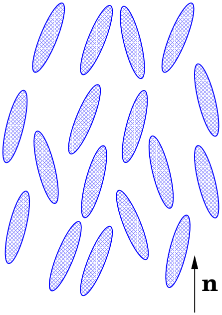


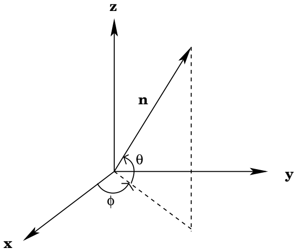


- Director: represents average direction of molecular alignment
- Represent using unit vectors with $\mathbf{n} = -\mathbf{n}$:

$$\mathbf{n} = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)$$

Director-based model





- Director: represents average direction of molecular alignment
- Represent using unit vectors with $\mathbf{n} = -\mathbf{n}$:

$$\mathbf{n} = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)$$

- Model with Leslie-Ericksen dynamic theory.

Liquid crystal model: $\mathbf{Q}$-tensor theory

- Describe orientation of each molecule by a single vector $\mathbf{u}$ in direction of its main axis.

Liquid crystal model: $\mathbf{Q}$-tensor theory

- Describe orientation of each molecule by a single vector $\mathbf{u}$ in direction of its main axis.
- Represent average orientation by symmetric and traceless order tensor

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

- Describe orientation of each molecule by a single vector $\mathbf{u}$ in direction of its main axis.

- Represent average orientation by symmetric and traceless order tensor

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

- With orthogonal eigenframe $\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}$, write

$$\mathbf{Q} = S\left(\mathbf{n} \otimes \mathbf{n} - \frac{1}{3}\mathbf{I}\right) + T(\mathbf{m} \otimes \mathbf{m} - \mathbf{l} \otimes \mathbf{l})$$

where S, T are uniaxial and biaxial order parameters.

Liquid crystal model: $\mathbf{Q}$-tensor theory

- Describe orientation of each molecule by a single vector $\mathbf{u}$ in direction of its main axis.
- Represent average orientation by symmetric and traceless order tensor

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

- With orthogonal eigenframe $\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}$, write

$$\mathbf{Q} = S\left(\mathbf{n} \otimes \mathbf{n} - \frac{1}{3}\mathbf{I}\right) + T(\mathbf{m} \otimes \mathbf{m} - \mathbf{l} \otimes \mathbf{l})$$

  where S, T are uniaxial and biaxial order parameters.
- Consider uniaxial molecular distribution ($T = 0$) where $\mathbf{n}$ is the liquid crystal director.

$\mathbf{Q}$-tensor representation

- Symmetric traceless tensor $\mathbf{Q}$ has five degrees of freedom.

$\mathbf{Q}$-tensor representation

- Symmetric traceless tensor $\mathbf{Q}$ has five degrees of freedom.

- Represent $\mathbf{Q}$ using a (non-unique) basis of five linearly-independent tensors, e.g.

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

$\mathbf{Q}$-tensor representation

- Symmetric traceless tensor $\mathbf{Q}$ has five degrees of freedom.

- Represent $\mathbf{Q}$ using a (non-unique) basis of five linearly-independent tensors, e.g.

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

- Five unknowns for PDE model:

$$q_1, \; q_2, \; q_3, \; q_4, \; q_5.$$

Q-tensor equations

- Minimise the free energy:

$$F = \int_V F_{bulk}(\mathbf{Q}, \nabla\mathbf{Q})\, dv + \int_S F_{surface}(\mathbf{Q})\, dS$$

$$F_{bulk} = F_{elastic} + F_{thermotropic} + F_{electrostatic}$$

Q-tensor equations

- Minimise the free energy:

$$F = \int_V F_{bulk}(\mathbf{Q}, \nabla\mathbf{Q})\, dv + \int_S F_{surface}(\mathbf{Q})\, dS$$

$$F_{bulk} = F_{elastic} + F_{thermotropic} + F_{electrostatic}$$

- Derive expressions for individual energy contributions in terms of $\mathbf{Q}$, $\nabla\mathbf{Q}$.

Q-tensor equations

- Minimise the free energy:

$$F = \int_V F_{bulk}(\mathbf{Q}, \nabla\mathbf{Q})\, dv + \int_S F_{surface}(\mathbf{Q})\, dS$$

$$F_{bulk} = F_{elastic} + F_{thermotropic} + F_{electrostatic}$$

- Derive expressions for individual energy contributions in terms of $\mathbf{Q}$, $\nabla\mathbf{Q}$.

- With strong anchoring (Dirichlet boundary conditions), there is no contribution from surface energy.

Q-tensor equations

- Minimise the free energy:

$$F = \int_V F_{bulk}(\mathbf{Q}, \nabla\mathbf{Q})\, dv + \int_S F_{surface}(\mathbf{Q})\, dS$$

$$F_{bulk} = F_{elastic} + F_{thermotropic} + F_{electrostatic}$$

- Derive expressions for individual energy contributions in terms of $\mathbf{Q}$, $\nabla\mathbf{Q}$.
- With strong anchoring (Dirichlet boundary conditions), there is no contribution from surface energy.
- Solutions with least energy are physically relevant: solve Euler-Lagrange equations.

