ECE 695
Numerical Simulations
Lecture 4: Eigenproblems for Electro-optic Systems

Prof. Peter Bermel
January 18, 2017
Outline

• Electrostatic Potentials
• Solving $Ax = b$
• Spin arrays
• Solving eigenproblems
• Bandstructure problem
• Bloch’s theorem
• Photonic bandstructures
  – 1D
  – 2D
Electrostatic Potential

• Consider an array of charges governed by Gauss’ law:

\[ \nabla \cdot E = \rho / \epsilon_0 \]

• Using the definition of potential yields Poisson’s equation:

\[ -\nabla^2 \phi = \rho / \epsilon_0 \]

• Consider solving this equation for an arbitrary set of charges – what to do?
Electrostatic Potential Example

• Strictly speaking, continuous fields have an uncountable number of possible values, and cannot be evaluated numerically.

• Key is to transform from continuous field values to those on a grid.

• Increase resolution as needed.

\[ \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \varphi_7, \varphi_8, \varphi_9, \varphi_{10} \]
Electrostatic Potential Solution: 1D

- Approximate Laplacian in 1D with:

\[
\nabla^2 \varphi \approx \frac{\varphi_{i-1} - 2\varphi_i + \varphi_{i+1}}{h^2}
\]

- Where \( h \) is the grid spacing
- Sets up the linear algebra problem:

\[
\begin{pmatrix}
-2 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & -2 & 1 & \cdots & 0 & 0 & 0 \\
0 & 1 & -2 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -2 & 1 & 0 \\
0 & 0 & 0 & \cdots & 1 & -2 & 1 \\
0 & 0 & 0 & \cdots & 0 & 1 & -2 \\
\end{pmatrix}
\begin{pmatrix}
\varphi_1 \\
\varphi_2 \\
\varphi_3 \\
\vdots \\
\varphi_{N-2} \\
\varphi_{N-1} \\
\varphi_N
\end{pmatrix}
\begin{pmatrix}
\rho_1 \\
\rho_2 \\
\rho_3 \\
\vdots \\
\rho_{N-2} \\
\rho_{N-1} \\
\rho_N
\end{pmatrix}
\begin{pmatrix}
= \frac{h^2}{\varepsilon_0}
\end{pmatrix}
\]
Electrostatic Potential Solution: 1D

• In MATLAB, use:

```matlab
>> N=10; N=-2; J=1; A=diag(M*ones(N,1))+diag(J*ones(N-1,1),1)+diag(J*ones(N-1,1),-1)

A =
-2 1 0 0 0 0 0 0 0 0
1 -2 1 0 0 0 0 0 0 0
0 1 -2 1 0 0 0 0 0 0
0 0 1 -2 1 0 0 0 0 0
0 0 0 1 -2 1 0 0 0 0
0 0 0 0 1 -2 1 0 0 0
0 0 0 0 0 1 -2 1 0 0
0 0 0 0 0 0 1 -2 1 0
0 0 0 0 0 0 0 1 -2 1
0 0 0 0 0 0 0 0 1 -2

>> rho=rand(10,1)

rho =
0.1190
0.4984
0.9697
0.3404
0.5853
0.2230
0.7513
0.2881
0.5060
0.6991
```
How to Solve $Ax = b$?

- Direct matrix inversion: $x = A^{-1}b$
- Linear algebra software packages:
  - Linear Algebra Package (LAPACK)
  - MATLAB backslash operator
Linear Algebra Package (LAPACK)

• Implements many linear algebra techniques:
  – Linear programming/least squares
  – Matrix decompositions/factorizations
  – Eigenvalues

• Designed in 1992 to deal with special cases efficiently

<table>
<thead>
<tr>
<th>Matrix type</th>
<th>full</th>
<th>banded</th>
<th>packed</th>
<th>tridiag</th>
<th>generalized problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>ge</td>
<td>gb</td>
<td>sp</td>
<td>gt</td>
<td>gg</td>
</tr>
<tr>
<td>symmetric</td>
<td>sy</td>
<td>sb</td>
<td>hp</td>
<td>st</td>
<td>st</td>
</tr>
<tr>
<td>Hermitian</td>
<td>he</td>
<td>hb</td>
<td>pp</td>
<td>pt</td>
<td>pt</td>
</tr>
<tr>
<td>SPD / HPD</td>
<td>po</td>
<td>pb</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>triangular</td>
<td>tr</td>
<td>tb</td>
<td>tp</td>
<td></td>
<td>tg</td>
</tr>
<tr>
<td>upper Hessenberg</td>
<td>hs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trapezoidal</td>
<td>tz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>orthogonal</td>
<td>or</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unitary</td>
<td>un</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diagonal</td>
<td>di</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bidiagonal</td>
<td>bd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Electrostatic Potential Solution: 1D

- MATLAB: use backslash operator to solve problems of the form $A \cdot x = b$

```
>> phi=A\rho
phi =
-2.3498
-4.5806
-6.3131
-7.0858
-7.5181
-7.3651
-6.9884
-5.8604
-4.4772
-2.5882
```

