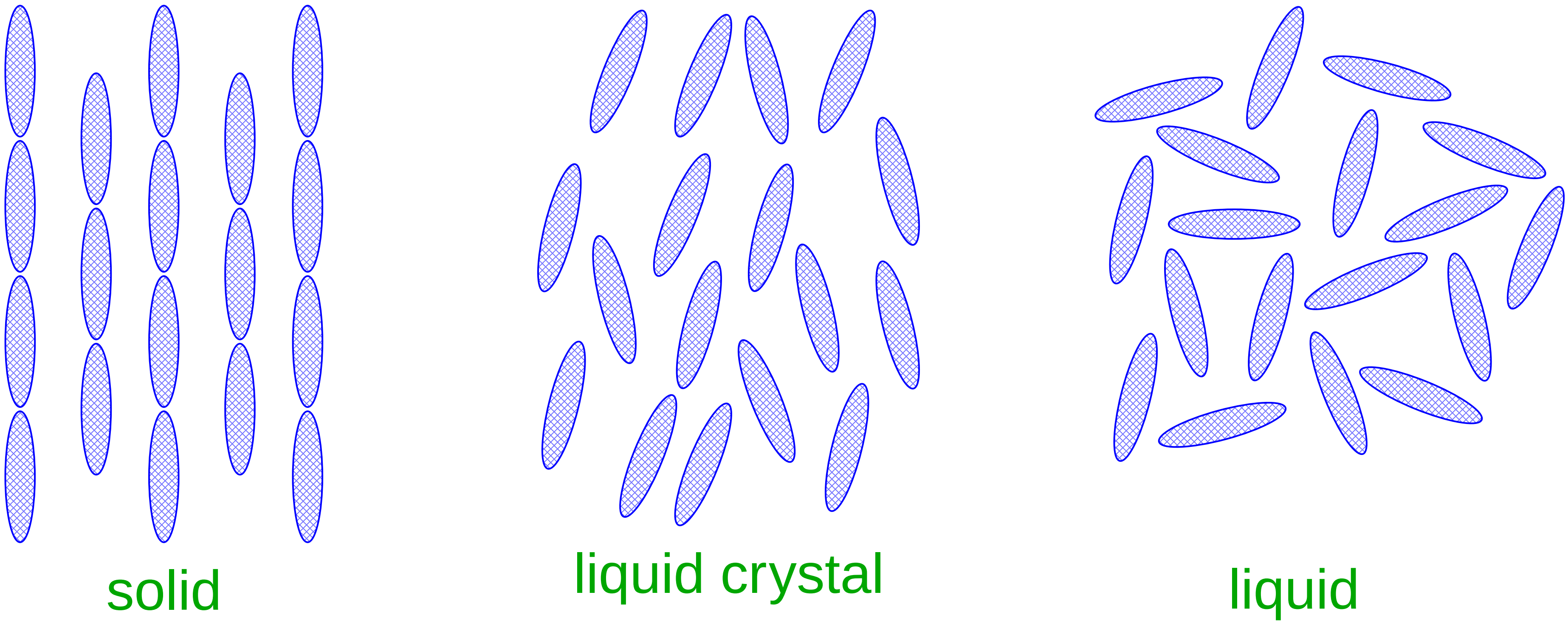
# Efficient iterative solvers for director-based models of LCDs

Alison Ramage
Mathematics and Statistics
University of Strathclyde
Glasgow, Scotland

Eugene C. Gartland, Jr.
Mathematics
Kent State University
Ohio, USA

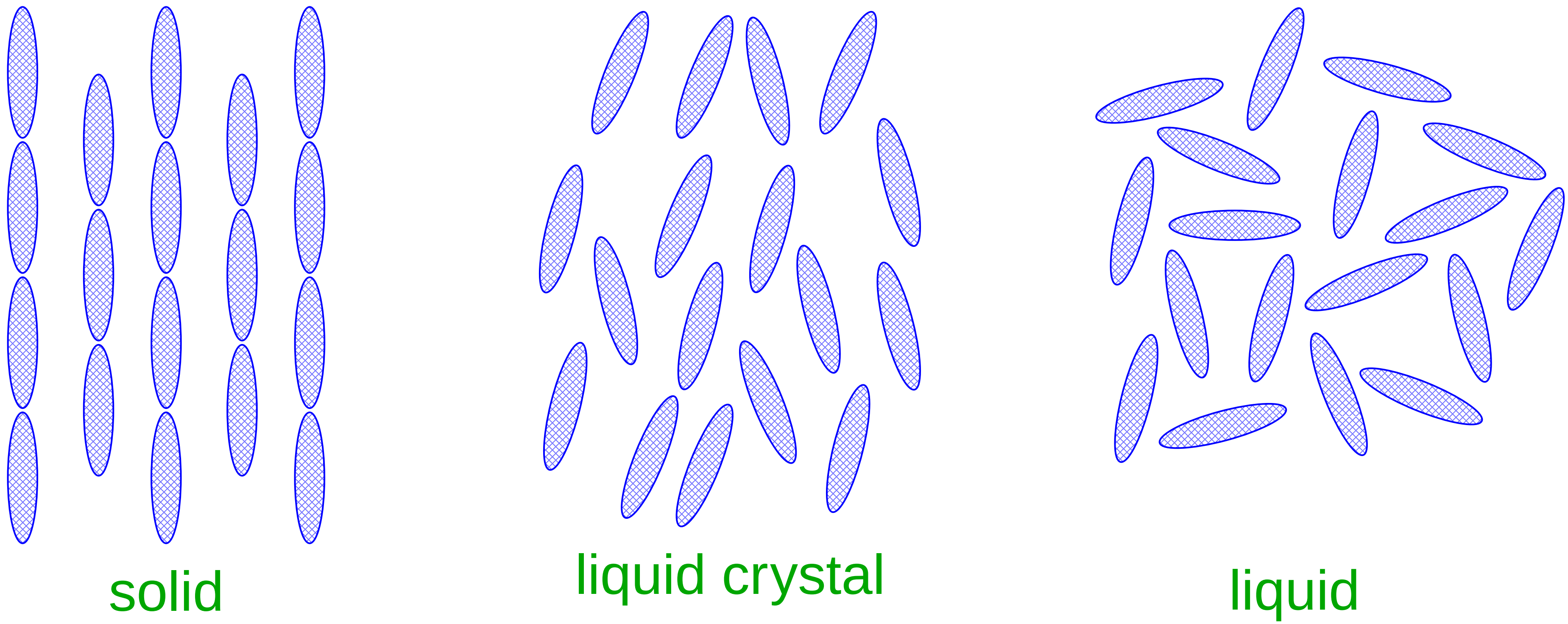
# Liquid Crystals

- occur between solid crystal and isotropic liquid states



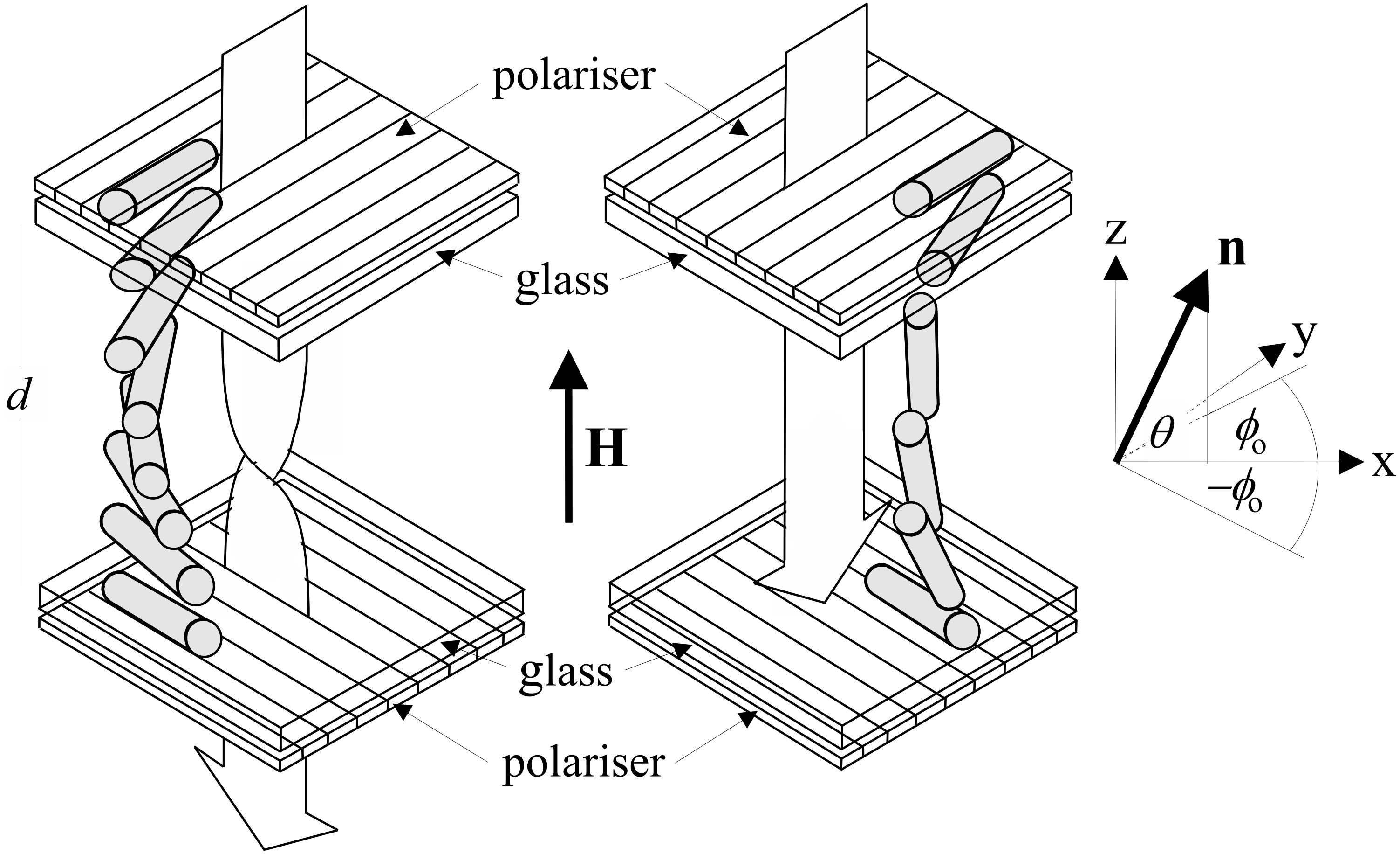
# Liquid Crystals

- occur between solid crystal and isotropic liquid states




- may have different equilibrium configurations
- switch between stable states by altering applied voltage, magnetic field, boundary conditions, . . .