Bulk energies

- Elastic energy: induced by distorting the **Q**-tensor in space.

$$F_{elastic} = \frac{1}{2}L_1(\mathrm{div}\ \mathbf{Q})^2 + \frac{1}{2}L_2|\nabla \times \mathbf{Q}|^2$$

Bulk energies

- Elastic energy: induced by distorting the **Q**-tensor in space.

$$F_{elastic} = \frac{1}{2}L_1(\mathrm{div}\ \mathbf{Q})^2 + \frac{1}{2}L_2|\nabla \times \mathbf{Q}|^2$$

- Thermotropic energy: potential function which dictates which preferred state (uniaxial, biaxial or isotropic).

$$F_{thermotropic} = \frac{1}{2}A(T - T^*)\ \mathrm{tr}\ \mathbf{Q}^2 - \frac{\sqrt{6}}{3}B\ \mathrm{tr}\ \mathbf{Q}^3 + \frac{1}{4}C(\mathrm{tr}\ \mathbf{Q}^2)^2$$

Bulk energies

- Elastic energy: induced by distorting the **Q**-tensor in space.

$$F_{elastic} = \frac{1}{2}L_1(\mathrm{div}\ \mathbf{Q})^2 + \frac{1}{2}L_2|\nabla \times \mathbf{Q}|^2$$

- Thermotropic energy: potential function which dictates which preferred state (uniaxial, biaxial or isotropic).

$$F_{thermotropic} = \frac{1}{2}A(T - T^*)\ \mathrm{tr}\ \mathbf{Q}^2 - \frac{\sqrt{6}}{3}B\ \mathrm{tr}\ \mathbf{Q}^3 + \frac{1}{4}C(\mathrm{tr}\ \mathbf{Q}^2)^2$$

- Electrostatic energy: due to an applied electric field **E** (electric potential U with $\mathbf{E} = -\nabla U$).

$$F_{electrostatic} = -\frac{1}{2}\epsilon_0\mathbf{E}\cdot\boldsymbol{\epsilon}\mathbf{E} - (\bar{e}\ \mathrm{div}\ \mathbf{Q})\cdot\mathbf{E}$$

Derivation of time-dependent PDEs

- Use a dissipation function with viscosity coefficient ν.

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

Derivation of time-dependent PDEs

- Use a dissipation function with viscosity coefficient ν.

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

- Obtain **Q**-tensor PDEs (for $i = 1, \ldots, 5$ and $j = 1, 2, 3$):

$$\frac{\partial \mathcal{D}}{\partial \dot{q}_i} = \nabla \cdot \hat{\boldsymbol{\Gamma}}_i - \hat{f}_i$$

$$(\hat{\boldsymbol{\Gamma}}_i)_j = \frac{\partial F_{bulk}}{\partial q_{i,j}}, \qquad q_{i,j} = \frac{\partial q_i}{\partial x_j}, \qquad \hat{f}_i = \frac{\partial F_{bulk}}{\partial q_i}$$

Derivation of time-dependent PDEs

- Use a dissipation function with viscosity coefficient ν.

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

- Obtain **Q**-tensor PDEs (for $i = 1, \ldots, 5$ and $j = 1, 2, 3$):

$$\frac{\partial \mathcal{D}}{\partial \dot{q}_i} = \nabla \cdot \hat{\mathbf{\Gamma}}_i - \hat{f}_i$$

$$(\hat{\mathbf{\Gamma}}_i)_j = \frac{\partial F_{bulk}}{\partial q_{i,j}}, \qquad q_{i,j} = \frac{\partial q_i}{\partial x_j}, \qquad \hat{f}_i = \frac{\partial F_{bulk}}{\partial q_i}$$

- Combining equations and manipulating terms gives

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

Coupling with electric field

- Additional unknown U such that $\mathbf{E} = -\nabla U$.

Coupling with electric field

- Additional unknown U such that $\mathbf{E} = -\nabla U$.
- Assuming no free charges, solve the Maxwell equation

$$\nabla \cdot \mathbf{D} = 0, \qquad \text{electric displacement } \mathbf{D}.$$

Coupling with electric field

- Additional unknown U such that $\mathbf{E} = -\nabla U$.
- Assuming no free charges, solve the Maxwell equation

$$\nabla \cdot \mathbf{D} = 0, \qquad \text{electric displacement } \mathbf{D}.$$

SUMMARY

- Final time-dependent physical PDEs (PPDEs) are

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

$$\nabla \cdot \mathbf{D} = 0$$

- 6 PDEs in 6 unknowns $(q_1, q_2, q_3, q_4, q_5, U)$

Adaptive finite element methods

- Three common forms of grid adaptivity in finite elements:
  - *h*-refinement: initially uniform mesh is locally coarsened or refined by inclusion or deletion of mesh points, normally based on *a posteriori* error estimates
  - *p*-refinement: order of local polynomial approximation is increased or decreased in accordance with solution error
  - *r*-refinement: original mesh points are moved to areas where high resolution is needed

Adaptive finite element methods

- Three common forms of grid adaptivity in finite elements:
  - *h*-refinement: initially uniform mesh is locally coarsened or refined by inclusion or deletion of mesh points, normally based on *a posteriori* error estimates
  - *p*-refinement: order of local polynomial approximation is increased or decreased in accordance with solution error
  - *r*-refinement: original mesh points are moved to areas where high resolution is needed

- Advantages of moving meshes:
  - retaining fixed number of mesh points and connectivity;
  - interpolation from old to new mesh unnecessary for time-dependent problems.

Adaptive finite element methods

- Three common forms of grid adaptivity in finite elements:
  - *h*-refinement: initially uniform mesh is locally coarsened or refined by inclusion or deletion of mesh points, normally based on *a posteriori* error estimates
  - *p*-refinement: order of local polynomial approximation is increased or decreased in accordance with solution error
  - *r*-refinement: original mesh points are moved to areas where high resolution is needed

- Advantages of moving meshes:
  - retaining fixed number of mesh points and connectivity;
  - interpolation from old to new mesh unnecessary for time-dependent problems.

- Focus here on Moving Mesh PDE model.

Adapt PPDEs for mesh movement

- Physical domain Ω, computational domain Ω_c.