Electric potential

Charge distribution
Electrostatic Potential Solution: 2D

- Connections in two directions creates a total of 5 non-vanishing diagonals in our linear algebra problem:

\[
\begin{pmatrix}
-4 & 1 & 0 & & & 1 & 0 & 0 \\
1 & -4 & 1 & & & 0 & 1 & 0 \\
0 & 1 & -4 & & & 0 & 0 & 1 \\
& & & \ddots & & & \ddots & \\
1 & 0 & 0 & & -4 & 1 & 0 \\
0 & 1 & 0 & & & 1 & -4 & 1 \\
0 & 0 & 1 & & 0 & 1 & -4 \\
\end{pmatrix}
\begin{pmatrix}
\phi_1 \\
\phi_2 \\
\phi_3 \\
\vdots \\
\phi_{N-2} \\
\phi_{N-1} \\
\phi_N \\
\end{pmatrix}
= \frac{h^2}{\varepsilon_o}
\begin{pmatrix}
\rho_1 \\
\rho_2 \\
\rho_3 \\
\vdots \\
\rho_{N-2} \\
\rho_{N-1} \\
\rho_N \\
\end{pmatrix}
\]
Electrostatic Potential Solution: 2D

Charge distribution in 2D (7x7 grid)

Electrostatic potential in 2D (7x7 grid)
Spin Array Example

- Consider an array of spins \( \{\sigma_i\} \), coupled by an exchange interaction

- Ising-model Hamiltonian is given by:

\[
\mathcal{H} = \sum_i M_i \sigma_i + \sum_{\langle ij \rangle} J_{ij} \sigma_i \sigma_j
\]

- The brackets often are interpreted to mean nearest-neighbor interactions only
Spin Array Solution: 1D

- Convert Hamiltonian into matrix, assuming nearest neighbor interaction only:

\[
\mathcal{H} = \begin{pmatrix}
M & J & 0 & 0 & 0 & 0 & 0 \\
J & M & J & \cdots & 0 & 0 & 0 \\
0 & J & M & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & M & J & 0 \\
0 & 0 & 0 & \cdots & J & M & J \\
0 & 0 & 0 & \cdots & J & M & J \\
0 & 0 & 0 & 0 & J & M & M
\end{pmatrix}
\]

- Use Schrodinger equation to set up eigenproblem:

\[
\mathcal{H}\Psi = E\Psi
\]

- Choose basis spinor wavefunction:

\[
\Psi = (\sigma_1, \sigma_2, \cdots, \sigma_N)^T
\]
Spin Array Solution: 1D

- In MATLAB, use the following:

```matlab
>> N=6; M=2; J=1; A=diag(M*ones(N,1))+diag(J*ones(N-1,1),1)+diag(J*ones(N-1,1),-1)
A =
    2  1  0  0  0  0  0
    1  2  1  0  0  0  0
    0  1  2  1  0  0  0
    0  0  1  2  1  0  0
    0  0  0  1  2  1  0
    0  0  0  0  1  2  0
    0  0  0  0  0  1  2
```
How to Solve $Ax = \lambda x$?

- Eigenproblem Solution Techniques
  - Power Methods
  - Inverse Iteration
  - Atomic Transformations
  - Factorization Methods

- Linear algebra software packages
  - Linear Algebra Package (LAPACK)
  - MATLAB `eigs` package
Power Method

\[ \phi_1 \]
Inverse Iteration

\[ \phi_3 \]

Error vs. iteration number
Inverse Iteration Challenges

φ

5
Transformation Methods

\( \phi_6 \)
Jacobi Transformations

\[ A' = P_{pq}^T \cdot A \cdot P_{pq} \]

\[ S' = S - 2|a_{pq}|^2 \]
Householder Transformation

- Approximate Laplacian in 1D with:
  \[ \nabla^2 \varphi \approx \varphi_{i-1} - 2\varphi_i + \varphi_{i+1} \]

- Where
  \[ P = 1 - 2w \cdot w^T \]
  \[ \begin{bmatrix} a_{11} & k & 0 & \cdots & \cdots & 0 \\ k & 0 & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix} \]

- Sets up the linear algebra:
  \[ \begin{bmatrix} \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_{N-2} \\ \varphi_{N-1} \\ \varphi_N \end{bmatrix} = \frac{h^2}{\epsilon_0} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_{N-2} \\ \rho_{N-1} \\ \rho_N \end{bmatrix} \]
Factorization in Eigenproblems

• Most common approach known as QR method

\[ A = Q \cdot R \quad A' = R \cdot Q \quad A' = Q^T \cdot A \cdot Q \]

• Can also do the same with \( A = Q \cdot L \)