# Liquid Crystal Displays

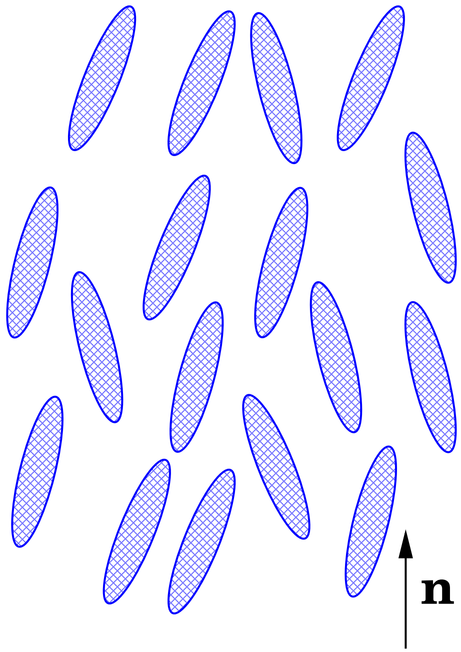


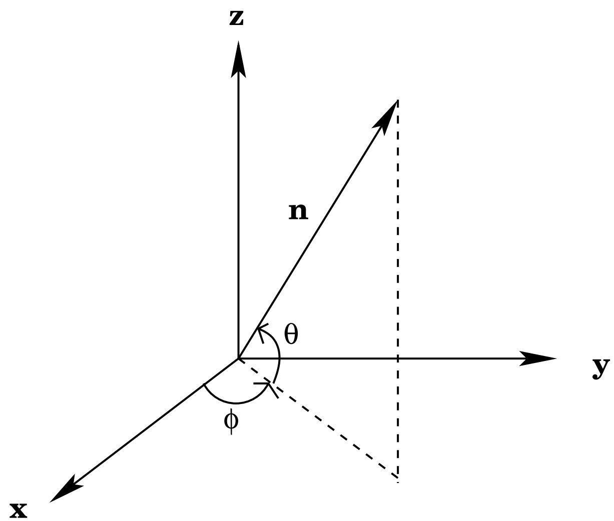

(a) 'off state' $H < H_c$ (b) 'on state' $H > H_c$

twisted nematic device

*Static and Dynamic Continuum Theory of Liquid Crystals,*
*Iain W. Stewart (2004)*

# Modelling: Director-based Models




- director: average direction of molecular alignment

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

- order parameter: measure of orientational order

$$S = \frac{1}{2} < 3\cos^2\theta_m - 1 >$$

# Alternative Model: $\mathrm{Q}$-tensor Theory

- tensor order parameter

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

- symmetric tensor

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

$$tr(Q) = 0, \qquad tr(Q^2) = S^2$$

- five unknowns $\quad q_1, q_2, q_3, q_4, q_5$

# Finding Equilibrium Configurations

- minimise the free energy

$$\mathcal{F} = \int_V F_{bulk}(\theta, \phi, \nabla\theta, \nabla\phi) + \int_{\mathcal{S}} F_{surface}(\theta, \phi)\, d\mathcal{S}$$

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

# Finding Equilibrium Configurations

- minimise the free energy

$$\mathcal{F} = \int_V F_{bulk}(\theta, \phi, \nabla\theta, \nabla\phi) + \int_{\mathcal{S}} F_{surface}(\theta, \phi)\, d\mathcal{S}$$

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

- if fixed boundary conditions are applied, surface energy term can be ignored

# Finding Equilibrium Configurations

- minimise the free energy

$$\mathcal{F} = \int_V F_{bulk}(\theta, \phi, \nabla\theta, \nabla\phi) + \int_{\mathcal{S}} F_{surface}(\theta, \phi)\, d\mathcal{S}$$

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

- if fixed boundary conditions are applied, surface energy term can be ignored

- solutions with least energy are physically relevant

# Elastic Energy

- Frank-Oseen elastic energy

$$
\begin{aligned}
F_{elastic} &= \frac{1}{2}K_1(\nabla\cdot\mathbf{n})^2 + \frac{1}{2}K_2(\mathbf{n}\cdot\nabla\times\mathbf{n})^2 \\
&+ \frac{1}{2}K_3(\mathbf{n}\times\nabla\times\mathbf{n})^2 \\
&+ \frac{1}{2}(K_2+K_4)\nabla\cdot[(\mathbf{n}\cdot\nabla)\mathbf{n}-(\nabla\cdot\mathbf{n})\mathbf{n}]
\end{aligned}
$$

- Frank elastic constants

| | |
|---|---|
| $K_1$ | splay |
| $K_2$ | twist |
| $K_3$ | bend |
| $K_2 + K_4$ | saddle-splay |

# One-Constant Approximation

- set

$$K = K_1 = K_2 = K_3, \qquad K_4 = 0$$

# One-Constant Approximation

- set

$$K = K_1 = K_2 = K_3, \qquad K_4 = 0$$

- vector identities

$$(\nabla \times \mathbf{n})^2 = (\mathbf{n} \cdot \nabla \times \mathbf{n})^2 + (\mathbf{n} \times \nabla \times \mathbf{n})^2$$

$$\nabla(\mathbf{n} \cdot \mathbf{n}) = 0$$

$$\left[(\nabla \cdot \mathbf{n})^2 + (\nabla \times \mathbf{n})^2\right] + \nabla \cdot [(\mathbf{n} \cdot \nabla)\mathbf{n} - (\nabla \cdot \mathbf{n})\mathbf{n}] = \|\nabla \mathbf{n}\|^2$$

# One-Constant Approximation

- set

$$K = K_1 = K_2 = K_3, \qquad K_4 = 0$$

- vector identities

$$(\nabla \times \mathbf{n})^2 = (\mathbf{n} \cdot \nabla \times \mathbf{n})^2 + (\mathbf{n} \times \nabla \times \mathbf{n})^2$$

$$\nabla(\mathbf{n} \cdot \mathbf{n}) = 0$$

$$\left[(\nabla \cdot \mathbf{n})^2 + (\nabla \times \mathbf{n})^2\right] + \nabla \cdot [(\mathbf{n} \cdot \nabla)\mathbf{n} - (\nabla \cdot \mathbf{n})\mathbf{n}] = \|\nabla \mathbf{n}\|^2$$

- elastic energy $\qquad F_{elastic} = \dfrac{1}{2} K \|\nabla \mathbf{n}\|^2$

# Electrostatic Energy

- applied electric field $\mathbf{E}$ of magnitude $E$

- electrostatic energy

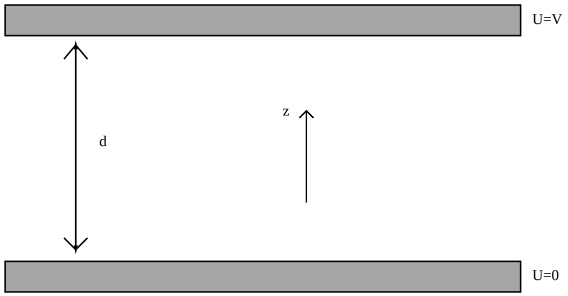
$$F_{electrostatic} = -\frac{1}{2}\epsilon_0\epsilon_\perp E^2 - \frac{1}{2}\epsilon_0\epsilon_a(\mathbf{n}\cdot\mathbf{E})^2$$

- dielectric anisotropy $\epsilon_a = \epsilon_\parallel - \epsilon_\perp$

- permittivity of free space $\epsilon_0$

# Model Problem: Twisted Nematic Device

- two parallel plates distance $d$ apart




- strong anchoring parallel to plate surfaces ($\mathbf{n}$ fixed)

- rotate one plate through $\pi/2$ radians

- electric field $\mathbf{E} = (0, 0, E(z))$,    voltage $V$

# Equilibrium Equations 1

- equilibrium equations on $z \in [0, d]$

$$F = \frac{1}{2} \int_0^d \left\{ K \|\nabla \mathbf{n}\|^2 - \epsilon_0 \epsilon_\perp E^2 - \epsilon_0 \epsilon_a (\mathbf{n} \cdot \mathbf{E})^2 \right\} dz$$

- director $\mathbf{n} = (u, v, w)$, $|\mathbf{n}| = 1$
- electric potential $U$: $E = \dfrac{dU}{dz}$
- unknowns $u$, $v$, $w$, $U$

# Equilibrium Equations 2

- nondimensionalise: $\bar{z} = \dfrac{z}{d}, \qquad \bar{U} = \dfrac{U}{V}$

- nondimensionalised equilibrium equations on $z \in [0, 1]$

$$F = \frac{1}{2} \int_0^1 \left[ (u_z^2 + v_z^2 + w_z^2) - \alpha^2 \pi^2 (\beta + w^2) U_z^2 \right] \, dz$$

- dimensionless parameters

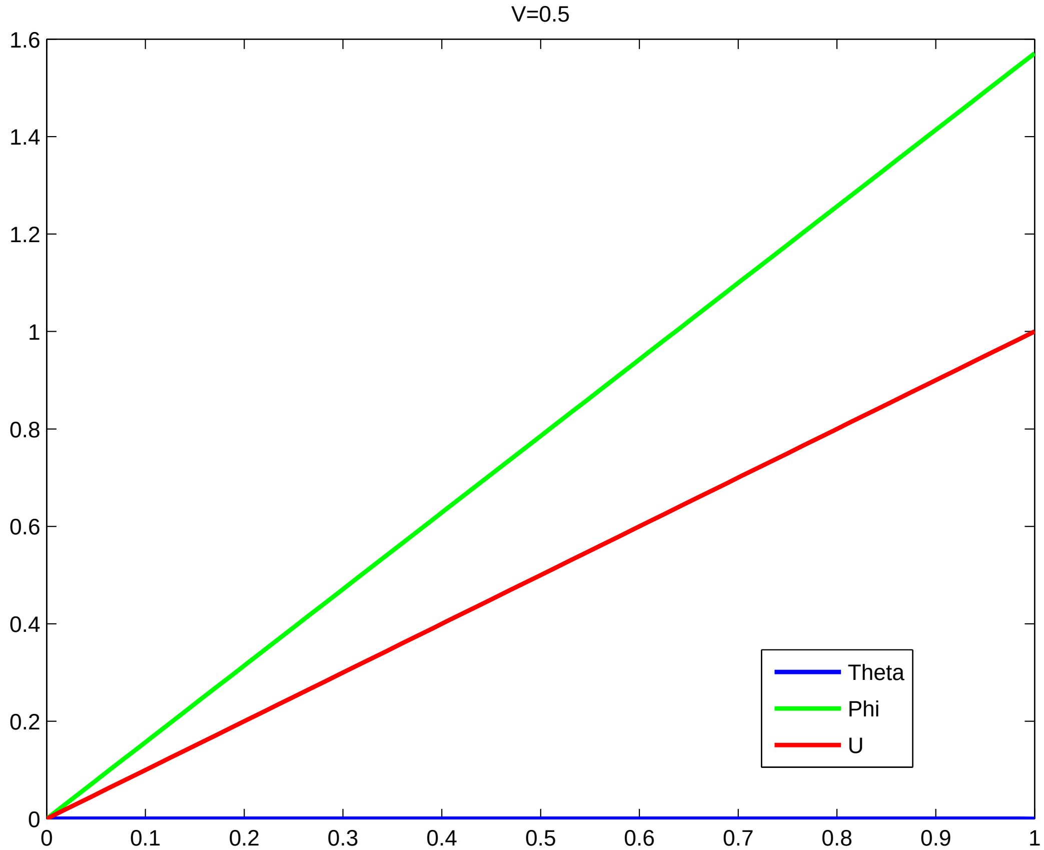
$$\alpha^2 = \frac{\epsilon_0 \epsilon_a V^2}{K \pi^2}, \qquad \beta = \frac{\epsilon_\perp}{\epsilon_a}$$