Adapt PPDEs for mesh movement

- Physical domain Ω, computational domain Ω_c.
- Bijective mappings $\mathcal{A}_t : \Omega_c \to \Omega$ map $\boldsymbol{\xi} = (\xi, \eta) \subset \Omega_c$ to $\mathbf{x} = (x, y) \subset \Omega$:

$$\mathbf{x}(\boldsymbol{\xi}, t) = \mathcal{A}_t(\boldsymbol{\xi}) \ .$$

Adapt PPDEs for mesh movement

- Physical domain Ω, computational domain Ω_c.
- Bijective mappings $\mathcal{A}_t : \Omega_c \to \Omega$ map $\boldsymbol{\xi} = (\xi, \eta) \subset \Omega_c$ to $\mathbf{x} = (x, y) \subset \Omega$:

$$\mathbf{x}(\boldsymbol{\xi}, t) = \mathcal{A}_t(\boldsymbol{\xi}) \ .$$

- Define mesh velocity

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

and apply the Chain Rule to get

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

Adapt PPDEs for mesh movement

- Physical domain Ω, computational domain Ω_c.
- Bijective mappings $\mathcal{A}_t : \Omega_c \to \Omega$ map $\boldsymbol{\xi} = (\xi, \eta) \subset \Omega_c$ to $\mathbf{x} = (x, y) \subset \Omega$:

$$\mathbf{x}(\boldsymbol{\xi}, t) = \mathcal{A}_t(\boldsymbol{\xi}) \ .$$

- Define mesh velocity

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

and apply the Chain Rule to get

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

- Additional convection-like term due to the mesh movement.

Finite elements for the physical PDEs

- Final set of six coupled PDEs ($i = 1, \ldots, 5$):

$$\left.\frac{\partial q_i}{\partial t}\right|_{\xi} - \dot{\mathbf{x}} \cdot \nabla q = \nabla \cdot \Gamma_i - f_i, \qquad \nabla \cdot \mathbf{D} = 0$$

Finite elements for the physical PDEs

- Final set of six coupled PDEs ($i = 1, \ldots, 5$):

$$\left.\frac{\partial q_i}{\partial t}\right|_{\xi} - \dot{\mathbf{x}} \cdot \nabla q = \nabla \cdot \Gamma_i - f_i, \qquad \nabla \cdot \mathbf{D} = 0$$

- Find $q_{ih}(t)$, U_h such that for test functions v_h

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

$$\int_{\Omega} \mathbf{D}_h \cdot \nabla v_h \,\mathrm{d}\mathbf{x} = 0.$$

Finite elements for the physical PDEs

- Final set of six coupled PDEs ($i = 1, \ldots, 5$):

$$\left.\frac{\partial q_i}{\partial t}\right|_{\xi} - \dot{\mathbf{x}} \cdot \nabla q = \nabla \cdot \Gamma_i - f_i, \qquad \nabla \cdot \mathbf{D} = 0$$

- Find $q_{ih}(t)$, U_h such that for test functions v_h

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

$$\int_\Omega \mathbf{D}_h \cdot \nabla v_h \,\mathrm{d}\mathbf{x} = 0.$$

- Non-linear differential algebraic system ($i = 1, \ldots, 5$)

$$\frac{d}{dt}(M(t)\mathbf{q}_i(t)) = \mathbf{G}_i(t, \mathbf{q}_i(t), \mathbf{u}(t)), \qquad \mathbf{C}(\mathbf{q}_i(t), \mathbf{u}(t)) = \mathbf{0}.$$

Moving Mesh PDEs

- Avoid mesh crossings by evolving inverse mapping

$$\mathcal{A}_t^{-1}(\mathbf{x}) = \xi(\mathbf{x}, t).$$

Moving Mesh PDEs

- Avoid mesh crossings by evolving inverse mapping

$$\mathcal{A}_t^{-1}(\mathbf{x}) = \xi(\mathbf{x}, t).$$

- Choose mapping $\xi(\mathbf{x})$ for a fixed t to minimise

$$I[\xi] = \frac{1}{2}\int_{\Omega_t} \left[(\nabla\xi)^T G^{-1}(\nabla\xi) + (\nabla\eta)^T G^{-1}(\nabla\eta)\right] \mathrm{d}\mathbf{x}$$

with 2×2 symmetric positive definite monitor matrix G.

Moving Mesh PDEs

- Avoid mesh crossings by evolving inverse mapping

$$\mathcal{A}_t^{-1}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x}, t).$$

- Choose mapping $\boldsymbol{\xi}(\mathbf{x})$ for a fixed t to minimise

$$I[\boldsymbol{\xi}] = \frac{1}{2}\int_{\Omega_t} [(\nabla\xi)^T G^{-1}(\nabla\xi) + (\nabla\eta)^T G^{-1}(\nabla\eta)]\,\mathrm{d}\mathbf{x}$$

with 2×2 symmetric positive definite monitor matrix G.

- For robustness, evolve mesh via gradient flow equations

$$\frac{\partial\xi}{\partial t} = \frac{P}{\tau}\nabla\cdot(G^{-1}\nabla\xi), \qquad \frac{\partial\eta}{\partial t} = \frac{P}{\tau}\nabla\cdot(G^{-1}\nabla\eta).$$

Moving Mesh PDEs

- Avoid mesh crossings by evolving inverse mapping

$$\mathcal{A}_t^{-1}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x}, t).$$

- Choose mapping $\boldsymbol{\xi}(\mathbf{x})$ for a fixed t to minimise

$$I[\boldsymbol{\xi}] = \frac{1}{2}\int_{\Omega_t} \left[(\nabla\xi)^T G^{-1}(\nabla\xi) + (\nabla\eta)^T G^{-1}(\nabla\eta)\right] \mathrm{d}\mathbf{x}$$

  with 2×2 symmetric positive definite monitor matrix G.

- For robustness, evolve mesh via gradient flow equations

$$\frac{\partial\xi}{\partial t} = \frac{P}{\tau}\nabla\cdot(G^{-1}\nabla\xi), \qquad \frac{\partial\eta}{\partial t} = \frac{P}{\tau}\nabla\cdot(G^{-1}\nabla\eta).$$

- User-specified parameters:
  - positive temporal smoothing parameter τ,
  - positive function spatial balancing parameter $P(\mathbf{x}, t)$.

