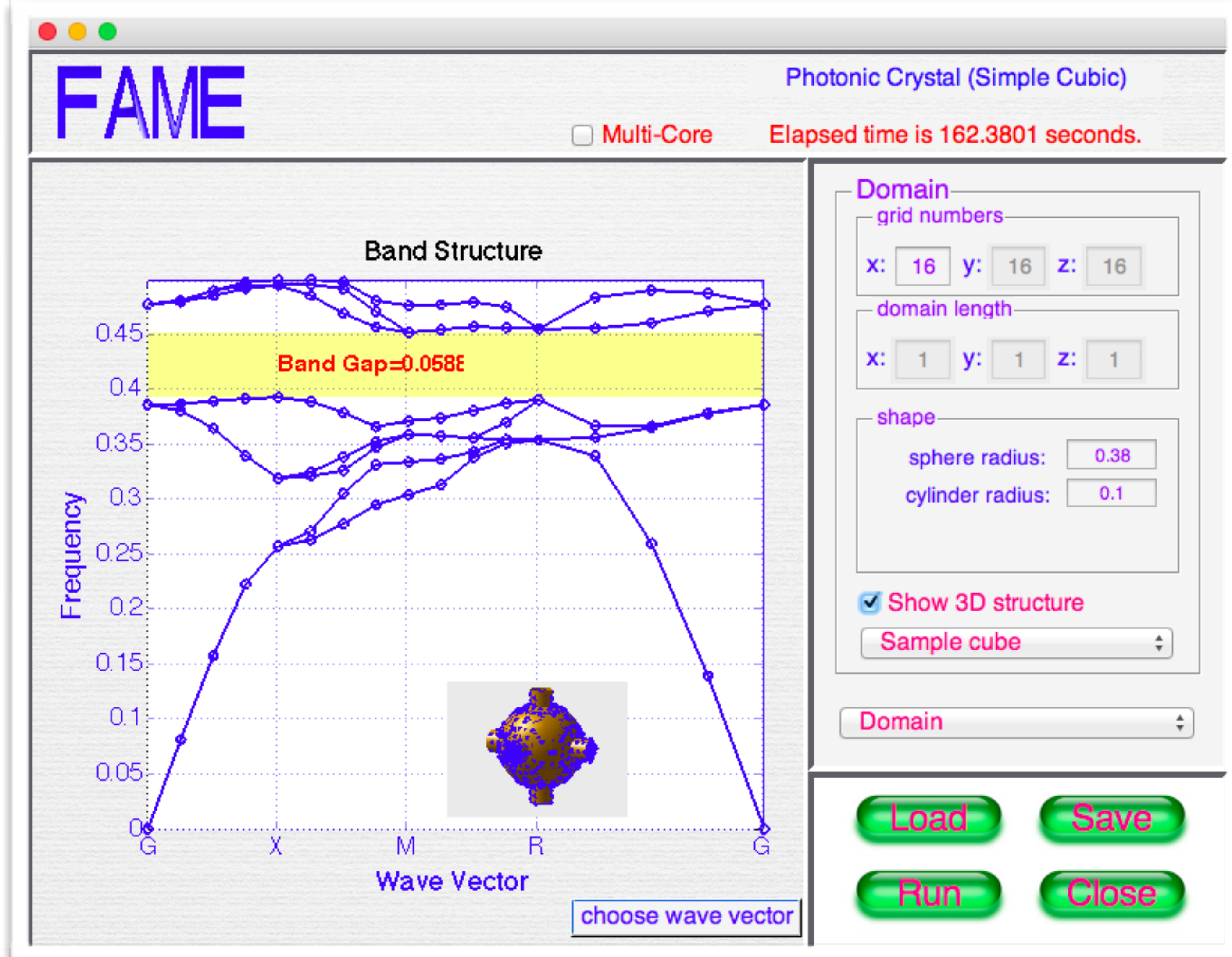
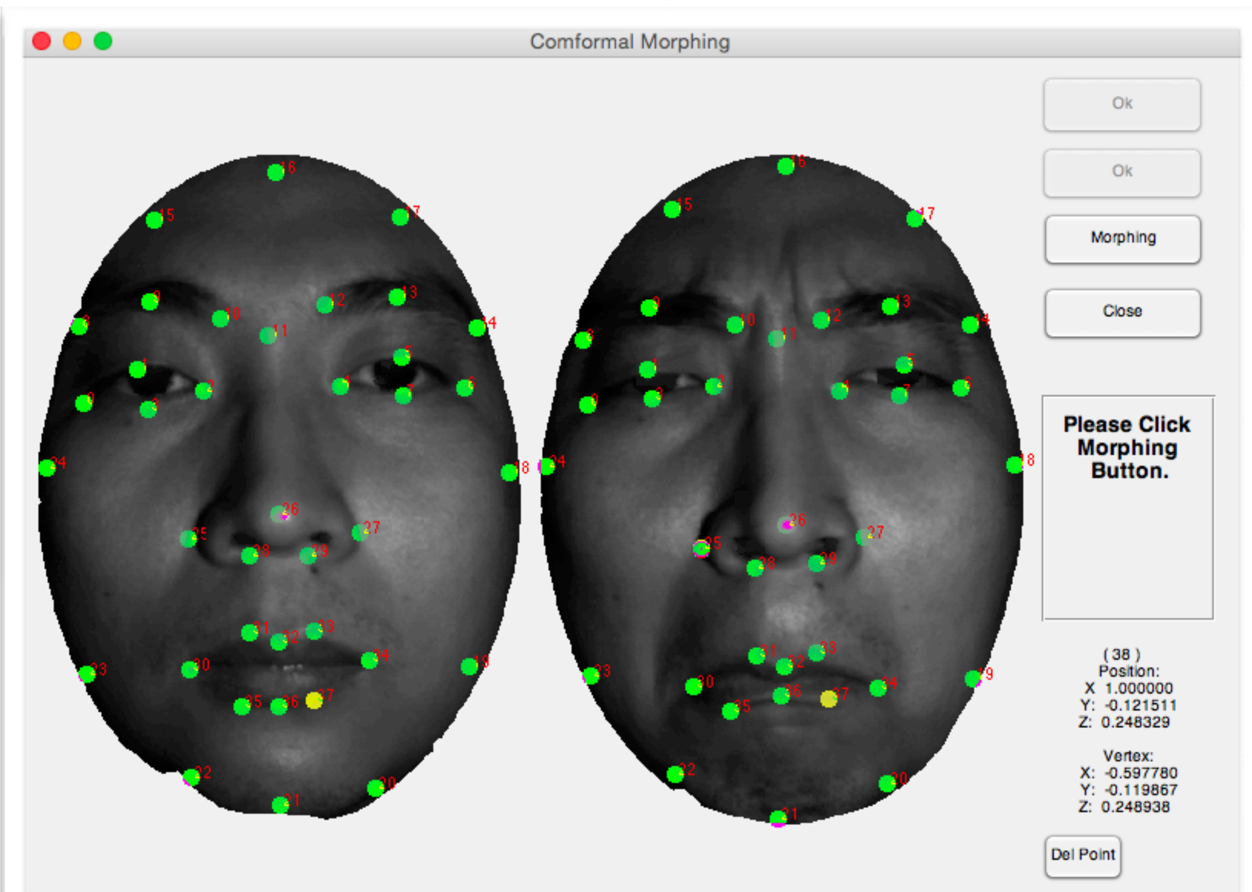
FAME-matlab Package: Fast Algorithm for Maxwell Equations

Tsung-Ming Huang

Modelling, Simulation and Analysis of Nonlinear Optics, NUK, September, 4-8, 2015


FAME
Photonic Crystal (Simple Cubic)
Multi-Core
Elapsed time is 162.3801 seconds.
Band Structure
Band Gap=0.058
Frequency
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
G X M R G
Wave Vector
choose wave vector
Domain
grid numbers
x: 16 y: 16 z: 16
domain length
x: 1 y: 1 z: 1
shape
sphere radius: 0.38
cylinder radius: 0.1
Show 3D structure
Sample cube
Domain
Load
Save
Run
Close


Comformal Morphing
Ok
Ok
Morphing
Close
Please Click Morphing Button.
(38)
Position:
X 1.000000
Y: -0.121511
Z: 0.248329
Vertex:
X: -0.597780
Y: -0.119867
Z: 0.248938
Del Point