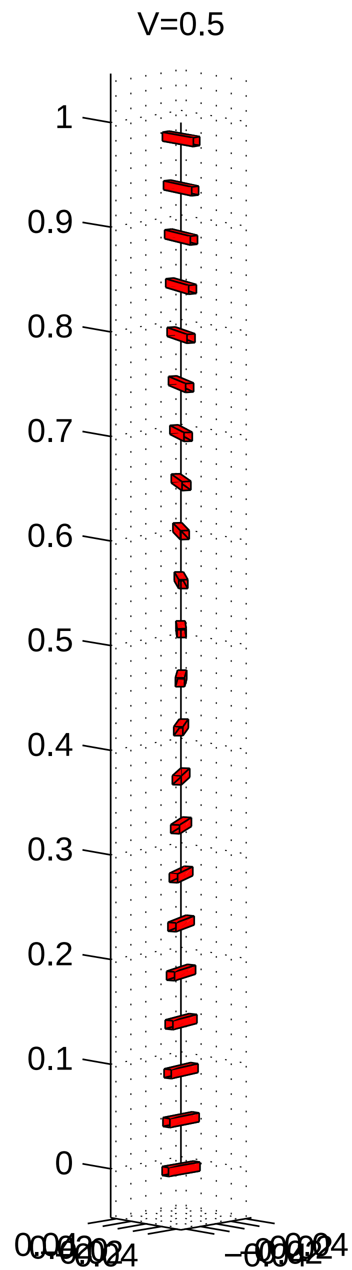
- boundary conditions:

at $z = 0$: $\mathbf{n} = (1, 0, 0)$, at $z = 1$: $\mathbf{n} = (0, 1, 0)$

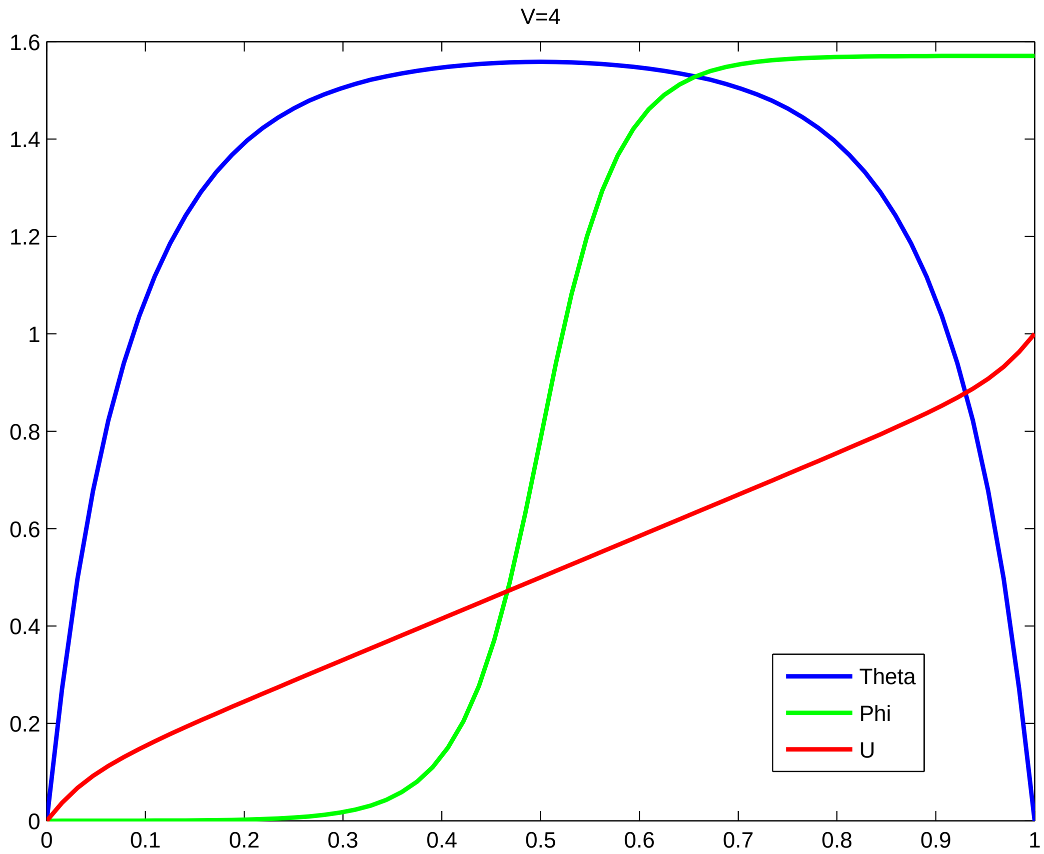
# Off State

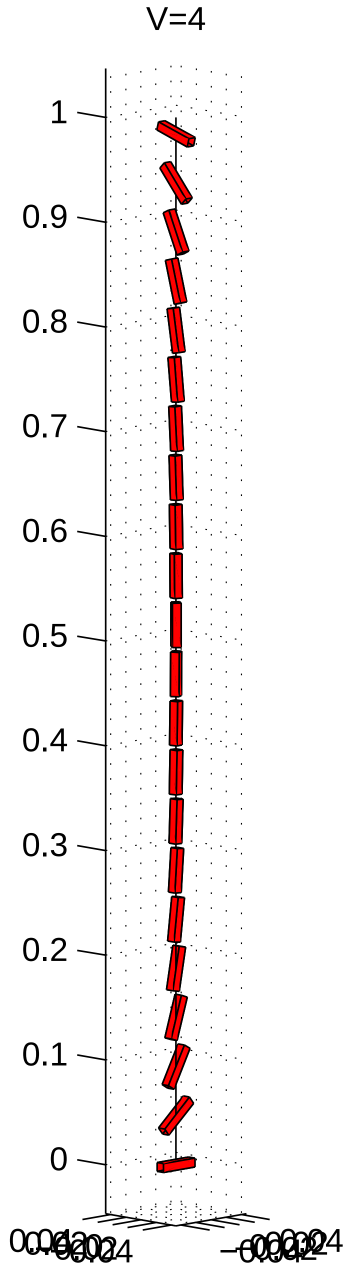
$$\theta(z) \equiv 0, \qquad \phi(z) = \frac{\pi}{2} z, \qquad U(z) = z$$



# On State

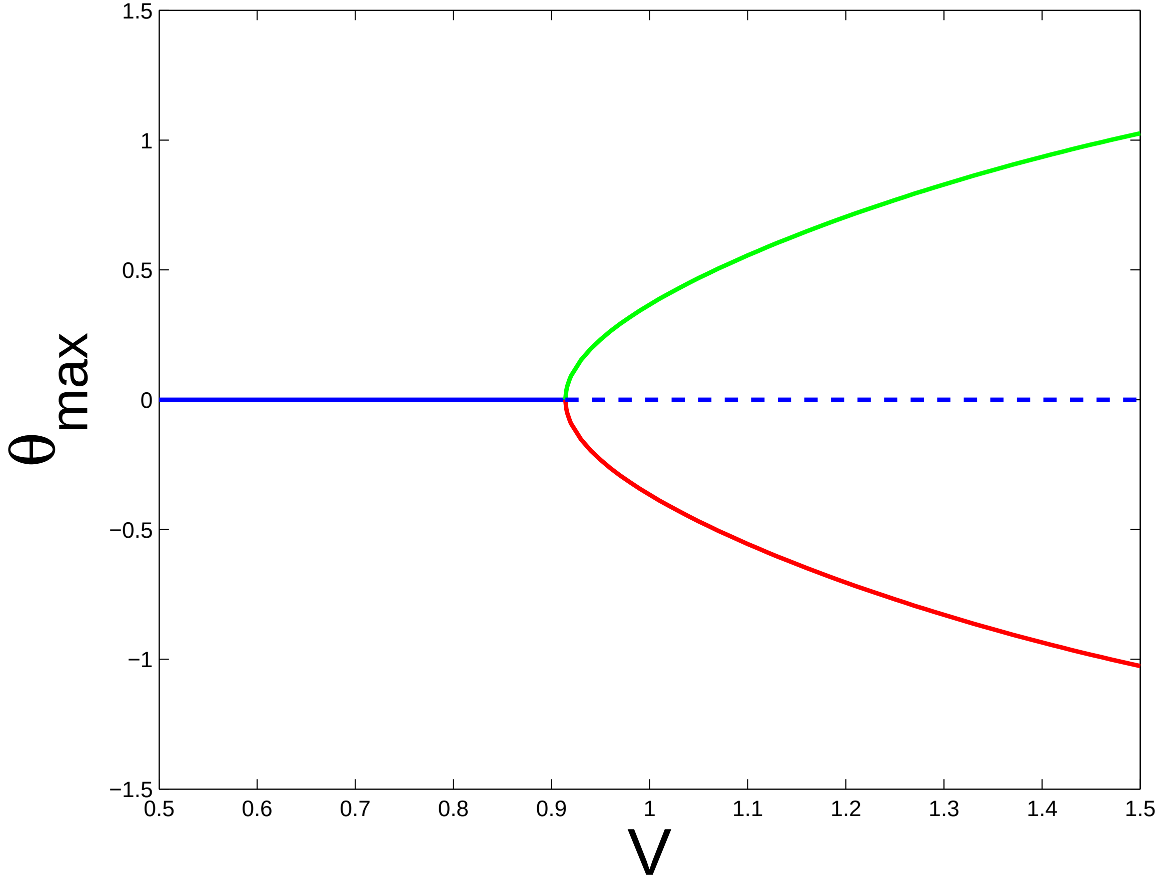

V=4
Theta
Phi
U
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.2
1.4
1.6


V=4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

# Critical Voltage

- switching occurs at

$$V_c = \frac{\pi}{2}\sqrt{\frac{3K}{\epsilon_0 \epsilon_a}}$$



# Discrete Free Energy

- grid of $N+1$ points $z_k$ a distance $\Delta z$ apart, $n = N-1$ unknowns for each variable