Final form of MMPDE

- Use monitor function $w(\mathbf{x}, t)$ with Winslow monitor matrix

$$G = \begin{bmatrix} w & 0 \\ 0 & w \end{bmatrix}.$$

Final form of MMPDE

- Use monitor function $w(\mathbf{x}, t)$ with Winslow monitor matrix

$$G = \begin{bmatrix} w & 0 \\ 0 & w \end{bmatrix}.$$

- In practice, interchange variable roles in MMPDE to obtain

$$\tau \frac{\partial \mathbf{x}}{\partial t} = P(a\mathbf{x}_{\xi\xi} + b\mathbf{x}_{\xi\eta} + c\mathbf{x}_{\eta\eta} + d\mathbf{x}_{\xi} + e\mathbf{x}_{\eta})$$

$$a = \frac{1}{w}\frac{x_\eta^2 + y_\eta^2}{J^2}, \quad b = -\frac{2}{w}\frac{(x_\xi x_\eta + y_\xi y_\eta)}{J^2}, \quad c = \frac{1}{w}\frac{x_\xi^2 + y_\xi^2}{J^2},$$

$$d = \frac{1}{(wJ)^2}[w_\xi(x_\eta^2 + y_\eta^2) - w_\eta(x_\xi x_\eta + y_\xi y_\eta),$$

$$e = \frac{1}{(wJ)^2}\left[-w_\xi(x_\xi x_\eta + y_\xi y_\eta) + w_\eta(x_\xi^2 + y_\xi^2)\right].$$

Additional details for MMPDE

- Boundary conditions obtained using a 1D MMPDE.

Additional details for MMPDE

- Boundary conditions obtained using a 1D MMPDE.
- Discretise in space using linear finite elements.
- Discretise in time using a backward Euler scheme.

Additional details for MMPDE

- Boundary conditions obtained using a 1D MMPDE.
- Discretise in space using linear finite elements.
- Discretise in time using a backward Euler scheme.
- To avoid solving nonlinear algebraic systems, at $t = t^{n+1}$ evaluate coefficients a, b, c, d, e at the time $t = t^n$.

Additional details for MMPDE

- Boundary conditions obtained using a 1D MMPDE.
- Discretise in space using linear finite elements.
- Discretise in time using a backward Euler scheme.
- To avoid solving nonlinear algebraic systems, at $t = t^{n+1}$ evaluate coefficients a, b, c, d, e at the time $t = t^n$.
- Solve resulting linear systems using iterative method BiCGSTAB with Incomplete LU preconditioner.

Additional details for MMPDE

- Boundary conditions obtained using a 1D MMPDE.
- Discretise in space using linear finite elements.
- Discretise in time using a backward Euler scheme.
- To avoid solving nonlinear algebraic systems, at $t = t^{n+1}$ evaluate coefficients a, b, c, d, e at the time $t = t^n$.
- Solve resulting linear systems using iterative method BiCGSTAB with Incomplete LU preconditioner.
- Adaptive time-stepping based on computed solutions of PPDEs and MMPDE.

Overview of full algorithm

Set an initial uniform mesh Δ_N^0. Set the initial guess $\mathbf{q}_i^0$.
Select an initial Δt^0. Set $n = 0$.
while $(t^n < t^{\max})$;
    Evaluate monitor function at time t^n.
    Integrate MMPDE forward in time to obtain new grid Δ_N^{n+1}.
    Integrate PPDEs forward using SDIRK2 to obtain $\mathbf{q}_i^{n+1}$, $\mathbf{u}^{n+1}$.
    $n := n + 1$.
end while.

Choosing the monitor function

- Consider three different forms of monitor function:
  - AL. Based on a measure of the **arc-length** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \left(1 + |\nabla\mathcal{T}(\mathbf{x},t)|^2\right)^{\frac{1}{2}}$$

Choosing the monitor function

- Consider three different forms of monitor function:
  - AL. Based on a measure of the **arc-length** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \left(1 + |\nabla \mathcal{T}(\mathbf{x},t)|^2\right)^{\frac{1}{2}}$$

  - BM1. Based on **first-order partial derivatives** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \alpha(\mathbf{x},t) + |\nabla \mathcal{T}(\mathbf{x},t)|^{\frac{1}{m}}$$

Choosing the monitor function

- Consider three different forms of monitor function:
  - AL. Based on a measure of the **arc-length** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \left(1 + |\nabla\mathcal{T}(\mathbf{x},t)|^2\right)^{\frac{1}{2}}$$

  - BM1. Based on **first-order partial derivatives** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \alpha(\mathbf{x},t) + |\nabla\mathcal{T}(\mathbf{x},t)|^{\frac{1}{m}}$$

  - BM2. Based on **second-order partial derivatives** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \alpha(\mathbf{x},t) + \left(\sqrt{\left(\frac{\partial^2\mathcal{T}}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2\mathcal{T}}{\partial x\partial y}\right)^2 + \left(\frac{\partial^2\mathcal{T}}{\partial y^2}\right)^2}\right)^{\frac{1}{m}}$$

Choosing the monitor function

- Consider three different forms of monitor function:
  - AL. Based on a measure of the **arc-length** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \left(1 + |\nabla\mathcal{T}(\mathbf{x},t)|^2\right)^{\frac{1}{2}}$$

  - BM1. Based on **first-order partial derivatives** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \alpha(\mathbf{x},t) + |\nabla\mathcal{T}(\mathbf{x},t)|^{\frac{1}{m}}$$

  - BM2. Based on **second-order partial derivatives** of $\mathcal{T}$:

$$w(\mathcal{T}(\mathbf{x},t)) = \alpha(\mathbf{x},t) + \left(\sqrt{\left(\frac{\partial^2\mathcal{T}}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2\mathcal{T}}{\partial x\partial y}\right)^2 + \left(\frac{\partial^2\mathcal{T}}{\partial y^2}\right)^2}\right)^{\frac{1}{m}}$$

- Scaling parameters α and m regulate mesh clustering.