FAME group

Wen-Wei Lin
Department of Applied Mathematics
National Chiao-Tung University

Weichung Wang
Department of Mathematics
National Taiwan University

Chien-Chih Huang(黃建智)
Department of Mathematics
National Taiwan Normal University

Han-En Hsieh(謝函恩)
Department of Mathematics
National Taiwan University

Fast Algorithm for Maxwell Eq

FAME

FAME.m

FAME.GPU

FAME.mpi

FAME

Eigen-solvers (JD, SIRA)

Photonic Crystals

Dispersive Metallic materials

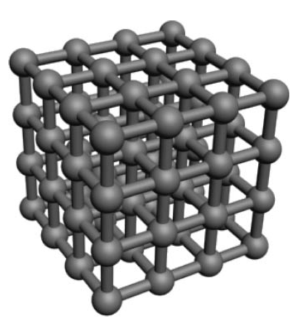
Complex materials

$$\nabla\times\nabla\times E(\mathbf{r})=\lambda\varepsilon(\mathbf{r})E(\mathbf{r})$$

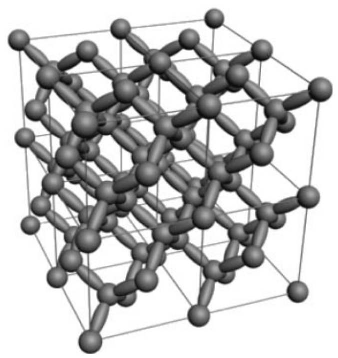
$$\nabla\times\nabla\times E(\mathbf{r})=\omega^2\varepsilon(\mathbf{r},\omega)E(\mathbf{r})$$

$$\begin{bmatrix}\nabla\times & 0\\ 0 & \nabla\times\end{bmatrix}\begin{bmatrix}E\\ H\end{bmatrix}=\mathrm{i}\omega\begin{bmatrix}\zeta & \mu\\ -\varepsilon & -\xi\end{bmatrix}\begin{bmatrix}E\\ H\end{bmatrix}$$

Simple Cubic



Face-Centered Cubic

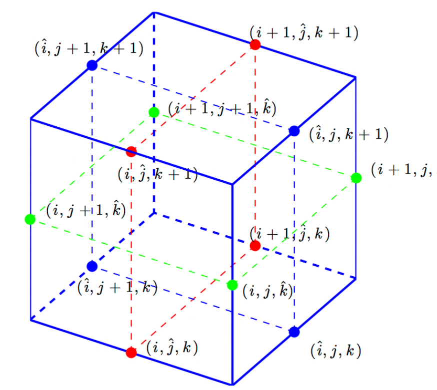


Generalized eigenvalue problems for 3D photonic crystal

$$\nabla \times \nabla \times E(\mathbf{r}) = \omega^2 \varepsilon(\mathbf{r}) E(\mathbf{r})$$

- Curl operator

$$\nabla \times E = \begin{bmatrix} 0 & -\frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & -\frac{\partial}{\partial x} \\ -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}$$

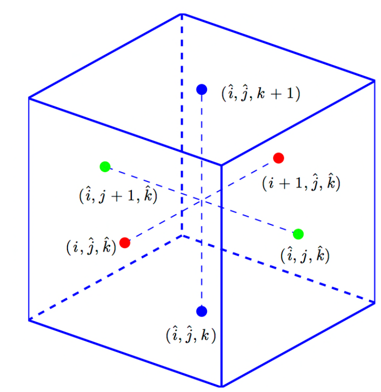


- Central edge points

$$\nabla \times H(\mathbf{r}) = \omega^2 \varepsilon(\mathbf{r}) E(\mathbf{r}) \quad \Rightarrow \quad C^* \mathbf{h} = \omega^2 B \mathbf{e}$$

- Central face points

$$\nabla \times E(\mathbf{r}) = H(\mathbf{r}) \quad \Rightarrow \quad C \mathbf{e} = \mathbf{h}$$



where

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix} \in \mathbb{C}^{3n \times 3n}$$

$$\boxed{C_1 = I_{n_2 n_3} \otimes K_1 \in \mathbb{C}^{n \times n}, C_2 = I_{n_3} \otimes K_2 \in \mathbb{C}^{n \times n}, C_3 = K_3 \in \mathbb{C}^{n \times n}}$$

- Resulting generalized eigenvalue problem

$$\boxed{\left(C^* C - \omega^2 B\right) \mathbf{x} \equiv \left(A - \lambda B\right) \mathbf{x} = 0}$$

with diagonal B

Finite Diff. Assoc. with Quasi-Periodic Cond.

$$K_1 = \frac{1}{\delta_x}\begin{bmatrix} -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1} & & & -1 \end{bmatrix} \in \mathbb{C}^{n_1\times n_1},$$

$$K_2 = \frac{1}{\delta_y}\begin{bmatrix} -I_{n_1} & I_{n_1} & & \\ & \ddots & \ddots & \\ & & -I_{n_1} & I_{n_1} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_2} J_2 & & & -I_{n_1} \end{bmatrix} \in \mathbb{C}^{(n_1 n_2)\times(n_1 n_2)},$$

$$K_3 = \frac{1}{\delta_z}\begin{bmatrix} -I_{n_1 n_2} & I_{n_1 n_2} & & \\ & \ddots & \ddots & \\ & & -I_{n_1 n_2} & I_{n_1 n_2} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_3} J_3 & & & -I_{n_1 n_2} \end{bmatrix} \in \mathbb{C}^{n\times n}$$

Finite Diff. Assoc. with Quasi-Periodic Cond.

$$E(\mathbf{r}+\mathbf{a}_\ell) = e^{\mathrm{i}2\pi\mathbf{k}\cdot\mathbf{a}_\ell} E(\mathbf{r})$$

$$K_1 = \frac{1}{\delta_x}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ e^{\imath 2\pi\mathbf{k}\cdot\mathbf{a}_1} & & & -1 \end{bmatrix} \in \mathbb{C}^{n_1\times n_1},$$

$$K_2 = \frac{1}{\delta_y}\begin{bmatrix} -I_{n_1} & I_{n_1} & & \\ & \ddots & \ddots & \\ & & -I_{n_1} & I_{n_1} \\ e^{\imath 2\pi\mathbf{k}\cdot\mathbf{a}_2} J_2 & & & -I_{n_1} \end{bmatrix} \in \mathbb{C}^{(n_1 n_2)\times(n_1 n_2)},$$

$$K_3 = \frac{1}{\delta_z}\begin{bmatrix} -I_{n_1 n_2} & I_{n_1 n_2} & & \\ & \ddots & \ddots & \\ & & -I_{n_1 n_2} & I_{n_1 n_2} \\ e^{\imath 2\pi\mathbf{k}\cdot\mathbf{a}_3} J_3 & & & -I_{n_1 n_2} \end{bmatrix} \in \mathbb{C}^{n\times n}$$

Finite Diff. Assoc. with Quasi-Periodic Cond.

$$K_1 = \frac{1}{\delta_x}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1} & & & -1 \end{bmatrix} \in \mathbb{C}^{n_1\times n_1},$$

$$K_2 = \frac{1}{\delta_y}\begin{bmatrix} -I_{n_1} & I_{n_1} & & \\ & \ddots & \ddots & \\ & & -I_{n_1} & I_{n_1} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_2} J_2 & & & -I_{n_1} \end{bmatrix} \in \mathbb{C}^{(n_1 n_2)\times(n_1 n_2)},$$

$$K_3 = \frac{1}{\delta_z}\begin{bmatrix} -I_{n_1 n_2} & I_{n_1 n_2} & & \\ & \ddots & \ddots & \\ & & -I_{n_1 n_2} & I_{n_1 n_2} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_3} J_3 & & & -I_{n_1 n_2} \end{bmatrix} \in \mathbb{C}^{n\times n}$$

Finite Diff. Assoc. with Quasi-Periodic Cond.

$$K_1 = \frac{1}{\delta_x}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1} & & & -1 \end{bmatrix} \in \mathbb{C}^{n_1\times n_1},$$

$$K_2 = \frac{1}{\delta_y}\begin{bmatrix} -I_{n_1} & I_{n_1} & & \\ & \ddots & \ddots & \\ & & -I_{n_1} & I_{n_1} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_2} J_2 & & & -I_{n_1} \end{bmatrix} \in \mathbb{C}^{(n_1 n_2)\times(n_1 n_2)},$$

$$K_3 = \frac{1}{\delta_z}\begin{bmatrix} -I_{n_1 n_2} & I_{n_1 n_2} & & \\ & \ddots & \ddots & \\ & & -I_{n_1 n_2} & I_{n_1 n_2} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_3} J_3 & & & -I_{n_1 n_2} \end{bmatrix} \in \mathbb{C}^{n\times n}$$