- piecewise linear approximation, weighted average

$$
\begin{aligned}
F \;\; \simeq \;\; & \frac{\Delta z}{2} \sum_{k=0}^{N-1} \left\{ \left[ \frac{u_{k+1} - u_k}{\Delta z} \right]^2 + \left[ \frac{v_{k+1} - v_k}{\Delta z} \right]^2 + \left[ \frac{w_{k+1} - w_k}{\Delta z} \right]^2 \right. \\
- \;\; & \left. \alpha^2 \pi^2 \left( \beta + \left[ \frac{w_k^2 + w_{k+1}^2}{2} \right] \right) \left[ \frac{U_{k+1} - U_k}{\Delta z} \right]^2 \right\}
\end{aligned}
$$

- equivalent to mid-point finite differences, linear finite elements

# Constrained Minimisation I

- discrete free energy

$$F \simeq \frac{\Delta z}{2} f(u_1, \ldots, u_n, v_1, \ldots, v_n, w_1, \ldots, w_n, U_1, \ldots, U_n)$$

- minimise $F$ subject to pointwise constraint

$$u_j^2 + v_j^2 + w_j^2 = 1, \qquad j = 1, \ldots, n$$

- constraints are applied via Lagrange multipliers: minimise

$$\begin{aligned} G = \frac{\Delta z}{2} \Big[ f \quad &- \quad \lambda_1 (u_1^2 + v_1^2 + w_1^2 - 1) - \ldots \\ & \quad\quad \lambda_n (u_n^2 + v_n^2 + w_n^2 - 1) \Big] \end{aligned}$$

# Constrained Minimisation II

- solve $\nabla \mathbf{G}(\mathbf{x}) = \mathbf{0}$ for $\mathbf{x} = [\mathbf{u}, \mathbf{v}, \mathbf{w}, \lambda, \mathbf{U}]$

  $N + 1$ gridpoints $\Rightarrow n = N - 1$ unknowns

# Constrained Minimisation II

- solve $\nabla \mathbf{G}(\mathbf{x}) = \mathbf{0}$ for $\mathbf{x} = [\mathbf{u}, \mathbf{v}, \mathbf{w}, \lambda, \mathbf{U}]$

  $N+1$ gridpoints $\Rightarrow n = N-1$ unknowns

- use Newton's method: solve

$$\nabla^2 \mathbf{G}(\mathbf{x}_j) \cdot \delta \mathbf{x}_j = -\nabla \mathbf{G}(\mathbf{x}_j)$$

# Constrained Minimisation II

- solve $\nabla \mathbf{G}(\mathbf{x}) = \mathbf{0}$ for $\mathbf{x} = [\mathbf{u}, \mathbf{v}, \mathbf{w}, \lambda, \mathbf{U}]$

  $N+1$ gridpoints $\Rightarrow n = N-1$ unknowns

- use Newton's method: solve

$$\nabla^2 \mathbf{G}(\mathbf{x}_j) \cdot \delta \mathbf{x}_j = -\nabla \mathbf{G}(\mathbf{x}_j)$$

- $5n \times 5n$ coefficient matrix is Hessian $\nabla^2 \mathbf{G}(\mathbf{x})$

$$\nabla^2 \mathbf{G} = \begin{bmatrix} \nabla^2_{\mathbf{nn}} \mathbf{G} & \nabla^2_{\mathbf{n}\lambda} \mathbf{G} & \nabla^2_{\mathbf{nU}} \mathbf{G} \\ \nabla^2_{\lambda\mathbf{n}} \mathbf{G} & \nabla^2_{\lambda\lambda} \mathbf{G} & \nabla^2_{\mathbf{U}\lambda} \mathbf{G} \\ \nabla^2_{\mathbf{Un}} \mathbf{G} & \nabla^2_{\lambda\mathbf{U}} \mathbf{G} & \nabla^2_{\mathbf{UU}} \mathbf{G} \end{bmatrix}$$

# Hessian Components 1

- matrix notation: $\nabla^2_{\mathbf{nn}}\mathbf{G} = A$

$$A = \begin{bmatrix} \nabla^2_{\mathbf{uu}}\mathbf{G} & 0 & 0 \\ 0 & \nabla^2_{\mathbf{vv}}\mathbf{G} & 0 \\ 0 & 0 & \nabla^2_{\mathbf{ww}}\mathbf{G} \end{bmatrix} = \begin{bmatrix} A_{uu} & 0 & 0 \\ 0 & A_{vv} & 0 \\ 0 & 0 & A_{ww} \end{bmatrix}$$

- $A_{uu}$, $A_{vv}$ and $A_{ww}$ are $n \times n$ symmetric tridiagonal blocks

# Hessian Components 1

- matrix notation: $\nabla^2_{\mathbf{nn}}\mathbf{G} = A$

$$A = \begin{bmatrix} \nabla^2_{\mathbf{uu}}\mathbf{G} & 0 & 0 \\ 0 & \nabla^2_{\mathbf{vv}}\mathbf{G} & 0 \\ 0 & 0 & \nabla^2_{\mathbf{ww}}\mathbf{G} \end{bmatrix} = \begin{bmatrix} A_{uu} & 0 & 0 \\ 0 & A_{vv} & 0 \\ 0 & 0 & A_{ww} \end{bmatrix}$$

- $A_{uu}$, $A_{vv}$ and $A_{ww}$ are $n \times n$ symmetric tridiagonal blocks
- $A_{uu} = A_{vv} = \dfrac{1}{\Delta z}\mathtt{tri}(-1, 2 - \Delta z^2\lambda_j, -1)$
- $A_{ww} = \dfrac{1}{\Delta z}\mathtt{tri}(-1, 2 - \Delta z^2\lambda_j - \gamma_j, -1)$

$$\gamma_j = \frac{\alpha^2\pi^2}{2}[(U_{j+1} - U_j)^2 + (U_j - U_{j-1})^2]$$

# Eigenvalues of $A$

- off state: first Newton step, linear $U$, constant $\lambda$

$$\lambda_j = \lambda = \frac{4}{\Delta z^2} \sin^2 \left( \frac{\pi \Delta z}{4} \right)$$

- block matrices are Toeplitz

# Eigenvalues of $A$

- off state: first Newton step, linear $U$, constant $\lambda$

$$\lambda_j = \lambda = \frac{4}{\Delta z^2} \sin^2 \left( \frac{\pi \Delta z}{4} \right)$$

- block matrices are Toeplitz

- $\sigma_{\min}(A_{uu}) = \sigma_{\min}(A_{vv}) \simeq \frac{3\pi^2}{4} \Delta z > 0$

  $A_{uu}$ and $A_{vv}$ are positive definite

- $\sigma_{\min}(A_{ww}) \simeq \left( \frac{3\pi^2}{4} - \alpha^2 \pi^2 \right) \Delta z$

  $A_{ww}$ is positive definite iff $V < V_c$

# Eigenvalues of $A$

- off state: first Newton step, linear $U$, constant $\lambda$

$$\lambda_j = \lambda = \frac{4}{\Delta z^2} \sin^2 \left( \frac{\pi \Delta z}{4} \right)$$

- block matrices are Toeplitz

- $\sigma_{\min}(A_{uu}) = \sigma_{\min}(A_{vv}) \simeq \frac{3\pi^2}{4} \Delta z > 0$

  $A_{uu}$ and $A_{vv}$ are positive definite

- $\sigma_{\min}(A_{ww}) \simeq \left( \frac{3\pi^2}{4} - \alpha^2 \pi^2 \right) \Delta z$

  $A_{ww}$ is positive definite iff $V < V_c$

- number of negative eigenvalues increases with $V$

# Hessian Components 2

- matrix notation: $\nabla^2_{\mathbf{n}\lambda}\mathbf{G} = B$

- the $3n \times n$ matrix $B$ has structure

$$B = -\Delta z \begin{bmatrix} B_u \\ B_v \\ B_w \end{bmatrix}, \qquad \begin{array}{l} B_u = \mathtt{diag}(\mathbf{u}) \\ B_v = \mathtt{diag}(\mathbf{v}) \\ B_w = \mathtt{diag}(\mathbf{w}) \end{array}$$

# Hessian Components 2

- matrix notation: $\nabla^2_{\mathbf{n}\lambda}\mathbf{G} = B$

- the $3n \times n$ matrix $B$ has structure

$$B = -\Delta z \begin{bmatrix} B_u \\ B_v \\ B_w \end{bmatrix}, \qquad \begin{aligned} B_u &= \mathtt{diag}(\mathbf{u}) \\ B_v &= \mathtt{diag}(\mathbf{v}) \\ B_w &= \mathtt{diag}(\mathbf{w}) \end{aligned}$$

- $B^T B = \Delta z^2 I_n$ when constraints are satisfied

- $\mathrm{rank}(B) = \mathrm{rank}(B^T) = \mathrm{rank}(BB^T) = \mathrm{rank}(B^T B) = n$

# Hessian Components 3

- matrix notation: $\nabla^2_{\mathbf{UU}}\mathbf{G} = -C$
- the $n \times n$ matrix $C$ is symmetric and tridiagonal

# Hessian Components 3

- matrix notation: $\nabla^2_{\mathbf{UU}}\mathbf{G} = -C$
- the $n \times n$ matrix $C$ is symmetric and tridiagonal
- $C = \dfrac{1}{\Delta z}\mathtt{tri}(-a_{j-\frac{1}{2}}, a_{j-\frac{1}{2}} + a_{j+\frac{1}{2}}, -a_{j+\frac{1}{2}})$

$$a_{j-\frac{1}{2}} = \alpha^2\pi^2(\beta + \frac{1}{2}(w_{j-1}^2 + w_j^2)) > 0$$

$$a_{j+\frac{1}{2}} = \alpha^2\pi^2(\beta + \frac{1}{2}(w_j^2 + w_{j+1}^2)) > 0$$

# Hessian Components 3

- matrix notation: $\nabla^2_{\mathbf{UU}}\mathbf{G} = -C$
- the $n \times n$ matrix $C$ is symmetric and tridiagonal
- $C = \dfrac{1}{\Delta z}\texttt{tri}(-a_{j-\frac{1}{2}}, a_{j-\frac{1}{2}} + a_{j+\frac{1}{2}}, -a_{j+\frac{1}{2}})$

$$a_{j-\frac{1}{2}} = \alpha^2\pi^2(\beta + \frac{1}{2}(w_{j-1}^2 + w_j^2)) > 0$$

$$a_{j+\frac{1}{2}} = \alpha^2\pi^2(\beta + \frac{1}{2}(w_j^2 + w_{j+1}^2)) > 0$$