Choosing the input function

- Consider two different forms of input function:
  - Scalar order parameter. Based on the trace of $\mathbf{Q}^2$

$$\mathcal{T}(\mathbf{x}, t) = \mathrm{tr}(\mathbf{Q}^2)$$

as $\mathrm{tr}(\mathbf{Q}^2) = S^2$ for uniaxial state with order parameter S.

Choosing the input function

- Consider two different forms of input function:
  - Scalar order parameter. Based on the trace of $\mathbf{Q}^2$

$$\mathcal{T}(\mathbf{x}, t) = \mathrm{tr}(\mathbf{Q}^2)$$

    as $\mathrm{tr}(\mathbf{Q}^2) = S^2$ for uniaxial state with order parameter S.

  - Biaxiality. Based on a direct invariant measure of biaxiality

$$\mathcal{T}(\mathbf{x}, t) = \left[1 - \frac{6\,\mathrm{tr}(\mathbf{Q}^3)^2}{\mathrm{tr}(\mathbf{Q}^2)^3}\right]^{\frac{1}{2}}$$

    taking values ranging from 0 (uniaxial) to 1 (fully biaxial).

Choosing the input function

- Consider two different forms of input function:
  - Scalar order parameter. Based on the trace of $\mathbf{Q}^2$

    $$\mathcal{T}(\mathbf{x}, t) = \mathrm{tr}(\mathbf{Q}^2)$$

    as $\mathrm{tr}(\mathbf{Q}^2) = S^2$ for uniaxial state with order parameter S.

  - Biaxiality. Based on a direct invariant measure of biaxiality

    $$\mathcal{T}(\mathbf{x}, t) = \left[1 - \frac{6\,\mathrm{tr}(\mathbf{Q}^3)^2}{\mathrm{tr}(\mathbf{Q}^2)^3}\right]^{\frac{1}{2}}$$

    taking values ranging from 0 (uniaxial) to 1 (fully biaxial).

- Both have extrema at the centre of a defect and vary rapidly in the immediate neighbourhood of the defect centre.

Numerical experiments

- PPDEs non-dimensionalised with respect to lengths and energies.

Numerical experiments

- PPDEs non-dimensionalised with respect to lengths and energies.
- Use quadratic triangular finite elements for PPDEs, linear finite elements for MMPDE.

Numerical experiments

- PPDEs non-dimensionalised with respect to lengths and energies.
- Use quadratic triangular finite elements for PPDEs, linear finite elements for MMPDE.
- Monitor/input function combinations:

Method name	AL	BM1a	BM1b	BM2b
Monitor function	AL	BM1	BM1	BM2
Input function	$\mathrm{tr}(\mathbf{Q}^2)$	$\mathrm{tr}(\mathbf{Q}^2)$	biaxiality	biaxiality

Numerical experiments

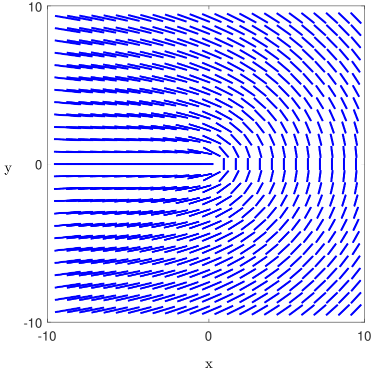
- PPDEs non-dimensionalised with respect to lengths and energies.
- Use quadratic triangular finite elements for PPDEs, linear finite elements for MMPDE.
- Monitor/input function combinations:

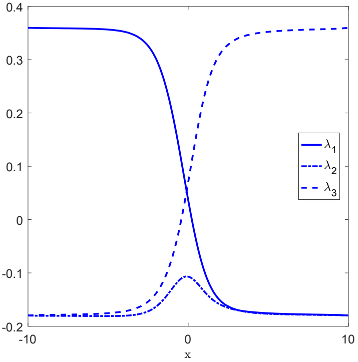
Method name	AL	BM1a	BM1b	BM2b
Monitor function	AL	BM1	BM1	BM2
Input function	$\mathrm{tr}(\mathbf{Q}^2)$	$\mathrm{tr}(\mathbf{Q}^2)$	biaxiality	biaxiality

- All experiments carried out in MATLAB.

Test problem 1: stationary defect

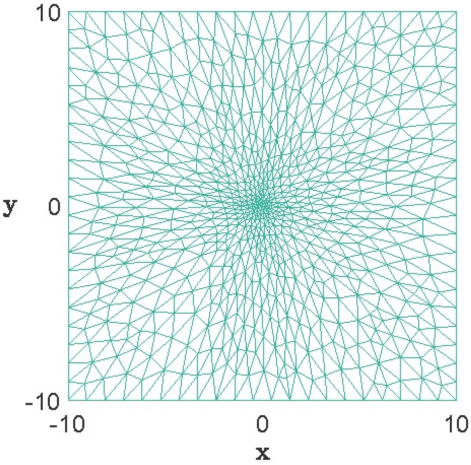
- Director field of 1/2 defect and eigenvalue exchange along the line $y = 0$.