- For SC lattice

$$J_2 = I_{n_1}, \quad J_3 = I_{n_1 n_2}$$

Finite Diff. Assoc. with Quasi-Periodic Cond.

$$K_1 = \frac{1}{\delta_x}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1} & & & -1 \end{bmatrix} \in \mathbb{C}^{n_1\times n_1},$$

$$K_2 = \frac{1}{\delta_y}\begin{bmatrix} -I_{n_1} & I_{n_1} & & \\ & \ddots & \ddots & \\ & & -I_{n_1} & I_{n_1} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_2} J_2 & & & -I_{n_1} \end{bmatrix} \in \mathbb{C}^{(n_1 n_2)\times(n_1 n_2)},$$

$$K_3 = \frac{1}{\delta_z}\begin{bmatrix} -I_{n_1 n_2} & I_{n_1 n_2} & & \\ & \ddots & \ddots & \\ & & -I_{n_1 n_2} & I_{n_1 n_2} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_3} J_3 & & & -I_{n_1 n_2} \end{bmatrix} \in \mathbb{C}^{n\times n}$$

- For SC lattice

$$J_2 = I_{n_1}, \quad J_3 = I_{n_1 n_2}$$

- For FCC lattice

$$J_2 = \begin{bmatrix} 0 & e^{-\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1} I_{n_1/2} \\ I_{n_1/2} & 0 \end{bmatrix} \in \mathbb{C}^{n_1\times n_1},$$

$$J_3 = \begin{bmatrix} 0 & e^{-\imath 2\pi \mathbf{k}\cdot\mathbf{a}_2} I_{\frac{1}{3}n_2} \otimes I_{n_1} \\ I_{\frac{2}{3}n_2} \otimes J_2 & 0 \end{bmatrix} \in \mathbb{C}^{(n_1 n_2)\times(n_1 n_2)}$$

Power method

- Let (λ_i, x_i) for $i = 1, \ldots, n$ be the eigenpairs of A where $x_1, \ldots, x_n$ is linearly independent
- For any nonzero vector u

$$u = \alpha_1 x_1 + \cdots + \alpha_n x_n$$

- Since $A^k x_i = \lambda_i^k x_i$, we have

$$A^k u = \alpha_1 \lambda_1^k x_1 + \cdots + \alpha_n \lambda_n^k x_n$$

- If $|\lambda_1| > |\lambda_i|$ for i >1 and $\alpha_1 \neq 0$, then

$$\frac{1}{\lambda_1^k} A^k u = \alpha_1 x_1 + (\frac{\lambda_2}{\lambda_1})^k \alpha_2 x_2 + \cdots + \alpha_n (\frac{\lambda_n}{\lambda_1})^k x_n \to \alpha_1 x_1 \text{ as } k \to \infty$$

- Given shift value

$$\left\{(A - \sigma I)^{-1}\right\}^k u = \alpha_1 \left\{(\lambda_1 - \sigma)^{-1}\right\}^k x_1 + \cdots + \alpha_n \left\{(\lambda_n - \sigma)^{-1}\right\}^k x_n$$

Solving $\left(A-\lambda B\right)\mathbf{x}=0$

- Use shift-and-invert Lanczos method
- In each iteration of shift-and-invert Lanczos method, we need to solve

$$(A-\sigma B)y=b$$

- How to efficiently solve this linear system?

Solving linear system

$$(A - \sigma B)y = b$$

Solve $(A-\sigma B)y=b$

- Direct method (Gaussian elimination)

$$y=(A-\sigma B)\backslash b$$

- Iterative method
  - Matrix vector multiplication with $A-\sigma B$
  - Preconditioner M

```
sol = bicgstabl(coef_mtx, rhs, tol, maxit,
@(x)SSOR_prec(x, diag_coef_mtx, lower_L));
```

Solve $(A-\sigma B)y=b$

- Direct method (Gaussian elimination)

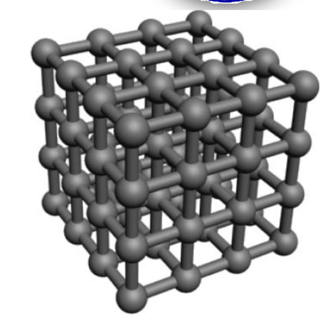
$$y=(A-\sigma B)\backslash b$$

- Iterative method
  - Matrix vector multiplication with $A-\sigma B$
  - Preconditioner M

```
sol = bicgstabl(coef_mtx, rhs, tol, maxit,
@(x)SSOR_prec(x, diag_coef_mtx, lower_L));
```

Demo performance

Eigen-decomp. of C_1, C_2, C_3 for SC lattice



- Define

$$D_{\mathbf{a},m} = \mathrm{diag}\left(1, e^{\theta_{\mathbf{a},m}}, \cdots, e^{(m-1)\theta_{\mathbf{a},m}}\right), \quad \Lambda_{\mathbf{a},m} = \mathrm{diag}\left(e^{\theta_{m,1}+\theta_{\mathbf{a},m}} - 1 \quad \cdots \quad e^{\theta_{m,m}+\theta_{\mathbf{a},m}} - 1 \right),$$

$$U_m = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{\theta_{m,1}} & e^{\theta_{m,2}} & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ e^{(m-1)\theta_{m,1}} & e^{(m-1)\theta_{m,2}} & \cdots & 1 \end{bmatrix} \in \mathbb{C}^{m\times m}, \quad \theta_{\mathbf{a},m} = \frac{\imath 2\pi \mathbf{k}\cdot\mathbf{a}}{m}, \quad \theta_{m,i} = \frac{\imath 2\pi i}{m}$$

- Define unitary matrix T as

$$\boxed{T = \frac{1}{\sqrt{n}}\left(D_{\mathbf{a}_3,n_3} \otimes D_{\mathbf{a}_2,n_2} \otimes D_{\mathbf{a}_1,n_1}\right)\left(U_{n_3} \otimes U_{n_2} \otimes U_{n_1}\right)}$$

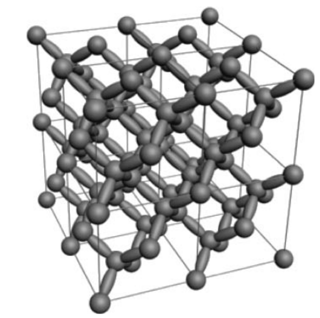
Then it holds that

$$C_1 T = \delta_x^{-1} T \left(I_{n_3} \otimes I_{n_2} \otimes \Lambda_{\mathbf{a}_1,n_1}\right) \equiv T\Lambda_1,$$

$$C_2 T = \delta_y^{-1} T \left(I_{n_3} \otimes \Lambda_{\mathbf{a}_2,n_2} \otimes I_{n_1}\right) \equiv T\Lambda_2,$$

$$C_3 T = \delta_z^{-1} T \left(\Lambda_{\mathbf{a}_3,n_3} \otimes I_{n_2} \otimes I_{n_1}\right) \equiv T\Lambda_3$$

Eigen-decomp. of C_1, C_2, C_3 for FCC lattice



- Define

$$\psi_{\mathbf{x}} = \frac{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1}{n_1}, \qquad D_{\mathbf{x}} = \operatorname{diag}\left(1, e^{\psi_{\mathbf{x}}}, \cdots, e^{(n_1-1)\psi_{\mathbf{x}}}\right),$$

$$\psi_{\mathbf{y},i} = \frac{\imath 2\pi}{n_2}\left\{\mathbf{k}\cdot\left(\mathbf{a}_2 - \frac{\mathbf{a}_1}{2}\right) - \frac{i}{2}\right\}, \qquad D_{\mathbf{y},i} = \operatorname{diag}\left(1, e^{\psi_{\mathbf{y},i}}, \cdots, e^{(n_2-1)\psi_{\mathbf{y},i}}\right),$$

$$\psi_{\mathbf{z},i+j} = \frac{\imath 2\pi}{n_3}\left\{\mathbf{k}\cdot\left(\mathbf{a}_3 - \frac{\mathbf{a}_1+\mathbf{a}_2}{3}\right) - \frac{i+j}{3}\right\}, \quad D_{\mathbf{z},i+j} = \operatorname{diag}\left(1, e^{\psi_{\mathbf{y},i+j}}, \cdots, e^{(n_3-1)\psi_{\mathbf{y},i+j}}\right)$$

$$\mathbf{x}_i = D_{\mathbf{x}} U_{n_1}(:,i), \quad \mathbf{y}_{i,j} = D_{\mathbf{y},i} U_{n_2}(:,j)$$