- diagonally dominant with positive real diagonal entries

$C$ is positive definite

# Hessian Components 4

- matrix notation: $\nabla^2_{\mathbf{nU}}\mathbf{G} = D$

$$D = \frac{\alpha^2\pi^2}{\Delta z}\begin{bmatrix} 0 \\ 0 \\ D_w \end{bmatrix}$$

- the $n \times n$ matrix $D_w$ is tridiagonal

$$D_w = \texttt{diag}(\mathbf{w})\texttt{tri}(U_j - U_{j-1}, U_{j-1} - 2U_j + U_{j+1}, U_j - U_{j+1})$$

# Hessian Components 4

- matrix notation: $\nabla^2_{\mathbf{nU}}\mathbf{G} = D$

$$D = \frac{\alpha^2\pi^2}{\Delta z}\begin{bmatrix} 0 \\ 0 \\ D_w \end{bmatrix}$$

- the $n \times n$ matrix $D_w$ is tridiagonal

$$D_w = \texttt{diag}(\mathbf{w})\texttt{tri}(U_j - U_{j-1}, U_{j-1} - 2U_j + U_{j+1}, U_j - U_{j+1})$$

- $D_w$ has complex eigenvalues in conjugate pairs and one zero eigenvalue ($N$ even)

- $\mathrm{rank}(D) = n - 1$

# Full Hessian Structure

$$\nabla^2 \mathbf{G} = \begin{bmatrix} \nabla^2_{\mathbf{nn}}\mathbf{G} & \nabla^2_{\mathbf{n}\lambda}\mathbf{G} & \nabla^2_{\mathbf{nU}}\mathbf{G} \\ \nabla^2_{\lambda\mathbf{n}}\mathbf{G} & \nabla^2_{\lambda\lambda}\mathbf{G} & \nabla^2_{\mathbf{U}\lambda}\mathbf{G} \\ \nabla^2_{\mathbf{Un}}\mathbf{G} & \nabla^2_{\lambda\mathbf{U}}\mathbf{G} & \nabla^2_{\mathbf{UU}}\mathbf{G} \end{bmatrix}$$

$$\nabla^2 \mathbf{G} = \begin{bmatrix} A & B & D \\ B^T & 0 & 0 \\ D^T & 0 & -C \end{bmatrix}$$

saddle-point problem

# Four Saddle-Point Problems

- for unknown vector ordered as $\mathbf{x} = [\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{U}, \lambda]$

$$H = \left[\begin{array}{c|cc} A & D & B \\ \hline D^T & -C & 0 \\ B^T & 0 & 0 \end{array}\right] \qquad H = \left[\begin{array}{cc|c} A & D & B \\ D^T & -C & 0 \\ \hline B^T & 0 & 0 \end{array}\right]$$

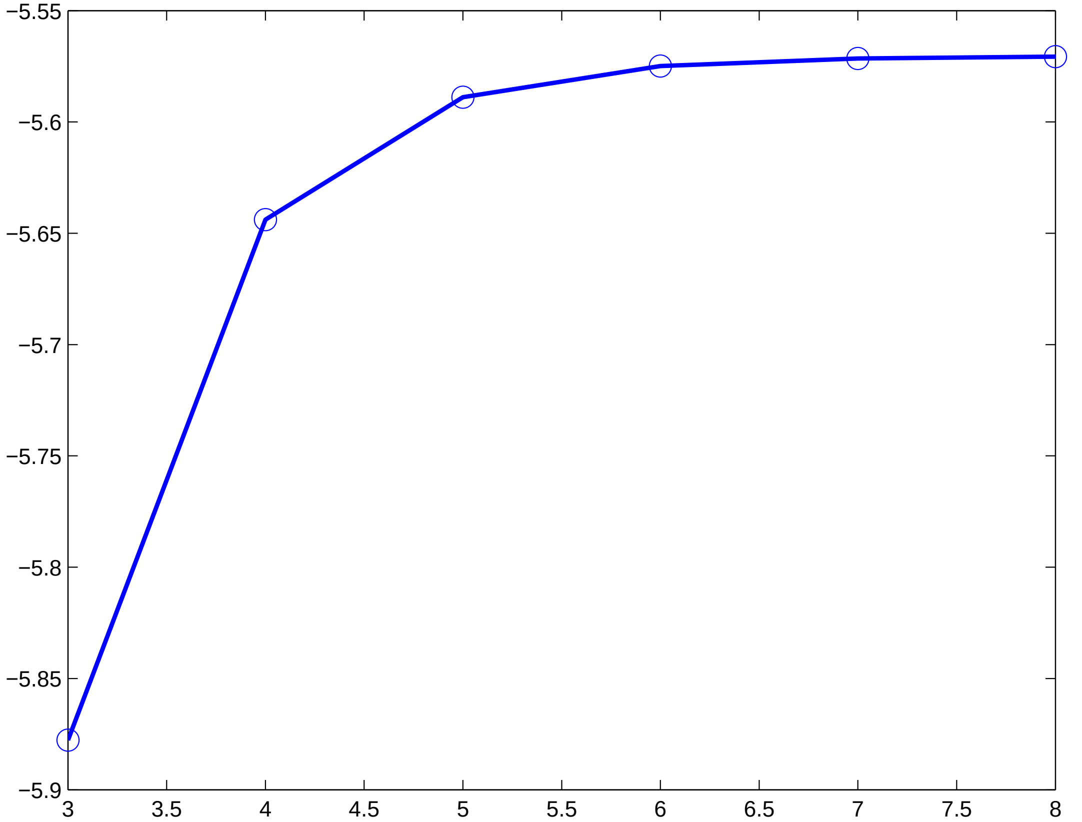
- for unknown vector ordered as $\mathbf{x} = [\mathbf{u}, \mathbf{v}, \mathbf{w}, \lambda, \mathbf{U}]$

$$H = \left[\begin{array}{c|cc} A & B & D \\ \hline B^T & 0 & 0 \\ D^T & 0 & -C \end{array}\right] \qquad H = \left[\begin{array}{cc|c} A & B & D \\ B^T & 0 & 0 \\ \hline D^T & 0 & -C \end{array}\right]$$

double saddle-point structure

# Iterative Solution

- outer iteration: Newton's method tol=$1e-4$
- inner iteration: MINRES tol=$1e-4$
- check accuracy by calculating energy of final solution



# MINRES

Paige and Saunders (1975)

Construct iterates $\mathbf{x}_k = \mathbf{x}_0 + V_k \mathbf{y}_k$ with properties

- $\mathbf{x}_k$ minimises $\|\mathbf{r}_k\|_2 = \|\mathbf{b} - H\mathbf{x}_k\|_2$
- uses three-term recurrence relation

$$V_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$$

$\mathbf{v}_k$ form an orthonormal basis for Krylov subspace

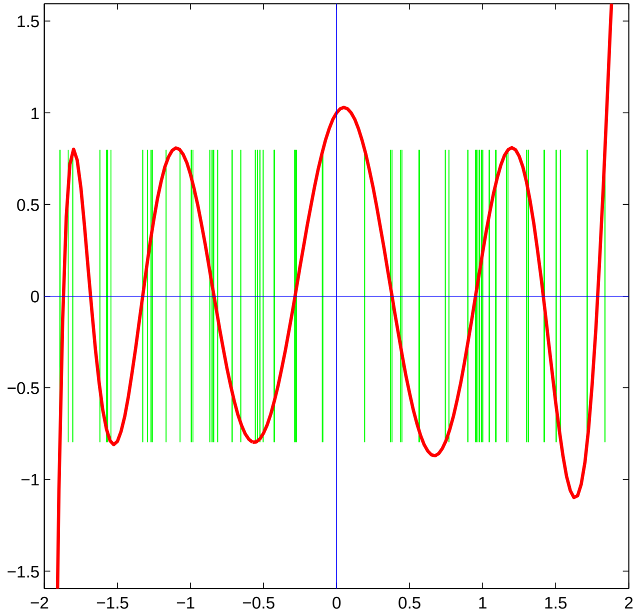
$$\kappa(H, \mathbf{r}_0, k) = \text{span}\{\mathbf{r}_0, H\mathbf{r}_0, \ldots, H^{k-1}\mathbf{r}_0\}$$

- use Lanczos method to find $\mathbf{v}_k$
- solve resulting least squares problem for $\mathbf{y}_k$ using Givens rotations and QR factorisation

# Convergence of MINRES

- at step $k$:

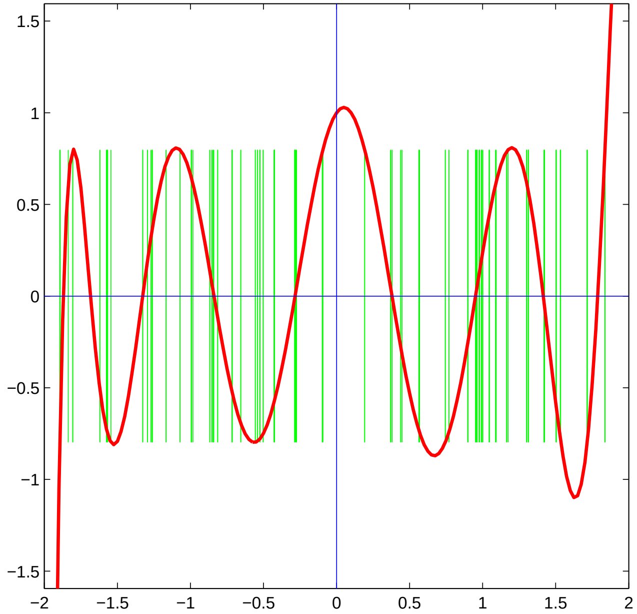
$$\|\mathbf{r}_k\|_2 \leq \min_{p_k \in \Pi_k^1} \max_i |p_k(\lambda_i)| \|\mathbf{r}_0\|_2$$



# Convergence of MINRES

- at step $k$:

$$\|\mathbf{r}_k\|_2 \leq \min_{p_k \in \Pi_k^1} \max_i |p_k(\lambda_i)| \|\mathbf{r}_0\|_2$$