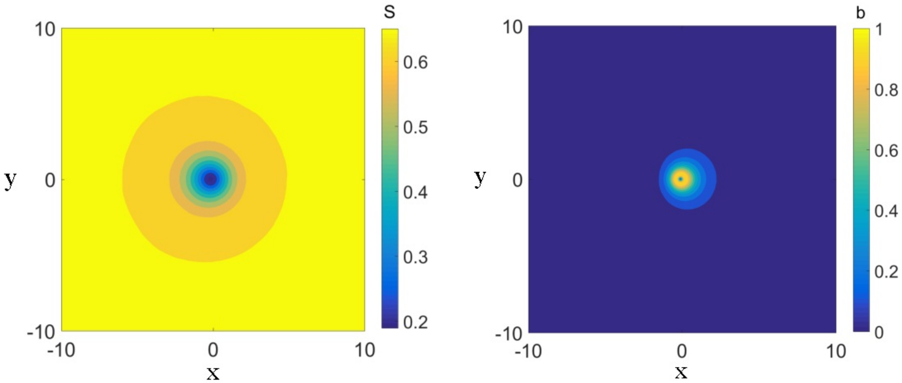
Typical adapted grid

- Sample adapted grid with 1388 quadratic elements.



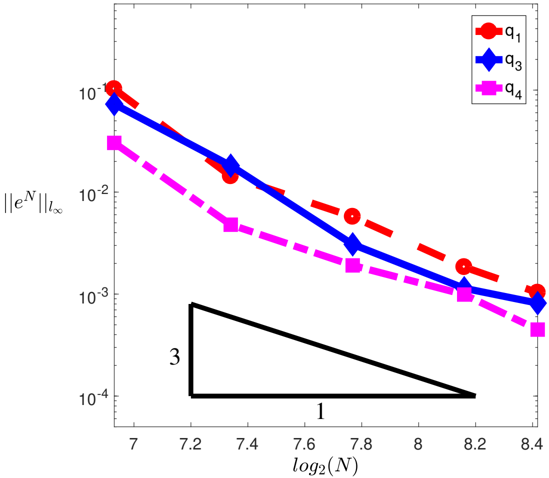
Typical solutions

- Scalar order parameter S (left) and biaxiality (right).



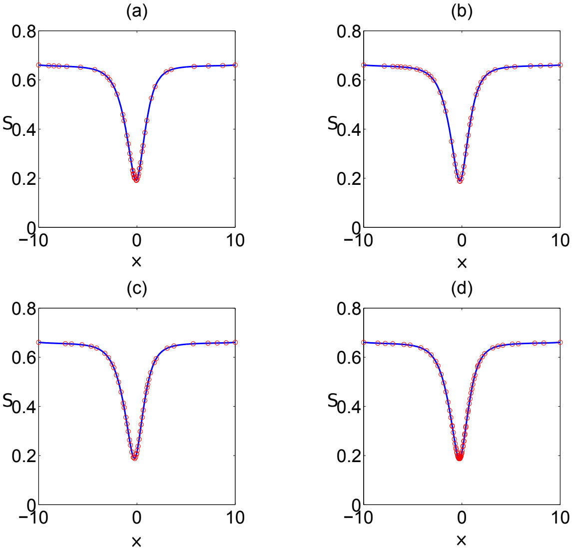
Estimated rate of spatial convergence

- ℓ_∞ error compared with reference solution is $O(N^{-3})$.



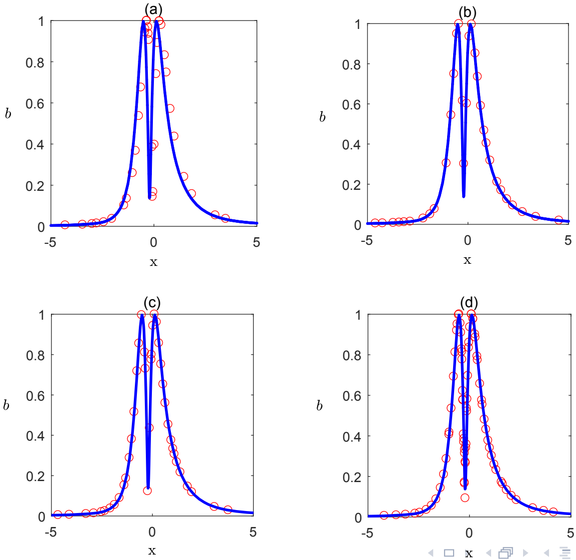
Scalar order parameter along line $y = 0$

- (a) AL; (b) BM1a; (c) BM1b; (d) BM2b

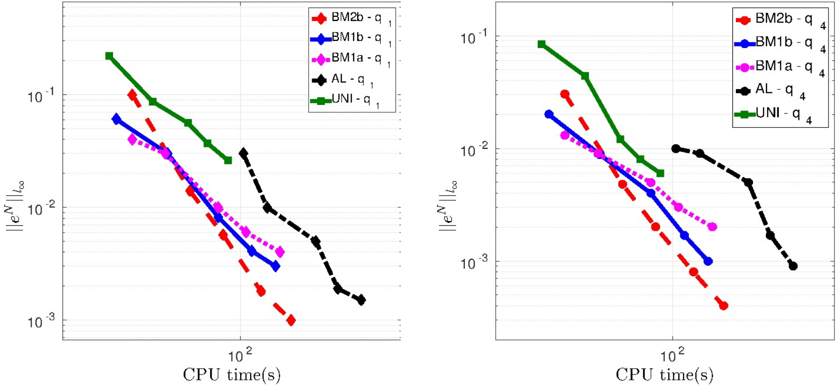


Biaxiality along line $y = 0$

- (a) AL; (b) BM1a; (c) BM1b; (d) BM2b

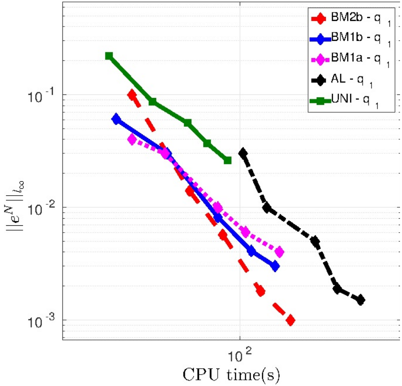


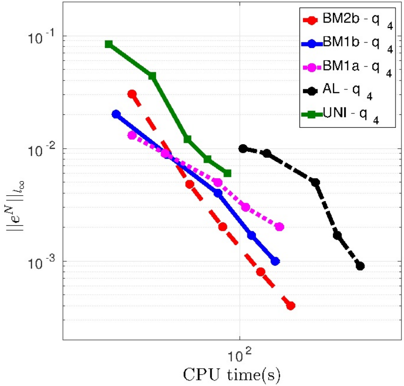
- CPU time versus ℓ_∞ error for different grid sizes.