- Define unitary matrix T as

$$T = \frac{1}{\sqrt{n}}\begin{bmatrix} T_1 & T_2 & \cdots & T_{n_1} \end{bmatrix} \in \mathbb{C}^{n\times n}, \quad T_i = \begin{bmatrix} T_{i,1} & T_{i,2} & \cdots & T_{i,n_2} \end{bmatrix} \in \mathbb{C}^{n\times(n_2 n_3)},$$
$$T_{i,j} = \left(D_{\mathbf{z},i+j} U_{n_3}\right) \otimes \left(\mathbf{y}_{i,j} \otimes \mathbf{x}_i\right)$$

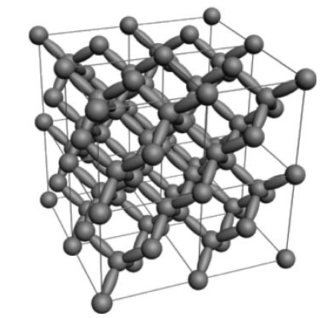
Then it holds that

$$C_1 T = T\left(\Lambda_{n_1} \otimes I_{n_2 n_3}\right) \equiv T\Lambda_1,$$

$$C_2 T = T\left(\left(\oplus_{i=1}^{n_1} \Lambda_{i,n_2}\right) \otimes I_{n_3}\right) \equiv T\Lambda_2,$$

$$C_3 T = T\left(\oplus_{i=1}^{n_1} \oplus_{j=1}^{n_2} \Lambda_{i,j,n_3}\right) \equiv T\Lambda_3$$

Eigen-decomp. of C_1, C_2, C_3 for FCC lattice



- Define

$$\psi_{\mathbf{x}} = \frac{\imath 2\pi \mathbf{k}\cdot \mathbf{a}_1}{n_1}, \qquad D_{\mathbf{x}} = \mathrm{diag}\left(1, e^{\psi_{\mathbf{x}}}, \cdots, e^{(n_1-1)\psi_{\mathbf{x}}}\right),$$

$$\psi_{\mathbf{y},i} = \frac{\imath 2\pi}{n_2}\left\{\mathbf{k}\cdot\left(\mathbf{a}_2 - \frac{\mathbf{a}_1}{2}\right) - \frac{i}{2}\right\}, \qquad D_{\mathbf{y},i} = \mathrm{diag}\left(1, e^{\psi_{\mathbf{y},i}}, \cdots, e^{(n_2-1)\psi_{\mathbf{y},i}}\right),$$

$$\psi_{\mathbf{z},i+j} = \frac{\imath 2\pi}{n_3}\left\{\mathbf{k}\cdot\left(\mathbf{a}_3 - \frac{\mathbf{a}_1+\mathbf{a}_2}{3}\right) - \frac{i+j}{3}\right\}, \quad D_{\mathbf{z},i+j} = \mathrm{diag}\left(1, e^{\psi_{\mathbf{y},i+j}}, \cdots, e^{(n_3-1)\psi_{\mathbf{y},i+j}}\right)$$

$$\mathbf{x}_i = D_{\mathbf{x}} U_{n_1}(:,i), \quad \mathbf{y}_{i,j} = D_{\mathbf{y},i} U_{n_2}(:,j)$$

- Define unitary matrix T as

$$T = \frac{1}{\sqrt{n}}\begin{bmatrix} T_1 & T_2 & \cdots & T_{n_1} \end{bmatrix} \in \mathbb{C}^{n\times n}, \quad T_i = \begin{bmatrix} T_{i,1} & T_{i,2} & \cdots & T_{i,n_2} \end{bmatrix} \in \mathbb{C}^{n\times(n_2 n_3)},$$

$$T_{i,j} = \left(D_{\mathbf{z},i+j} U_{n_3}\right) \otimes \left(\mathbf{y}_{i,j} \otimes \mathbf{x}_i\right)$$

Then it holds that

Demo performance

$$C_1 T = T\left(\Lambda_{n_1} \otimes I_{n_2 n_3}\right) \equiv T\Lambda_1,$$

$$C_2 T = T\left(\left(\oplus_{i=1}^{n_1} \Lambda_{i,n_2}\right) \otimes I_{n_3}\right) \equiv T\Lambda_2,$$

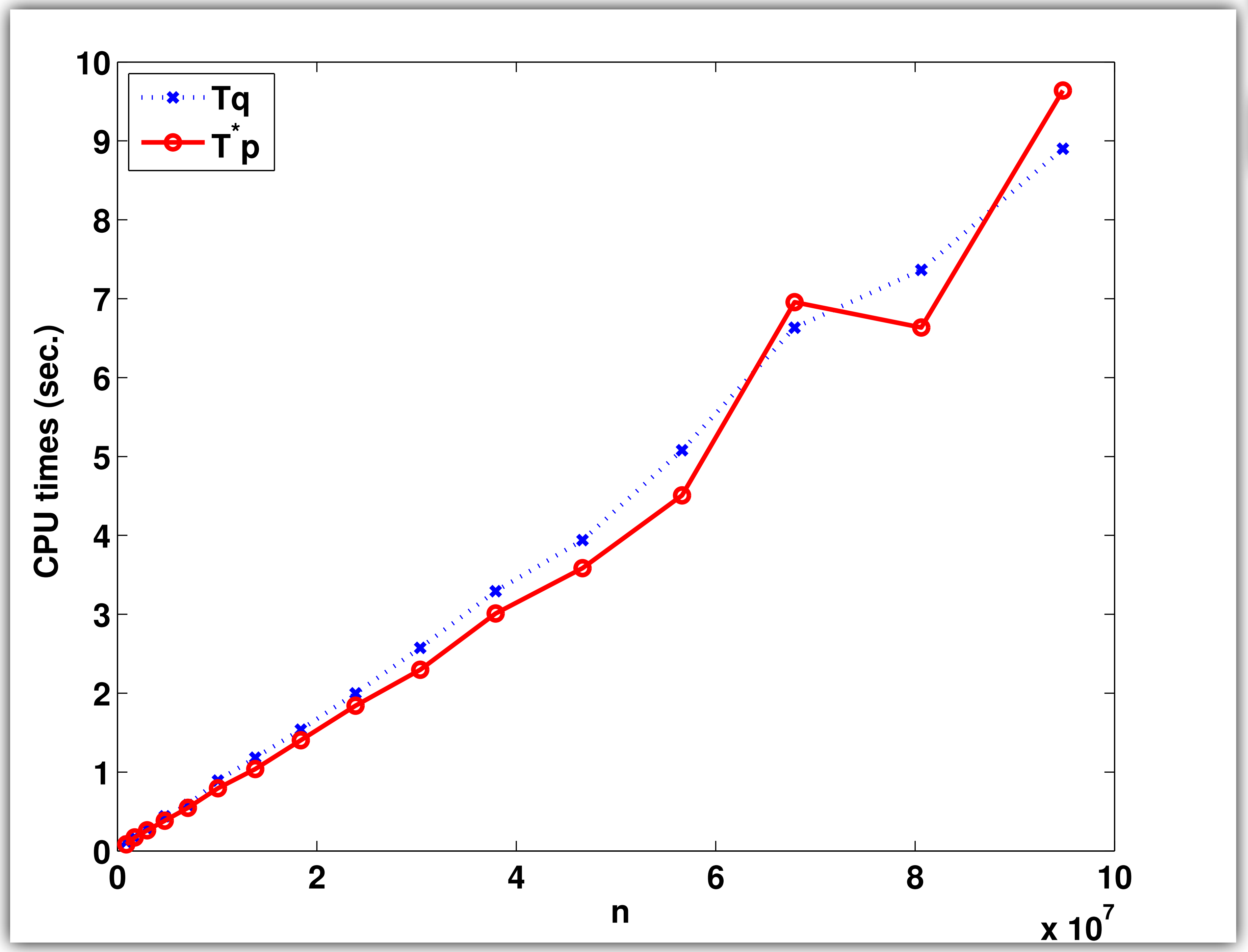
$$C_3 T = T\left(\oplus_{i=1}^{n_1} \oplus_{j=1}^{n_2} \Lambda_{i,j,n_3}\right) \equiv T\Lambda_3$$