- symmetric intervals: $[-\lambda_{\max}, -\lambda_{\min}] \cup [\lambda_{\min}, \lambda_{\max}]$

$$k \propto \frac{\lambda_{\max}}{\lambda_{\min}}$$

# Minres iterations for full system

| | | off state ($\alpha = 0.5\alpha_c$) | | on state ($\alpha = 1.5\alpha_c$) | |
|---|---|---|---|---|---|
| $N$ | $d$ | first step | last step | first step | last step |
| 32 | 155 | 226 | 499 | 291 | 691 |
| 64 | 315 | 728 | 2,004 | 1,172 | 3,571 |
| 128 | 635 | 2,680 | 8,528 | 4,106 | 17,498 |
| 256 | 1,275 | 10,253 | 41,666 | 15,727 | 85,784 |
| 512 | 2,555 | 38,809 | 194,753 | 57,499 | $>$200,000 |
| 1,024 | 5,115 | 150,376 | $>$200,000 | $>$200,000 | $>$200,000 |

- doubling $N$ quadruples iteration count

# Nullspace Method I

- Newton system:

$$\begin{bmatrix} A & B & D \\ B^T & 0 & 0 \\ D^T & 0 & -C \end{bmatrix} \begin{bmatrix} \delta\mathbf{n} \\ \delta\lambda \\ \delta\mathbf{U} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{n}} G \\ -\nabla_{\lambda} G \\ -\nabla_{\mathbf{U}} G \end{bmatrix}$$

# Nullspace Method I

- Newton system:

$$\begin{bmatrix} A & B & D \\ B^T & 0 & 0 \\ D^T & 0 & -C \end{bmatrix} \begin{bmatrix} \delta\mathbf{n} \\ \delta\lambda \\ \delta\mathbf{U} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{n}} G \\ -\nabla_{\lambda} G \\ -\nabla_{\mathbf{U}} G \end{bmatrix}$$

- Idea: use information about nullspace of $B$ to eliminate constraint blocks

# Nullspace Method I

- Newton system:

$$\begin{bmatrix} A & B & D \\ B^T & 0 & 0 \\ D^T & 0 & -C \end{bmatrix} \begin{bmatrix} \delta \mathbf{n} \\ \delta \lambda \\ \delta \mathbf{U} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{n}} G \\ -\nabla_{\lambda} G \\ -\nabla_{\mathbf{U}} G \end{bmatrix}$$

- Idea: use information about nullspace of $B$ to eliminate constraint blocks
- use $Z \in \mathbb{R}^{3n \times 2n}$ whose columns form a basis for the nullspace of $B^T$

$$B^T Z = Z^T B = 0$$

- $\mathrm{rank}(Z) = 2n$

# Nullspace Method I

- Newton system:

$$\begin{bmatrix} A & B & D \\ B^T & 0 & 0 \\ D^T & 0 & -C \end{bmatrix} \begin{bmatrix} \delta \mathbf{n} \\ \delta \lambda \\ \delta \mathbf{U} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{n}} G \\ -\nabla_{\lambda} G \\ -\nabla_{\mathbf{U}} G \end{bmatrix}$$

- Idea: use information about nullspace of $B$ to eliminate constraint blocks
- use $Z \in \mathbb{R}^{3n \times 2n}$ whose columns form a basis for the nullspace of $B^T$

$$B^T Z = Z^T B = 0$$

- $\mathrm{rank}(Z) = 2n$
- system size will reduce from $5n \times 5n$ to $3n \times 3n$

# Nullspace Method II

$$
\begin{aligned}
A\delta\mathbf{n} + B\delta\lambda + D\delta\mathbf{U} &= -\nabla_{\mathbf{n}} G && (1)\\
B^T\delta\mathbf{n} &= -\nabla_{\lambda} G && (2)\\
D^T\delta\mathbf{n} - C\delta\mathbf{U} &= -\nabla_{\mathbf{U}} G && (3)
\end{aligned}
$$

# Nullspace Method II

$$\begin{aligned} A\delta\mathbf{n} + B\delta\lambda + D\delta\mathbf{U} &= -\nabla_{\mathbf{n}}G && (1) \\ B^T\delta\mathbf{n} &= -\nabla_{\lambda}G && (2) \\ D^T\delta\mathbf{n} - C\delta\mathbf{U} &= -\nabla_{\mathbf{U}}G && (3) \end{aligned}$$

- write solution of (2) as

$$\delta\mathbf{n} = \widehat{\delta\mathbf{n}} + Z\mathbf{z}$$

  - particular solution satisfies $B^T\widehat{\delta\mathbf{n}} = -\nabla_{\lambda}G$
  - $Z\mathbf{z} \in \mathbb{R}^{2n}$ lies in nullspace of $B^T$

# Nullspace Method II

$$
\begin{aligned}
A\delta\mathbf{n} + B\delta\lambda + D\delta\mathbf{U} &= -\nabla_{\mathbf{n}} G && (1)\\
B^T\delta\mathbf{n} &= -\nabla_{\lambda} G && (2)\\
D^T\delta\mathbf{n} - C\delta\mathbf{U} &= -\nabla_{\mathbf{U}} G && (3)
\end{aligned}
$$

- write solution of (2) as

$$\delta\mathbf{n} = \widehat{\delta\mathbf{n}} + Z\mathbf{z}$$

  - particular solution satisfies $B^T\widehat{\delta\mathbf{n}} = -\nabla_{\lambda} G$
  - $Z\mathbf{z} \in \mathbb{R}^{2n}$ lies in nullspace of $B^T$

- find $\widehat{\delta\mathbf{n}}$ via $\widehat{\delta\mathbf{n}} = -B(B^TB)^{-1}\nabla_{\lambda} G$

# Nullspace Method III

- reduced system:

$$\begin{bmatrix} Z^T A Z & Z^T D \\ D^T Z & -C \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \delta \mathbf{U} \end{bmatrix} = \begin{bmatrix} -Z^T(\nabla_{\mathbf{n}} G + A\widehat{\delta \mathbf{n}}) \\ -\nabla_{\mathbf{U}} G - D^T \widehat{\delta \mathbf{n}} \end{bmatrix}$$

# Nullspace Method III

- reduced system:

$$\begin{bmatrix} Z^T A Z & Z^T D \\ D^T Z & -C \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \delta \mathbf{U} \end{bmatrix} = \begin{bmatrix} -Z^T(\nabla_{\mathbf{n}} G + A\widehat{\delta \mathbf{n}}) \\ -\nabla_{\mathbf{U}} G - D^T \widehat{\delta \mathbf{n}} \end{bmatrix}$$

- recover full solution from

$$\begin{aligned} \widehat{\delta \mathbf{n}} &= -B(B^T B)^{-1} \nabla_\lambda G \\ \delta \mathbf{n} &= Z\mathbf{z} + \widehat{\delta \mathbf{n}} \\ \delta \lambda &= (B^T B)^{-1} B^T (-\nabla_{\mathbf{n}} G - A\delta \mathbf{n} - D\delta \mathbf{U}) \end{aligned}$$

# Nullspace Method III

- reduced system:

$$\begin{bmatrix} Z^T A Z & Z^T D \\ D^T Z & -C \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \delta \mathbf{U} \end{bmatrix} = \begin{bmatrix} -Z^T(\nabla_{\mathbf{n}} G + A\widehat{\delta \mathbf{n}}) \\ -\nabla_{\mathbf{U}} G - D^T \widehat{\delta \mathbf{n}} \end{bmatrix}$$

- recover full solution from

$$\begin{aligned} \widehat{\delta \mathbf{n}} &= -B(B^T B)^{-1} \nabla_\lambda G \\ \delta \mathbf{n} &= Z\mathbf{z} + \widehat{\delta \mathbf{n}} \\ \delta \lambda &= (B^T B)^{-1} B^T(-\nabla_{\mathbf{n}} G - A\delta \mathbf{n} - D\delta \mathbf{U}) \end{aligned}$$

- here $B^T B$ is diagonal so solve is cheap

# Nullspace of $B^T$ I

- permute entries of B:

$$B = -\Delta z \begin{bmatrix} \mathbf{n}_1 & & & \\ & \mathbf{n}_2 & & \\ & & \ddots & \\ & & & \mathbf{n}_n \end{bmatrix}, \quad \mathbf{n}_j = \begin{bmatrix} u_j \\ v_j \\ w_j \end{bmatrix}$$

# Nullspace of $B^T$ I

- permute entries of B:

$$B = -\Delta z \begin{bmatrix} \mathbf{n}_1 & & & \\ & \mathbf{n}_2 & & \\ & & \ddots & \\ & & & \mathbf{n}_n \end{bmatrix}, \quad \mathbf{n}_j = \begin{bmatrix} u_j \\ v_j \\ w_j \end{bmatrix}$$

- eigenvectors of orthogonal projection

$$I - \mathbf{n}_j \otimes \mathbf{n}_j = \begin{bmatrix} 1-u_j^2 & -v_j u_j & -w_j u_j \\ -u_j v_j & 1-v_j^2 & -w_j v_j \\ -u_j w_j & -v_j w_j & 1-w_j^2 \end{bmatrix}$$

will be orthogonal to $\mathbf{n}_j$

# Nullspace of $B^T$ II

- eigenvectors of orthogonal projection: e.g.

$$\mathbf{l}_j = \begin{bmatrix} -\dfrac{v_j}{u_j} \\ 1 \\ 0 \end{bmatrix}, \qquad \mathbf{m}_j = \begin{bmatrix} -\dfrac{w_j}{u_j} \\ 0 \\ 1 \end{bmatrix} \qquad (u_j \neq 0)$$

- orthonormalise:

$$\mathbf{l}_j = \frac{1}{\sqrt{u_j^2 + v_j^2}} \begin{bmatrix} -v_j \\ u_j \\ 0 \end{bmatrix}, \qquad \mathbf{m}_j = \frac{1}{\sqrt{u_j^2 + v_j^2}} \begin{bmatrix} -u_j w_j \\ -v_j w_j \\ u_j^2 + v_j^2 \end{bmatrix}$$

- at least one of $u_j, v_j, w_j$ nonzero as $|\mathbf{n}_j| = 1$

# Nullspace of $B^T$ III

$$Z = \begin{bmatrix} \mathbf{l}_1 & \mathbf{m}_1 & & & & & & \\ & & \mathbf{l}_2 & \mathbf{m}_2 & & & & \\ & & & & \ddots & & & \\ & & & & & & \mathbf{l}_n & \mathbf{m}_n \end{bmatrix}$$

# Nullspace of $B^T$ III

$$Z = \begin{bmatrix} \mathbf{l}_1 & \mathbf{m}_1 & & & & & \\ & & \mathbf{l}_2 & \mathbf{m}_2 & & & \\ & & & & \ddots & & \\ & & & & & \mathbf{l}_n & \mathbf{m}_n \end{bmatrix}$$