Comparing computational costs

- CPU time versus ℓ_∞ error for different grid sizes.





- BM2b established as combination of choice.

Test problem 2: 2D Pi-cell

- Two-dimensional Pi-cell geometry.

  Zhang, Chung, Wang and Bos, *Liquid Crystals* 34(2), 2007

Test problem 2: 2D Pi-cell

- Two-dimensional Pi-cell geometry.
  Zhang, Chung, Wang and Bos, *Liquid Crystals* 34(2), 2007

- Electric field applied parallel to the cell thickness at time $t = 0$.

Test problem 2: 2D Pi-cell

- Two-dimensional Pi-cell geometry.
  Zhang, Chung, Wang and Bos, *Liquid Crystals* 34(2), 2007
- Electric field applied parallel to the cell thickness at time $t = 0$.
- Inhomogeneous transition mediated by the nucleation of defect pairs moving and annihilating each other.

Test problem 2: 2D Pi-cell

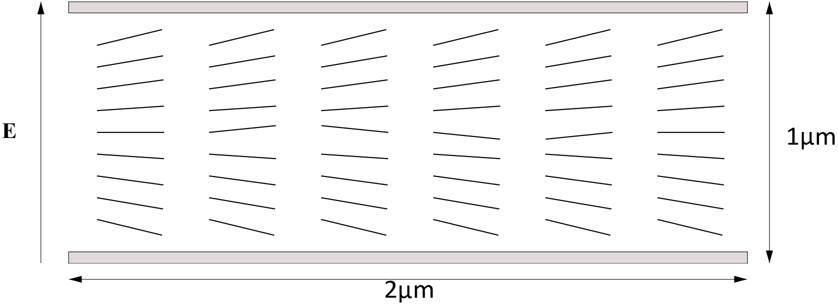
- Two-dimensional Pi-cell geometry.
  Zhang, Chung, Wang and Bos, *Liquid Crystals* 34(2), 2007
- Electric field applied parallel to the cell thickness at time $t = 0$.
- Inhomogeneous transition mediated by the nucleation of defect pairs moving and annihilating each other.
- Initial director angle across cell centre follows $\sin(2\pi x/p)$ for cell width p.

Test problem 2: 2D Pi-cell

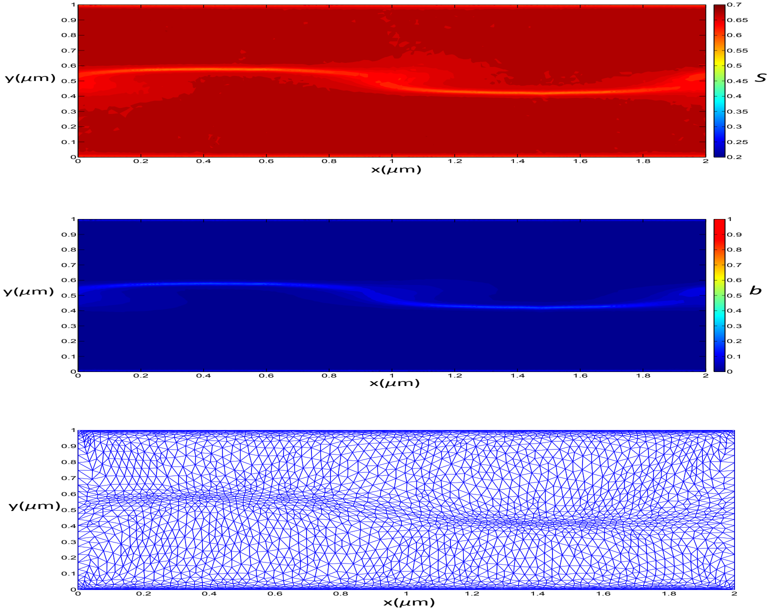
- Two-dimensional Pi-cell geometry.
  Zhang, Chung, Wang and Bos, *Liquid Crystals* 34(2), 2007
- Electric field applied parallel to the cell thickness at time $t = 0$.
- Inhomogeneous transition mediated by the nucleation of defect pairs moving and annihilating each other.
- Initial director angle across cell centre follows $\sin(2\pi x/p)$ for cell width p.
- Perturbation fixed only at $t = 0$ for one time step, but introduces solution gradients in two dimensions.

Pi-cell geometry

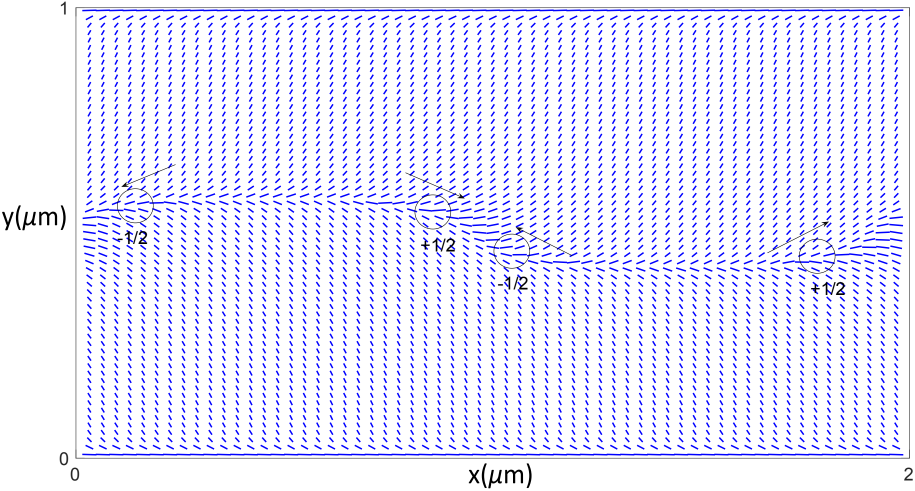
- Pre-tilt angle $\theta = \pm 6^{\circ}$ at boundaries.
- Electric field strength $18V\mu\mathrm{m}^{-1}$.