CPU Times for T*p and Tq with FCC

MATLAB

T*p : fft

Tq : ifft



Solving preconditioning linear system

$$\boxed{(C^*C - \tau I)\mathbf{y} = \mathbf{d}}$$

Solving preconditioning linear system

$$\boxed{(C^* C - \tau I)\mathbf{y} = \mathbf{d}}$$

$$G = [C_1^\top, C_2^\top, C_3^\top]^\top$$

$$C^* C = I_3 \otimes \left(G^* G\right) - GG^*$$

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix}$$

Solving preconditioning linear system

$$(C^*C - \tau I)\mathbf{y} = \mathbf{d}$$

$$G = [C_1^\top, C_2^\top, C_3^\top]^\top \qquad C^*C = I_3 \otimes \left(G^*G\right) - GG^*$$

$$\left\{I_3 \otimes (G^*G) - \tau I\right\}\mathbf{y} = \mathbf{d} + GG^*\mathbf{y}$$

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix}$$

Solving preconditioning linear system

$$(C^*C - \tau I)\mathbf{y} = \mathbf{d}$$

$$G = [C_1^\top, C_2^\top, C_3^\top]^\top \qquad C^*C = I_3 \otimes \left(G^*G\right) - GG^*$$

$$\left\{I_3 \otimes (G^*G) - \tau I\right\}\mathbf{y} = \mathbf{d} + GG^*\mathbf{y}$$

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix} \qquad CG = 0 \qquad GG^*\mathbf{y} = -\tau^{-1}GG^*\mathbf{d}$$

Solving preconditioning linear system

$$(C^*C - \tau I)\mathbf{y} = \mathbf{d}$$

$$G = [C_1^\top, C_2^\top, C_3^\top]^\top \qquad C^*C = I_3 \otimes \left(G^*G\right) - GG^*$$

$$\left\{I_3 \otimes (G^*G) - \tau I\right\}\mathbf{y} = \mathbf{d} + GG^*\mathbf{y}$$

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix} \qquad CG = 0 \qquad GG^*\mathbf{y} = -\tau^{-1}GG^*\mathbf{d}$$

$$\left\{I_3 \otimes (G^*G) - \tau I\right\}\mathbf{y} = \mathbf{d} - \tau^{-1}GG^*\mathbf{d}$$

Solving preconditioning linear system

$$\boxed{(C^*C - \tau I)\mathbf{y} = \mathbf{d}}$$

$$G = [C_1^\top, C_2^\top, C_3^\top]^\top \qquad C^*C = I_3 \otimes \left(G^*G\right) - GG^*$$

$$\boxed{\left\{I_3 \otimes (G^*G) - \tau I\right\}\mathbf{y} = \mathbf{d} + GG^*\mathbf{y}}$$

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix} \qquad CG = 0 \qquad GG^*\mathbf{y} = -\tau^{-1}GG^*\mathbf{d}$$

$$\boxed{\left\{I_3 \otimes (G^*G) - \tau I\right\}\mathbf{y} = \mathbf{d} - \tau^{-1}GG^*\mathbf{d}}$$

$$\Lambda_q = \Lambda_1^*\Lambda_1 + \Lambda_2^*\Lambda_2 + \Lambda_3^*\Lambda_3 \qquad C_1T = T\Lambda_1, \quad C_2T = T\Lambda_2, \quad C_3T = T\Lambda_3$$

Solving preconditioning linear system

$$(C^*C - \tau I)\mathbf{y} = \mathbf{d}$$

$G = [C_1^\top, C_2^\top, C_3^\top]^\top$ $\qquad$ $C^*C = I_3 \otimes (G^*G) - GG^*$

$$\left\{ I_3 \otimes (G^*G) - \tau I \right\} \mathbf{y} = \mathbf{d} + GG^*\mathbf{y}$$

$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix}$ $\qquad$ $CG = 0$ $\qquad$ $GG^*\mathbf{y} = -\tau^{-1}GG^*\mathbf{d}$

$$\left\{ I_3 \otimes (G^*G) - \tau I \right\} \mathbf{y} = \mathbf{d} - \tau^{-1}GG^*\mathbf{d}$$

$\Lambda_q = \Lambda_1^*\Lambda_1 + \Lambda_2^*\Lambda_2 + \Lambda_3^*\Lambda_3$ $\qquad$ $C_1T = T\Lambda_1, \quad C_2T = T\Lambda_2, \quad C_3T = T\Lambda_3$

$$\left(I_3 \otimes \Lambda_q - \tau I\right)\tilde{\mathbf{y}} = \left(I - \tau^{-1} \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{bmatrix} \begin{bmatrix} \Lambda_1^* & \Lambda_2^* & \Lambda_3^* \end{bmatrix} \right) \left(I_3 \otimes T\right)^* \mathbf{d}, \quad \mathbf{y} = \left(I_3 \otimes T\right)\tilde{\mathbf{y}}$$

Solving preconditioning linear system

Demo performance

$$(C^*C-\tau I)\mathbf{y}=\mathbf{d}$$

$$G=[C_1^\top, C_2^\top, C_3^\top]^\top \qquad C^*C=I_3\otimes\left(G^*G\right)-GG^*$$

$$\left\{I_3\otimes(G^*G)-\tau I\right\}\mathbf{y}=\mathbf{d}+GG^*\mathbf{y}$$

$$C=\begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix} \qquad CG=0 \qquad GG^*\mathbf{y}=-\tau^{-1}GG^*\mathbf{d}$$

$$\left\{I_3\otimes(G^*G)-\tau I\right\}\mathbf{y}=\mathbf{d}-\tau^{-1}GG^*\mathbf{d}$$

$$\Lambda_q=\Lambda_1^*\Lambda_1+\Lambda_2^*\Lambda_2+\Lambda_3^*\Lambda_3 \qquad C_1T=T\Lambda_1,\quad C_2T=T\Lambda_2,\quad C_3T=T\Lambda_3$$

$$\left(I_3\otimes\Lambda_q-\tau I\right)\tilde{\mathbf{y}}=\left(I-\tau^{-1}\begin{bmatrix}\Lambda_1\\ \Lambda_2\\ \Lambda_3\end{bmatrix}\begin{bmatrix}\Lambda_1^* & \Lambda_2^* & \Lambda_3^*\end{bmatrix}\right)\left(I_3\otimes T\right)^*\mathbf{d},\quad \mathbf{y}=\left(I_3\otimes T\right)\tilde{\mathbf{y}}$$

Preconditioner $M = C^*C - \tau I$

- Iterative solver with preconditioner M:

  sol = bicgstabl(coef_mtx, rhs, tol, maxit, @(vec)FFT_based_precond(vec, Lambda, tau, EigDecompDoubCurl_cell, fun_mtx_TH_prod_vec, fun_mtx_T_prod_vec));

Preconditioner $M = C^* C - \tau I$

- Iterative solver with preconditioner M:

  sol = bicgstabl(coef_mtx, rhs, tol, maxit, @(vec)FFT_based_precond(vec, Lambda, tau, EigDecompDoubCurl_cell, fun_mtx_TH_prod_vec, fun_mtx_T_prod_vec));

- Since

$$M^{-1}(A-\sigma B) = M^{-1}(A\text{-}\ \tau I + \tau I - \sigma B) = I + M^{-1}(\tau I - \sigma B)$$

  we have

$$\left\{ I + M^{-1}(\tau I - \sigma B) \right\} y = M^{-1} b$$

- No need to compute the matrix-vector multiplication involving A:

  sol = bicgstabl(@(vec)mtx_prod_vec_shift_invert_LS(vec, tau, Lambda_new, EigDecompDoubCurl_cell, mtx_B_sigma, fun_mtx_TH_prod_vec, fun_mtx_T_prod_vec), rhs, tol, maxit);