• QL algorithm:
  - Use Householder algorithm to construct \( Q_k \)
  - Factorize: \( A_k = Q_k L_k \)
  - Rearrange: \( A_{k+1} = L_k Q_k = Q_k^T A_k Q_k \)
Spin Array Solution: 1D

```matlab
>> [V,D]=eigs(full(A),N)
Warning: For real symmetric problems, must have number of eigenvalues k < n.
Using eig(full(A)) instead.
> In eigs>checkInputs at 926
  In eigs at 94

V =

0.2319  -0.4179   0.5211  -0.5211  -0.4179   0.2319
0.4179  -0.5211   0.2319   0.2319   0.5211  -0.4179
0.5211  -0.2319  -0.4179   0.4179  -0.2319   0.5211
0.5211   0.2319  -0.4179  -0.4179  -0.2319  -0.5211
0.4179   0.5211  -0.2319  -0.2319   0.5211   0.4179
0.2319   0.4179   0.5211   0.5211  -0.4179  -0.2319

D =

3.8019   0     0     0     0     0     0
0   3.2470  0     0     0     0     0
0     0  2.4450  0     0     0     0
0     0     0  1.5550  0     0     0
0     0   0     0  0.7530  0     0
0     0   0   0   0   0  0.1981
```
Spin Array Solution: 2D

- Convert Hamiltonian into matrix, assuming nearest neighbor interactions in 2 directions:

\[ H = \begin{pmatrix}
M & J & 0 & J & 0 & 0 \\
J & M & J & \cdots & 0 & J & 0 \\
0 & J & M & 0 & 0 & J \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
J & 0 & 0 & M & J & 0 \\
0 & J & 0 & \cdots & J & M & J \\
0 & 0 & J & 0 & J & M
\end{pmatrix} \]

- Use Schrodinger equation to set up eigenproblem:

\[ H\Psi = E\Psi \]

- Choose basis spinor wavefunction:

\[ \Psi = (\sigma_1, \sigma_2, \ldots, \sigma_N)^T \]
Spin Array Solution: 2D

- In MATLAB, use the following:

```matlab
N=9; M=2; J=1; A=diag(M*ones(N,1))+diag(J*ones(N-1,1),1)+diag(J*ones(N-1,1),-1);
A=A+diag(J*ones(N-sqrt(N),1),sqrt(N))+diag(J*ones(N-sqrt(N),1),-sqrt(N))
```

\[ A = \]

```
2   1   0   1   0   0   0   0   0
1   2   1   0   1   0   0   0   0
0   1   2   1   0   1   0   0   0
1   0   1   2   1   0   1   0   0
0   1   0   1   2   1   0   1   0
0   0   1   0   1   2   1   0   1
0   0   0   1   0   1   2   1   0
0   0   0   0   1   0   1   2   1
0   0   0   0   0   1   0   1   2
```
Spin Array Solution: 2D

```matlab
>> [V,D]=eig(full(A))

V =
   -0.2137  -0.4253  -0.2629  -0.1941  0.5774  -0.1941  0.2629  0.4253  0.2137
   0.2985   0.4253  -0.2629  0.4011 -0.0000 -0.4011 -0.2629  0.4253  0.2985
 -0.3362  -0.2629   0.4253  0.3701 -0.0000  0.3701 -0.4253  0.2629  0.3362
  0.4011   0.2629   0.4253 -0.2985 -0.0000  0.2985  0.4253  0.2629  0.4011
 -0.4274   0.0000   0.0000 -0.3882 -0.5774 -0.3882 -0.0000  0.0000  0.4274
  0.4011  -0.2629  -0.4253 -0.2985  0.0000  0.2985 -0.4253 -0.2629  0.4011
 -0.3362   0.2629  -0.4253  0.3701  0.0000  0.3701  0.4253 -0.2629  0.3362
  0.2985  -0.4253   0.2629  0.4011 -0.0000 -0.4011  0.2629 -0.4253  0.2985
 -0.2137   0.4253   0.2629 -0.1941  0.5774 -0.1941 -0.2629 -0.4253  0.2137

D =
  -1.2742   0    0    0    0    0    0    0    0
    0  0.3820   0    0    0    0    0    0    0
    0    0  1.3820   0    0    0    0    0    0
    0    0    0  1.4710   0    0    0    0    0
    0    0    0    0  2.0000   0    0    0    0
    0    0    0    0    0  2.5290   0    0    0
    0    0    0    0    0    0  2.6180   0    0
    0    0    0    0    0    0    0  3.6180   0
    0    0    0    0    0    0    0    0  5.2742
```
Bandstructure Problem

• Amounts to solving an eigenvalue equation for a system with discrete translational symmetry

• Examples include:
  – Electronic bandstructure: \[ \left[ -\frac{\hbar^2}{2m} \nabla^2 + V(x) \right] \Psi(x) = \hbar \omega \Psi(x) \]
  – Photonic bandstructure: \[ \nabla \times \left[ \epsilon^{-1} (\nabla \times H) \right] = \left( \frac{\omega}{c} \right)^2 H \]
  – Phononic bandstructure: \[ \nabla \times \left[ C (\nabla \times u) \right] = -\rho \omega^2 u \]
Next Class

• Continue formulating eigenproblems for electro-optic systems, with emphasis on band structures
• Continue reviewing Joannopoulos, Chapter 2 and Appendix D