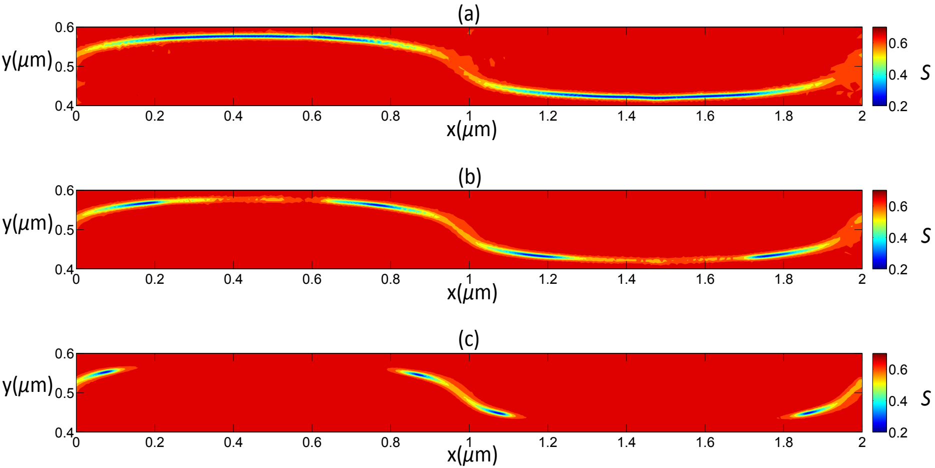
S, biaxiality and mesh after 12μs



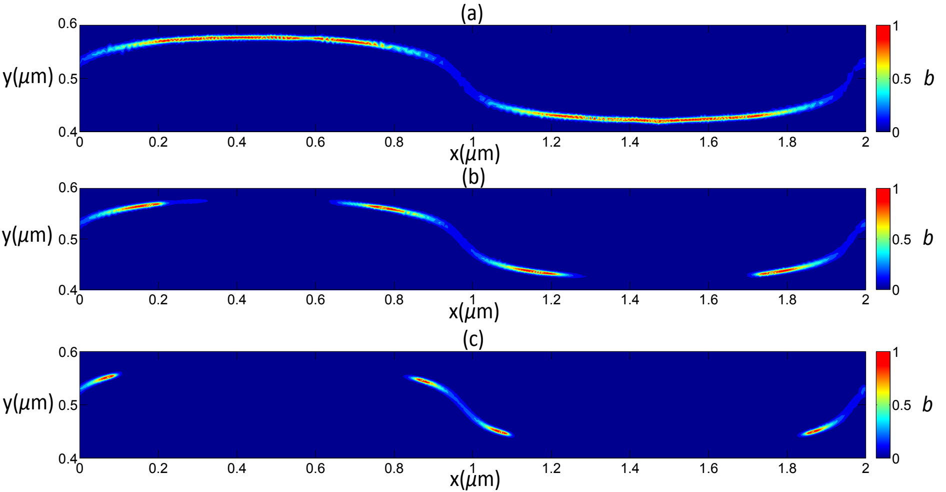
Director field after 15.5μs



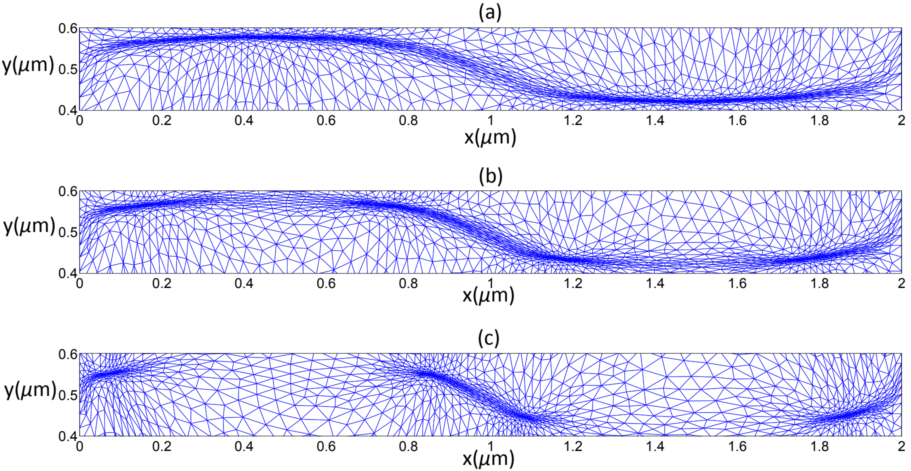
Order parameter S after (a) 15.5μs (b) 16μs and (c) 17μs



Biaxiality after (a) 15.5μs (b) 16μs and (c) 17μs



Adaptive mesh after (a) 15.5μs (b) 16μs and (c) 17μs



Summary and future work

- We have developed a new efficient moving mesh method for $\mathbf{Q}$-tensor models of liquid crystal cells.
- We have shown that biaxiality is a good choice for the monitor input function.
- We demonstrated optimal spatial convergence for a model of a static $+1/2$ defect.
- We resolved the movement and core details of defects in a time-dependent Pi-cell problem.
- Modelling the creation and annihilation of moving singularities on very small length and time scales is a real challenge for numerical methods.
- Future challenges involve the extension to three dimensions and more irregular geometries (e.g. the ZBD).