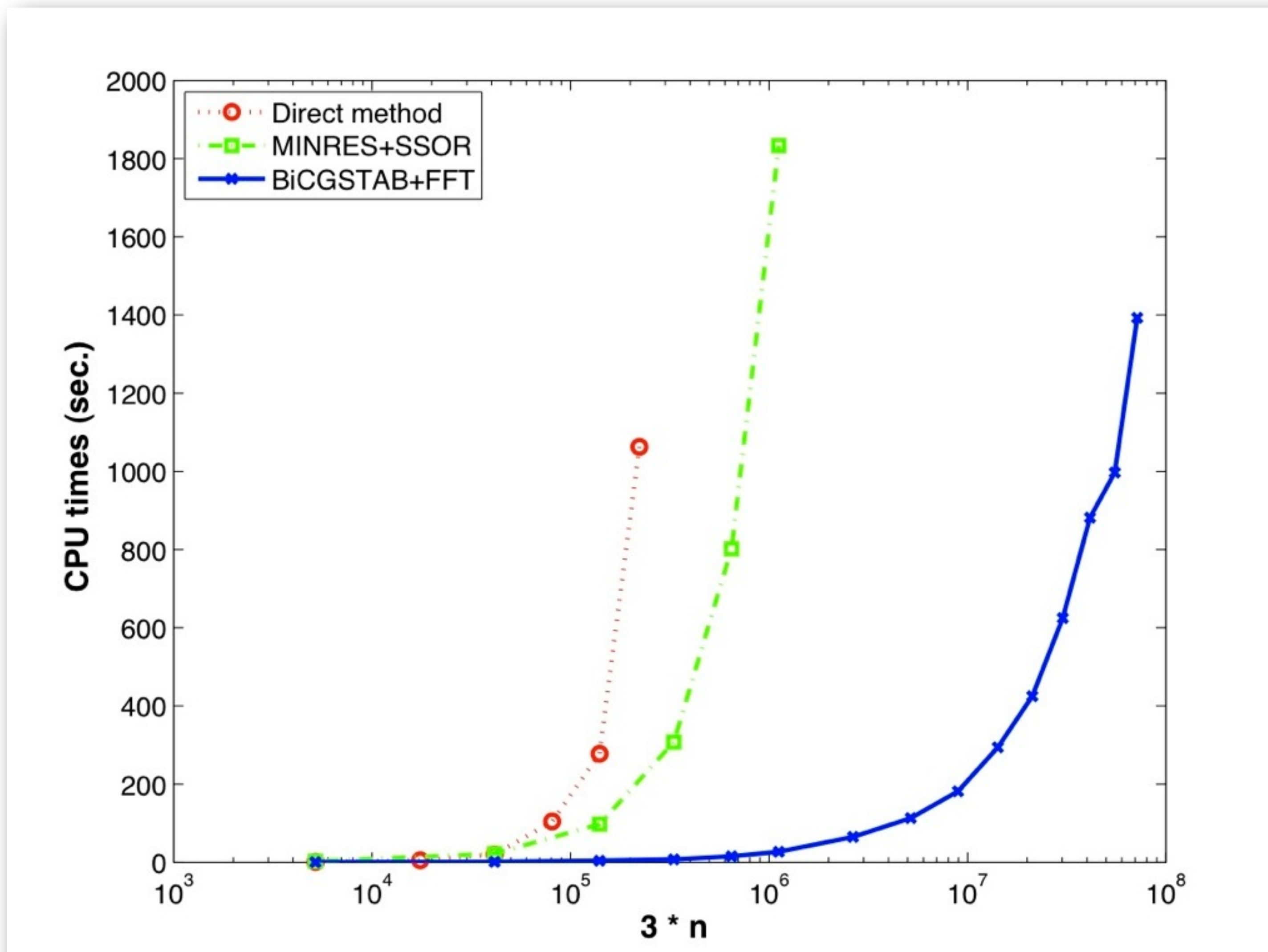
Challenge in Solving Linear System

- SC lattice (dim = 46875)

Index j	Jacobi	SSOR(0.8)	ICC(1)	ILU(1)	FFT
1	852	493	296	273	27
2	853	492	296	273	27
3	1,008	462	287	284	28

- FCC lattice

$$(A - \sigma B)\mathbf{z} = \mathbf{b}$$



Null-space free eigenvalue problem

Huge zero eigenvalues

$$Q^* A Q = \Lambda$$

- Eigen-decomposition

$$\boxed{\begin{bmatrix} Q_0 & Q \end{bmatrix}^* A \begin{bmatrix} Q_0 & Q \end{bmatrix} = \operatorname{diag}\left(0, \Lambda_q, \Lambda_q\right) \equiv \operatorname{diag}\left(0, \Lambda\right)}$$

where

$$\begin{bmatrix} Q_0 & Q \end{bmatrix} \coloneqq \left(I_3 \otimes T\right)\begin{bmatrix} \Pi_0 & \Pi_1 \end{bmatrix} \equiv \left(I_3 \otimes T\right)\begin{bmatrix} \Pi_{0,1} & \Pi_{1,1} & \Pi_{1,2} \\ \Pi_{0,2} & \Pi_{1,3} & \Pi_{1,4} \\ \Pi_{0,3} & \Pi_{1,5} & \Pi_{1,6} \end{bmatrix}$$

is unitary and $\Lambda_q = \Lambda_1^* \Lambda_1 + \Lambda_2^* \Lambda_2 + \Lambda_3^* \Lambda_3$

Huge zero eigenvalues

$$Q^* A Q = \Lambda$$

- Eigen-decomposition

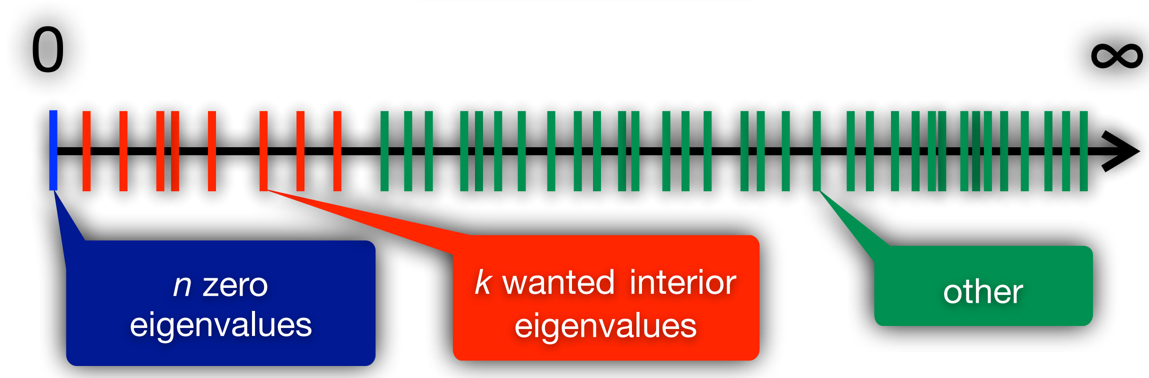
$$\boxed{\begin{bmatrix} Q_0 & Q \end{bmatrix}^* A \begin{bmatrix} Q_0 & Q \end{bmatrix} = \mathrm{diag}\left(0, \Lambda_q, \Lambda_q\right) \equiv \mathrm{diag}(0, \Lambda)}$$

where

$$\begin{bmatrix} Q_0 & Q \end{bmatrix} \coloneqq (I_3 \otimes T)\begin{bmatrix} \Pi_0 & \Pi_1 \end{bmatrix} \equiv (I_3 \otimes T)\begin{bmatrix} \Pi_{0,1} & \Pi_{1,1} & \Pi_{1,2} \\ \Pi_{0,2} & \Pi_{1,3} & \Pi_{1,4} \\ \Pi_{0,3} & \Pi_{1,5} & \Pi_{1,6} \end{bmatrix}$$

is unitary and $\Lambda_q = \Lambda_1^* \Lambda_1 + \Lambda_2^* \Lambda_2 + \Lambda_3^* \Lambda_3$