- consider $B^T Z \mathbf{p}$ where $\mathbf{p} = [p_1, q_1, p_2, q_2, \ldots, p_n, q_n]^T$:

$$B^T Z \mathbf{p} = \begin{bmatrix} \mathbf{n}_1^T & & & \\ & \mathbf{n}_2^T & & \\ & & \ddots & \\ & & & \mathbf{n}_n^T \end{bmatrix} \begin{bmatrix} p_1 \mathbf{l}_1 + q_1 \mathbf{m}_1 \\ p_2 \mathbf{l}_2 + q_2 \mathbf{m}_2 \\ \vdots \\ p_n \mathbf{l}_n + q_n \mathbf{m}_n \end{bmatrix} = 0$$

- columns of $Z$ form a basis for nullspace of $B^T$

# Minres iterations for reduced system

| | | off state ($\alpha = 0.5\alpha_c$) | | on state ($\alpha = 1.5\alpha_c$) | |
|---|---|---|---|---|---|
| $N$ | $d$ | first step | last step | first step | last step |
| 32 | 93 | 59 | 128 | 90 | 172 |
| 64 | 189 | 187 | 418 | 285 | 557 |
| 128 | 381 | 660 | 1,456 | 1,004 | 2,002 |
| 256 | 765 | 2,562 | 5,455 | 3,650 | 7,043 |
| 512 | 1,533 | 9,983 | 21,393 | 13,907 | 26,504 |
| 1,024 | 3,069 | 41,267 | 80,778 | 55,563 | 81,821 |
| 2,048 | 6,141 | 171,385 | $>$200,000 | $>$200,000 | $>$200,000 |

- doubling $N$ quadruples iteration count

# Preconditioning

Idea: instead of solving $\mathcal{H}\mathbf{x} = \mathbf{b}$, solve

$$\mathcal{P}^{-1}\mathcal{H}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

for some preconditioner $\mathcal{P}$

Choose $\mathcal{P}$ so that

(i) eigenvalues of $\mathcal{P}^{-1}\mathcal{H}$ are well clustered
(ii) $\mathcal{P}\mathbf{u} = \mathbf{r}$ is easily solved

# Preconditioning

Idea: instead of solving $\mathcal{H}\mathbf{x} = \mathbf{b}$, solve

$$\mathcal{P}^{-1}\mathcal{H}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

for some preconditioner $\mathcal{P}$

Choose $\mathcal{P}$ so that

(i) eigenvalues of $\mathcal{P}^{-1}\mathcal{H}$ are well clustered
(ii) $\mathcal{P}\mathbf{u} = \mathbf{r}$ is easily solved

Extreme cases:

- $\mathcal{P} = \mathcal{H}$: good for (i), bad for (ii)
- $\mathcal{P} = I$: good for (ii), bad for (i)

# Diagonal scaling

- set $\mathcal{P} = \mathrm{diag}(\mathcal{H})$
- $\mathcal{P}$ is cheap to invert

| | off state ($\alpha = 0.5\alpha_c$) | | on state ($\alpha = 1.5\alpha_c$) | |
|---|---|---|---|---|
| $N$ | first step | last step | first step | last step |
| 32 | 55 | 129 | 54 | 122 |
| 64 | 169 | 408 | 167 | 390 |
| 128 | 573 | 1,469 | 565 | 1,423 |
| 256 | 2,144 | 5,479 | 2060 | 5,301 |
| 512 | 8,254 | 21,196 | 8,148 | 20,804 |
| 1,024 | 33,438 | 85,154 | 33,849 | 80,221 |
| 2,048 | 136,015 | $>$200,000 | 133,605 | $>$200,000 |

# Ideal Block Preconditioner

- write $\bar{A} = Z^T A Z$ and $\bar{D} = Z^T D$:

$$\mathcal{H} = \begin{bmatrix} \bar{A} & \bar{D} \\ \bar{D}^T & -C \end{bmatrix}$$

# Ideal Block Preconditioner

- write $\bar{A} = Z^T A Z$ and $\bar{D} = Z^T D$:

$$\mathcal{H} = \begin{bmatrix} \bar{A} & \bar{D} \\ \bar{D}^T & -C \end{bmatrix}$$

- block preconditioner: $$\mathcal{P} = \begin{bmatrix} \bar{A} & 0 \\ 0 & C \end{bmatrix}$$

# Ideal Block Preconditioner

- write $\bar{A} = Z^T A Z$ and $\bar{D} = Z^T D$:

$$\mathcal{H} = \begin{bmatrix} \bar{A} & \bar{D} \\ \bar{D}^T & -C \end{bmatrix}$$

- block preconditioner: $\mathcal{P} = \begin{bmatrix} \bar{A} & 0 \\ 0 & C \end{bmatrix}$

- preconditioned matrix:

$$\tilde{\mathcal{H}} = \mathcal{P}^{-1/2}\mathcal{H}\mathcal{P}^{-1/2} = \begin{bmatrix} I & M^T \\ M & -I \end{bmatrix}$$

$$M = C^{-1/2}\bar{D}\bar{A}^{-1/2}$$

# Preconditioned Spectrum

$$\tilde{\mathcal{H}} = \mathcal{P}^{-1/2}\mathcal{H}\mathcal{P}^{-1/2} = \begin{bmatrix} I & M^T \\ M & -I \end{bmatrix}$$

- $M = C^{-1/2} Z^T D (Z^T A Z)^{-1/2}$
- rank($M$)=$n - 1$
- non-zero singular values $\sigma_k$

# Preconditioned Spectrum

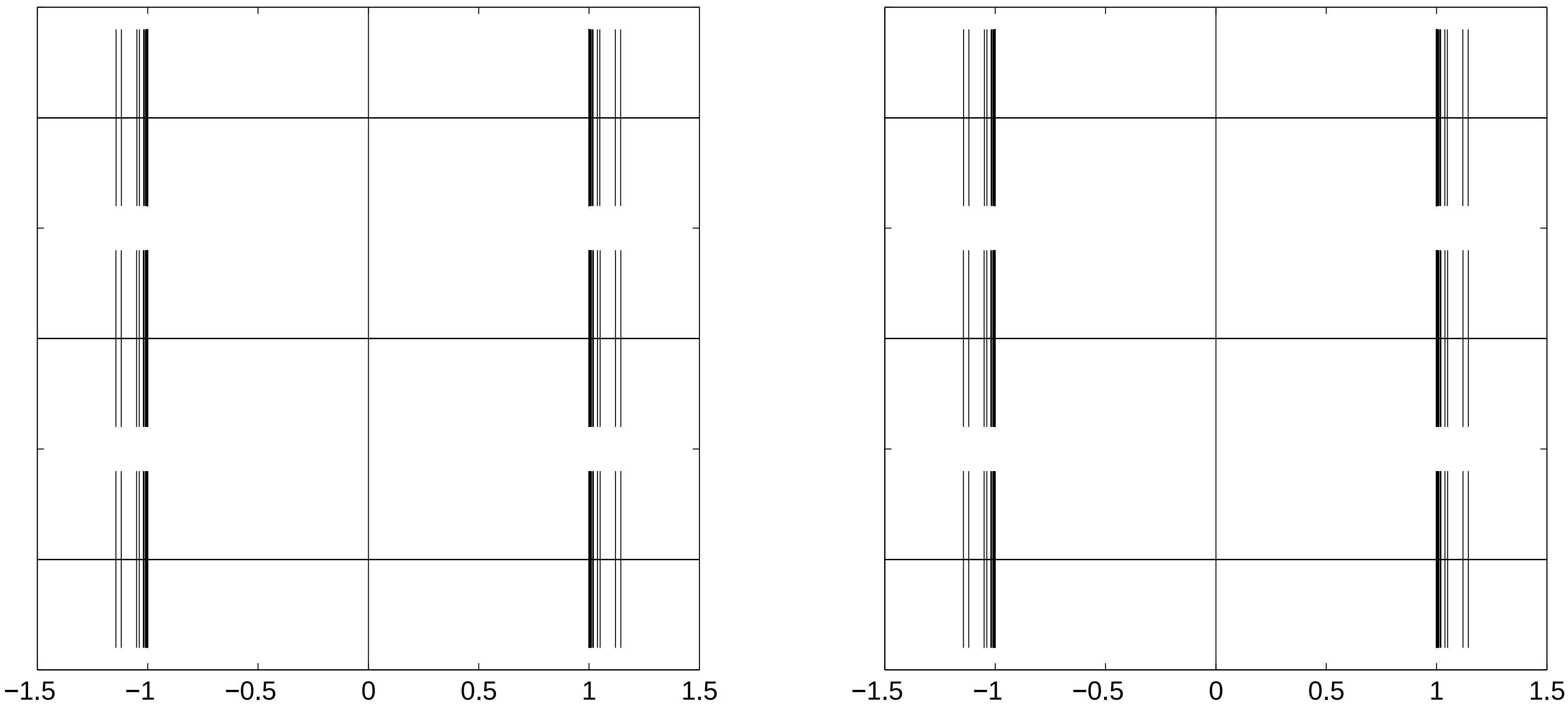
$$\tilde{\mathcal{H}} = \mathcal{P}^{-1/2}\mathcal{H}\mathcal{P}^{-1/2} = \begin{bmatrix} I & M^T \\ M & -I \end{bmatrix}$$

- $M = C^{-1/2}Z^T D(Z^T AZ)^{-1/2}$
- rank($M$)=$n-1$
- non-zero singular values $\sigma_k$

- $3n$ eigenvalues of $\tilde{\mathcal{H}}$ are

(i) 1 with multiplicity $n+1$
(ii) -1 with multiplicity $1$
(iii) $\pm\sqrt{1+\sigma_k^2}$ for $k = 1, \ldots, n-1$