$$\boxed{A\mathbf{x} = \lambda B\mathbf{x}}$$



Huge zero eigenvalues

$$Q^* A Q = \Lambda$$

- Eigen-decomposition

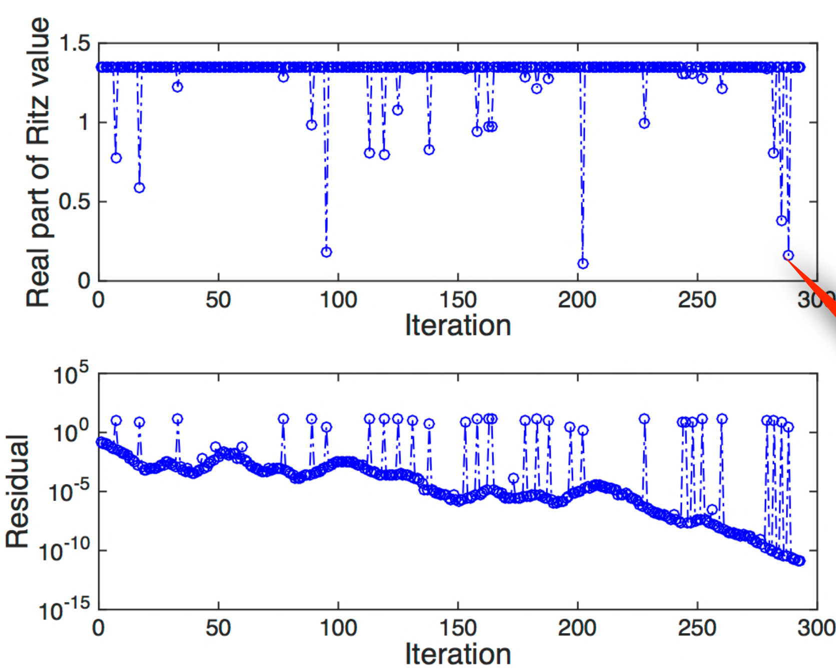
$$\boxed{\begin{bmatrix} Q_0 & Q \end{bmatrix}^* A \begin{bmatrix} Q_0 & Q \end{bmatrix} = \operatorname{diag}(0, \Lambda_q, \Lambda_q) \equiv \operatorname{diag}(0, \Lambda)}$$

where

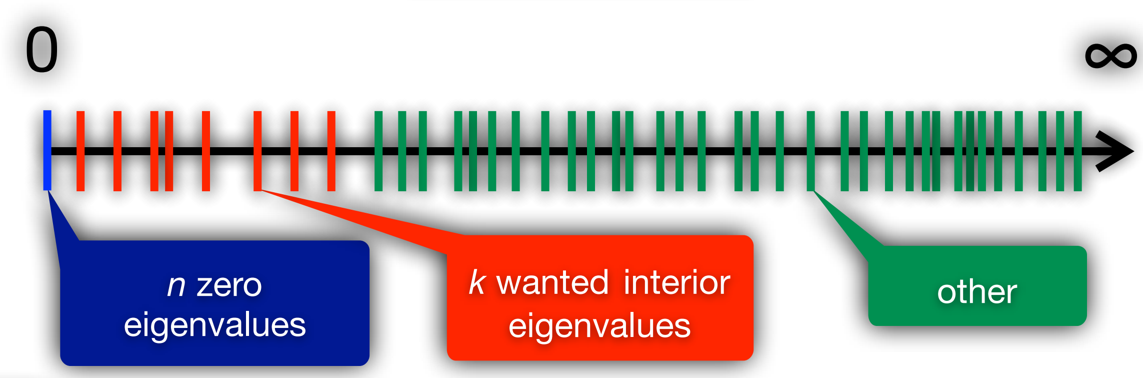
$$\begin{bmatrix} Q_0 & Q \end{bmatrix} \coloneqq (I_3 \otimes T)\begin{bmatrix} \Pi_0 & \Pi_1 \end{bmatrix} \equiv (I_3 \otimes T)\begin{bmatrix} \Pi_{0,1} & \Pi_{1,1} & \Pi_{1,2} \\ \Pi_{0,2} & \Pi_{1,3} & \Pi_{1,4} \\ \Pi_{0,3} & \Pi_{1,5} & \Pi_{1,6} \end{bmatrix}$$

is unitary and $\Lambda_q = \Lambda_1^* \Lambda_1 + \Lambda_2^* \Lambda_2 + \Lambda_3^* \Lambda_3$

$$\boxed{A\mathbf{x} = \lambda B\mathbf{x}}$$



Ritz values are dragged toward zero during the iteration



Null-space free method

$$Q^* A Q = \Lambda$$

- Theorem

$$\operatorname{span}\left(B^{-1} Q \Lambda^{1/2}\right) = \operatorname{span}\left\{\mathbf{x} \mid A\mathbf{x} = \lambda B\mathbf{x}, \lambda \neq 0\right\}$$

and

$$\left\{\lambda \neq 0 \mid A\mathbf{x} = \lambda B\mathbf{x}\right\} = \left\{\lambda \mid \Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}\right\}$$

Null-space free method

$$Q^* A Q = \Lambda$$

- Theorem

$$\mathrm{span}\left(B^{-1}Q\Lambda^{1/2}\right) = \mathrm{span}\left\{\mathbf{x} \mid A\mathbf{x} = \lambda B\mathbf{x}, \lambda \neq 0\right\}$$

and

$$\left\{\lambda \neq 0 \mid A\mathbf{x} = \lambda B\mathbf{x}\right\} = \left\{\lambda \mid \Lambda^{1/2}Q^* B^{-1} Q \Lambda^{1/2}\mathbf{u} = \lambda \mathbf{u}\right\}$$

- Null-space free SEP

$$A\mathbf{x} = \lambda B\mathbf{x} \quad \Rightarrow \quad \boxed{K\mathbf{u} \equiv \left(\Lambda^{1/2}Q^* B^{-1} Q \Lambda^{1/2}\right)\mathbf{u} = \lambda \mathbf{u}}$$

- Dim. of GEP and SEP are 3n and 2n, respectively
- GEP and SEP have same 2n nonzero eigenvalues.
  SEP has no zero eigenvalues

Null-space free method

$$Q^* A Q = \Lambda$$

- Theorem

$$\operatorname{span}\left(B^{-1} Q \Lambda^{1/2}\right) = \operatorname{span}\left\{\mathbf{x} \mid A\mathbf{x} = \lambda B\mathbf{x}, \lambda \neq 0\right\}$$

  and

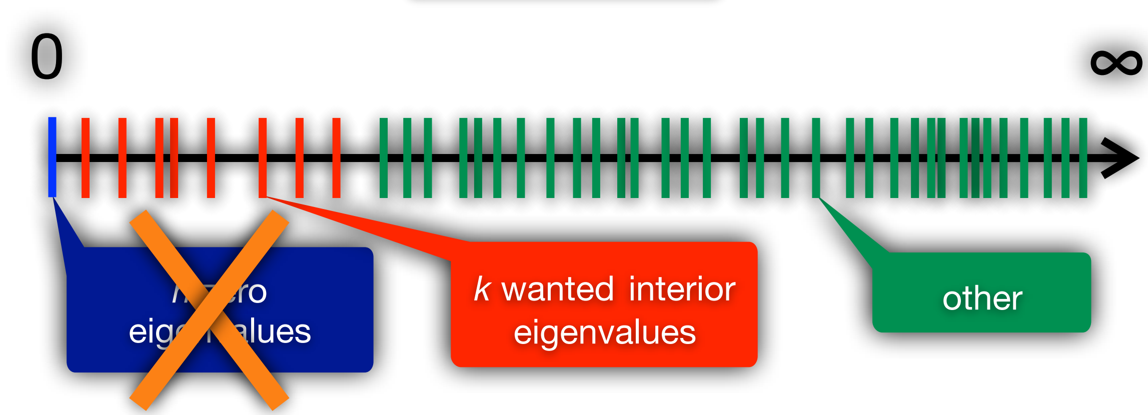
$$\left\{\lambda \neq 0 \mid A\mathbf{x} = \lambda B\mathbf{x}\right\} = \left\{\lambda \mid \Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}\right\}$$

- Null-space free SEP

$$A\mathbf{x} = \lambda B\mathbf{x} \quad \Rightarrow \quad \boxed{K\mathbf{u} \equiv \left(\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2}\right)\mathbf{u} = \lambda \mathbf{u}}$$

- Dim. of GEP and SEP are 3n and 2n, respectively
- GEP and SEP have same 2n nonzero eigenvalues.
  SEP has no zero eigenvalues

$$\boxed{K\mathbf{u} = \lambda\,\mathbf{u}}$$



Solving $\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}$

- Invert Lanczos method
- In each step, we need to solve a linear system

$$\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{v} = \mathbf{b}$$

Solving $\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}$

- Invert Lanczos method
- In each step, we need to solve a linear system

$$\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{v} = \mathbf{b}$$

- Solve LS by CG method
  sol = pcg(@(vec)NFSEP_mtx_prod_vec_Lambda(vec, EigDecompDoubCurl_cell, diag_B_eps, @(x)mtx_TH_prod_vec_SC(x, FFT_parameter), @(x)mtx_T_prod_vec_SC(x, FFT_parameter)), rhs, tol, maxit);

Solving $\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}$

- Invert Lanczos method

- In each step, we need to solve a linear system

$$\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{v} = \mathbf{b}$$

Demo performance

- Solve LS by CG method
  sol = pcg(@(vec)NFSEP_mtx_prod_vec_Lambda(vec, EigDecompDoubCurl_cell, diag_B_eps, @(x)mtx_TH_prod_vec_SC(x, FFT_parameter), @(x)mtx_T_prod_vec_SC(x, FFT_parameter)), rhs, tol, maxit);

Solving $\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}$

- Invert Lanczos method
- In each step, we need to solve a linear system

$$\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{v} = \mathbf{b}$$

Demo performance

- Solve LS by CG method
  sol = pcg(@(vec)NFSEP_mtx_prod_vec_Lambda(vec, EigDecompDoubCurl_cell, diag_B_eps, @(x)mtx_TH_prod_vec_SC(x, FFT_parameter), @(x)mtx_T_prod_vec_SC(x, FFT_parameter)), rhs, tol, maxit);
- Rewrite linear system as

$$Q^* B^{-1} Q \tilde{\mathbf{v}} = \Lambda^{-1/2} \mathbf{b}, \quad \mathbf{v} = \Lambda^{-1/2} \tilde{\mathbf{v}}$$

- Well condition number

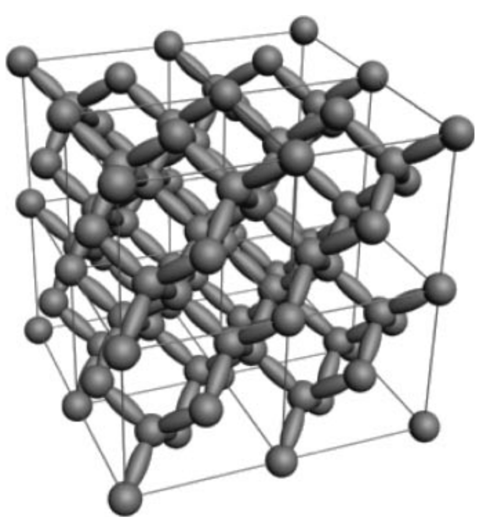
$$\kappa(Q^* B^{-1} Q) \le \kappa(B^{-1})$$

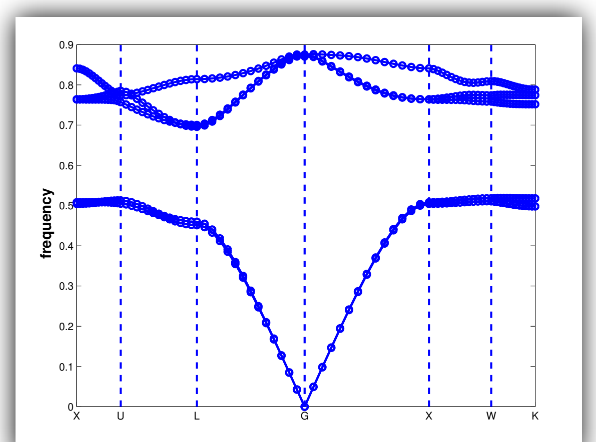
- Solve it by CG method

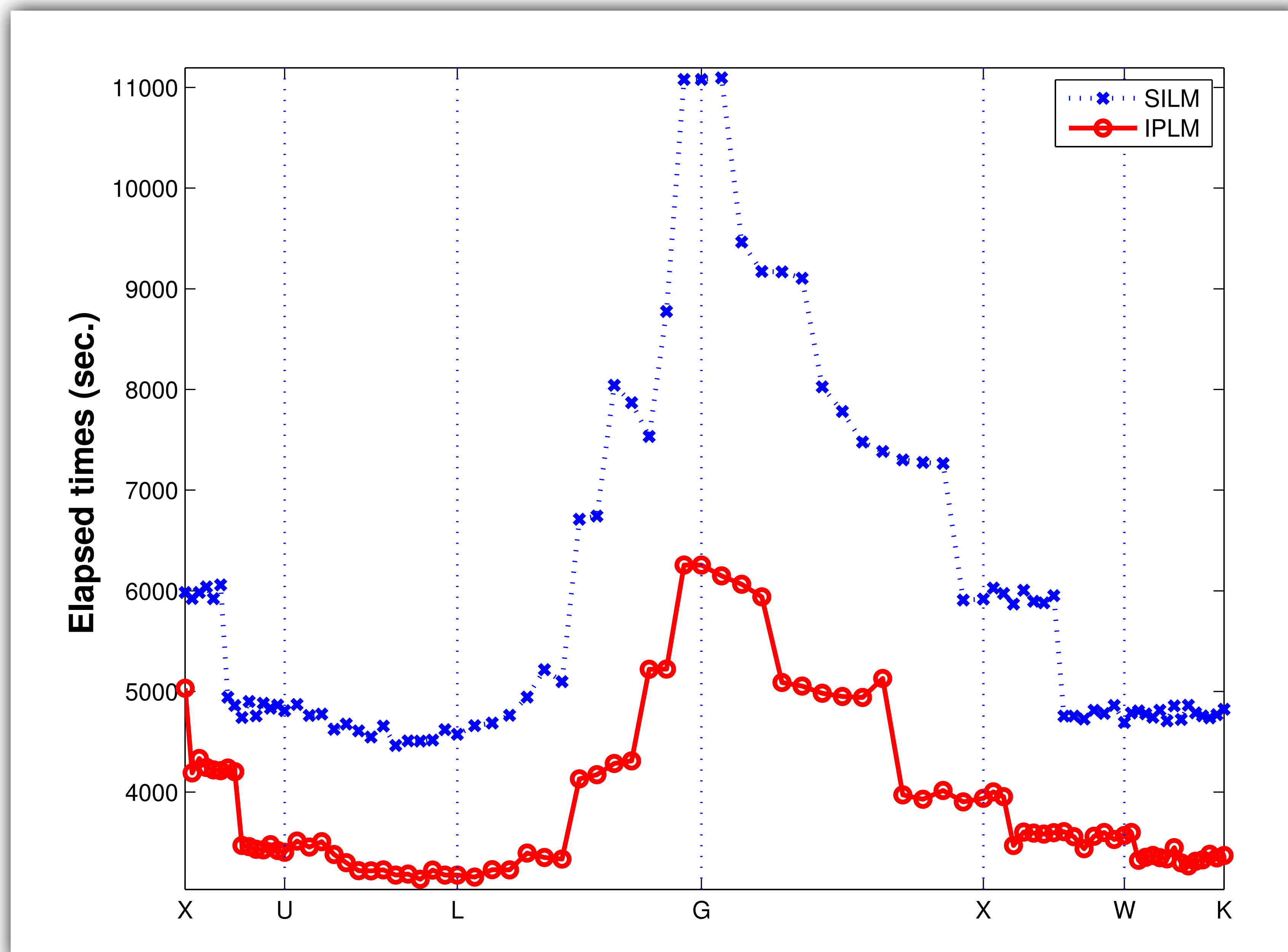
CPU Time Comparison

$$A\mathbf{x} = \lambda B\mathbf{x}$$

$$\Lambda^{1/2} Q^{*} B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}$$







Shift-Invert Residual Arnoldi method

Shift-Invert Residual Arnoldi method (SIRA)

- For a given search subspace V, let $(\theta, \tilde{\mathbf{z}})$ be an eigenpair of
$$V^*(\Lambda^{1/2}Q^*B^{-1}Q\Lambda^{1/2} - \lambda I)V\mathbf{z} = 0$$
and let $\tilde{\mathbf{x}} = V\tilde{\mathbf{z}}$ be the associated Ritz vector

- The new search direction v is chosen as
$$\boxed{\mathbf{v} = \left(\Lambda^{1/2}Q^*B^{-1}Q\Lambda^{1/2} - \sigma I\right)^{-1}\left[(\Lambda^{1/2}Q^*B^{-1}Q\Lambda^{1/2} - \theta I)\tilde{\mathbf{x}}\right] \equiv \left(\Lambda^{1/2}Q^*B^{-1}Q\Lambda^{1/2} - \sigma I\right)^{-1}\mathbf{r}}$$
where σ is a given shift value

- After re-orthogonalizing v against V, the vector is appended to V and one repeats this process until $(\theta, \tilde{\mathbf{x}})$ converges to the desired eigenpair.

CPU Time Comparison

$$\Lambda^{1/2} Q^* B^{-1} Q \Lambda^{1/2} \mathbf{u} = \lambda \mathbf{u}$$