# Sample Eigenvalue Plots




eigenvalue plots for $N = 32, 64, 128$
first and last Newton iteration

# Estimate of MINRES convergence

- eigenvalues in two symmetric intervals

$$[-\beta, -1] \cup [1, \beta], \qquad \beta = \sqrt{1 + \sigma_{\max}^2}$$

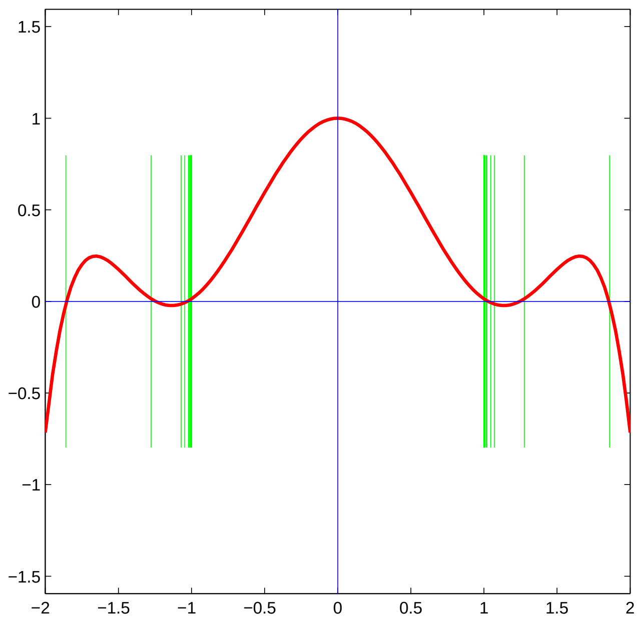
# Estimate of MINRES convergence

- eigenvalues in two symmetric intervals

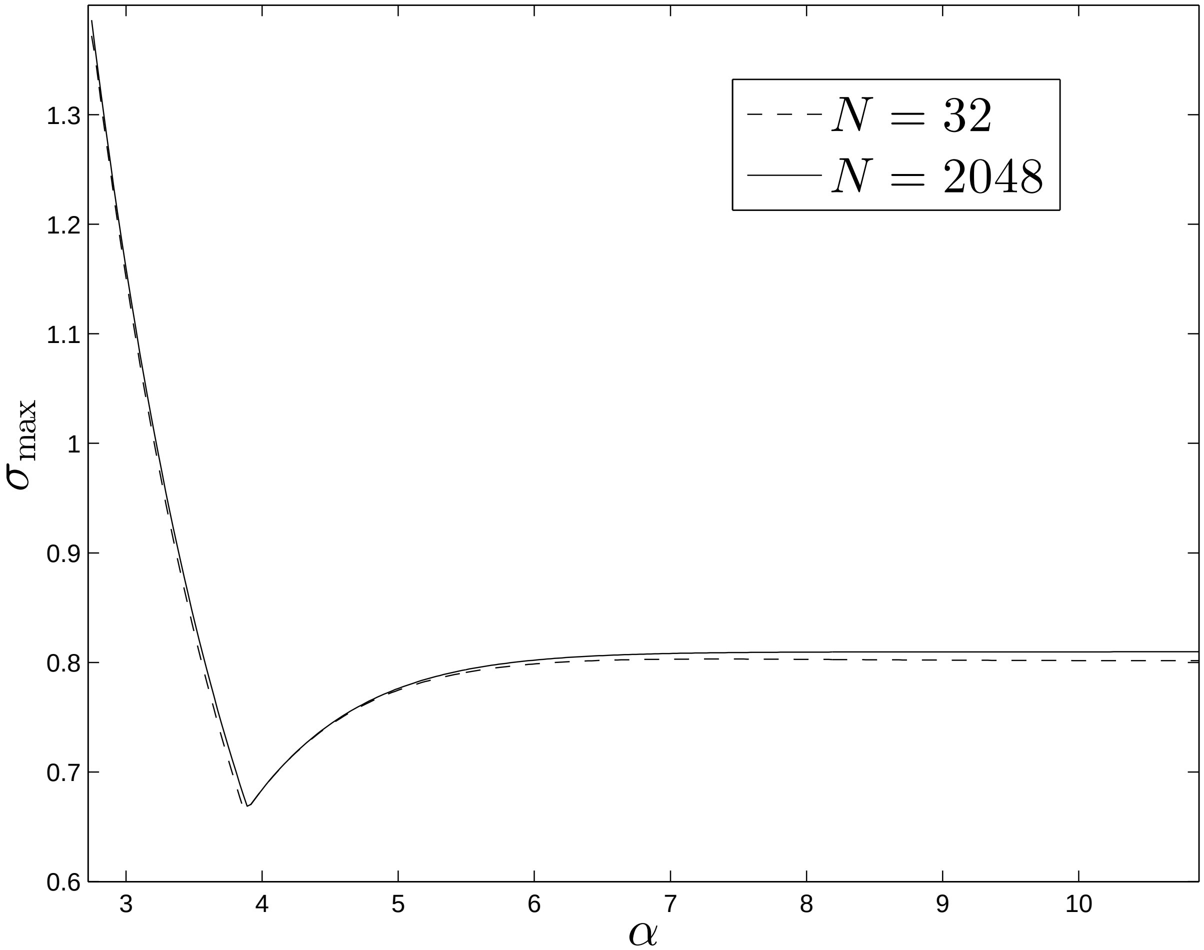
$$[-\beta, -1] \cup [1, \beta], \qquad \beta = \sqrt{1 + \sigma_{\max}^2}$$

- to achieve $\|\mathbf{r}_k\| \leq \epsilon \|\mathbf{r}_0\|$ need

$$k \simeq \frac{1}{2}\sqrt{1 + \sigma_{\max}^2}\, \ln\left(\frac{2}{\epsilon}\right)$$



# Largest singular value of $M$




- $\sigma_{\max}$ is essentially independent of $N$
- expect a constant number of MINRES iterations

# Iteration Counts

| | off state ($\alpha = 0.5\alpha_c$) | | on state ($\alpha = 1.5\alpha_c$) | |
|---|---|---|---|---|
| $N$ | first step | last step | first step | last step |
| 32 | 4 | 1 | 5 | 7 |
| 64 | 4 | 1 | 5 | 7 |
| 128 | 4 | 1 | 5 | 7 |
| 256 | 4 | 1 | 5 | 7 |
| 512 | 4 | 1 | 5 | 7 |
| 1,024 | 4 | 1 | 5 | 7 |
| 2,048 | 4 | 1 | 5 | 7 |
| 4,096 | 4 | 1 | 5 | 7 |
| 8,192 | 4 | 1 | 5 | 7 |
| 16,384 | 4 | 1 | 5 | 7 |
| 32,768 | 4 | 1 | 5 | 7 |
| 65,536 | 4 | 1 | 5 | 7 |

# Other options?

- Several other sophisticated saddle point solvers available.
- Many would work here as $\tilde{A}$ and $C$ are easy to invert.
- Example: ideal constraint preconditioner

$$\mathcal{P} = \begin{bmatrix} \tilde{A} & \tilde{D} \\ \tilde{D}^T & -C \end{bmatrix}$$

with Projected Preconditioned Conjugate Gradient method (Dollar et al. 2006)

  - converges in one iteration at each Newton step

# Computing Time

- elapsed time in seconds (tic/toc)

| $N$ | full direct | reduced direct | ideal block | ideal constraint |
|---|---|---|---|---|
| 1,024 | 9.95e$-02$ | 9.70e$-02$ | 3.48e$-01$ | 3.08e$-01$ |
| 2,048 | 1.42e$-01$ | 1.36e$-01$ | 5.32e$-01$ | 8.35e$-01$ |
| 4,096 | 2.91e$-01$ | 2.79e$-01$ | 1.05e$+00$ | 2.73e$+00$ |
| 8,192 | 6.02e$-01$ | 5.90e$-01$ | 2.20e$+00$ | 9.74e$+00$ |
| 16,384 | 1.42e$+00$ | 1.29e$+00$ | 4.69e$+00$ | 3.80e$+01$ |
| 32,768 | 3.36e$+00$ | 2.75e$+00$ | 9.70e$+00$ | 8.25e$+02$ |
| 65,536 | 9.27e$+00$ | 7.41e$+00$ | 2.53e$+01$ | — |

# Non-“ideal” versions?

- Block systems can alaso be solved iteratively.
- Example: use a fixed number of PCG iterations with AMG preconditioner (HSL_MI20).

| | 1 PCG/AMG iteration | | | | 3 PCG/AMG iterations | | | |
|---|---|---|---|---|---|---|---|---|
| | off state $(\alpha = 0.5\alpha_c)$ | | on state $(\alpha = 1.5\alpha_c)$ | | off state $(\alpha = 0.5\alpha_c)$ | | on state $(\alpha = 1.5\alpha_c)$ | |
| $N$ | first | last | first | last | first | last | first | last |
| 32 | 6 | 5 | 7 | 9 | 4 | 1 | 5 | 7 |
| 128 | 7 | 6 | 7 | 9 | 4 | 1 | 5 | 7 |
| 512 | 7 | 6 | 8 | 9 | 4 | 1 | 5 | 7 |
| 2,048 | 7 | 6 | 8 | 9 | 4 | 2 | 5 | 7 |
| 8,192 | 7 | 6 | 8 | 9 | 4 | 2 | 5 | 7 |

# Non-“ideal” versions?

- Block systems can alaso be solved iteratively.
- Example: use a fixed number of PCG iterations with AMG preconditioner (HSL_MI20).

| | 1 PCG/AMG iteration | | | | 3 PCG/AMG iterations | | | |
|---|---|---|---|---|---|---|---|---|
| | off state ($\alpha = 0.5\alpha_c$) | | on state ($\alpha = 1.5\alpha_c$) | | off state ($\alpha = 0.5\alpha_c$) | | on state ($\alpha = 1.5\alpha_c$) | |
| $N$ | first | last | first | last | first | last | first | last |
| 32 | 6 | 5 | 7 | 9 | 4 | 1 | 5 | 7 |
| 128 | 7 | 6 | 7 | 9 | 4 | 1 | 5 | 7 |
| 512 | 7 | 6 | 8 | 9 | 4 | 1 | 5 | 7 |
| 2,048 | 7 | 6 | 8 | 9 | 4 | 2 | 5 | 7 |
| 8,192 | 7 | 6 | 8 | 9 | 4 | 2 | 5 | 7 |

# Summary and other issues

- Nullspace method plus ideal block preconditioner works very well for this simple 1D director model.
- We have also proposed a modified outer iteration (the Renormalized Newton Method) with $\mathbf{n}$ normalised onto the constraint manifold at each iterative step.
- Overall this gives an efficient solution algorithm for repeated solution of liquid crystal director models.

# Summary and other issues

- Nullspace method plus ideal block preconditioner works very well for this simple 1D director model.
- We have also proposed a modified outer iteration (the Renormalized Newton Method) with $\mathbf{n}$ normalised onto the constraint manifold at each iterative step.
- Overall this gives an efficient solution algorithm for repeated solution of liquid crystal director models.
- Nullspace ideas also apply for full 2D and 3D problems.
- Issues remain re how to precondition $\tilde{A}$ for these more general cases.
- Use of spectrally equivalent preconditioned iteration looks promising.

# Summary and other issues

- Nullspace method plus ideal block preconditioner works very well for this simple 1D director model.
- We have also proposed a modified outer iteration (the Renormalized Newton Method) with $\mathbf{n}$ normalised onto the constraint manifold at each iterative step.
- Overall this gives an efficient solution algorithm for repeated solution of liquid crystal director models.
- Nullspace ideas also apply for full 2D and 3D problems.
- Issues remain re how to precondition $\tilde{A}$ for these more general cases.
- Use of spectrally equivalent preconditioned iteration looks promising.

THANKS!