ECE 595, Section 10
Numerical Simulations
Lecture 9: Programming for Linear Algebra

Prof. Peter Bermel

January 28, 2013

Outline

- Recap from Friday
- Application Examples
  - Electrostatic potential (Poisson's equation)
    - 1D array of charge
    - 2D grid of charge
  - Arrays of interacting spins
    - 1D interaction along a chain
    - 2D nearest-neighbor coupling

Recap from Friday

- Eigenproblem Solution Techniques
  – Power Methods
  – Inverse Iteration
  – Atomic Transformations
  – Factorization Methods
- Linear algebra software packages
  – Basic Linear Algebra Subroutines (BLAS)
  – Linear Algebra Package (LAPACK)

Electrostatic Potential Example

- Consider an array of charges governed by Gauss' law:

$$\nabla \cdot E = \rho/\epsilon_o$$

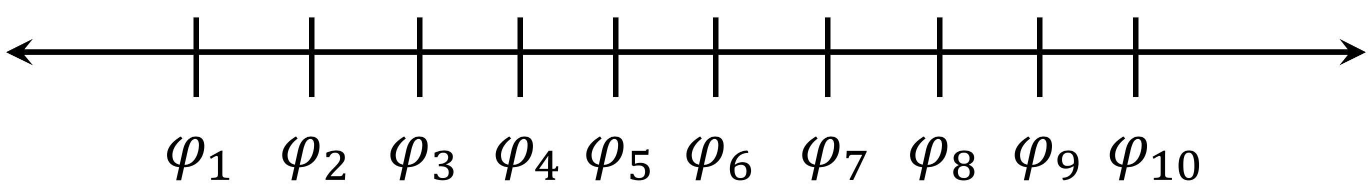
- Using the definition of potential yields Poisson's equation:

$$-\nabla^2 \varphi = \rho/\epsilon_o$$

- Consider solving this equation for an arbitrary set of charges – what to do?

Electrostatic Potential Example

- Strictly speaking, continuous variables have an uncountable number of possible values, and cannot be evaluated numerically
- Key is to transform from continuous variables to those on a grid
- Increase resolution as needed



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:

$$\begin{pmatrix} -2 & 1 & 0 & & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & & 0 & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_{N-2} \\ \varphi_{N-1} \\ \varphi_N \end{pmatrix} = \frac{h^2}{\epsilon_o} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_{N-2} \\ \rho_{N-1} \\ \rho_N \end{pmatrix}$$

Electrostatic Potential Solution: 1D

- In MATLAB, use:

```
>> N=10; 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     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
    0.9597
    0.3404
    0.5853
    0.2238
    0.7513
    0.2551
    0.5060
    0.6991
```

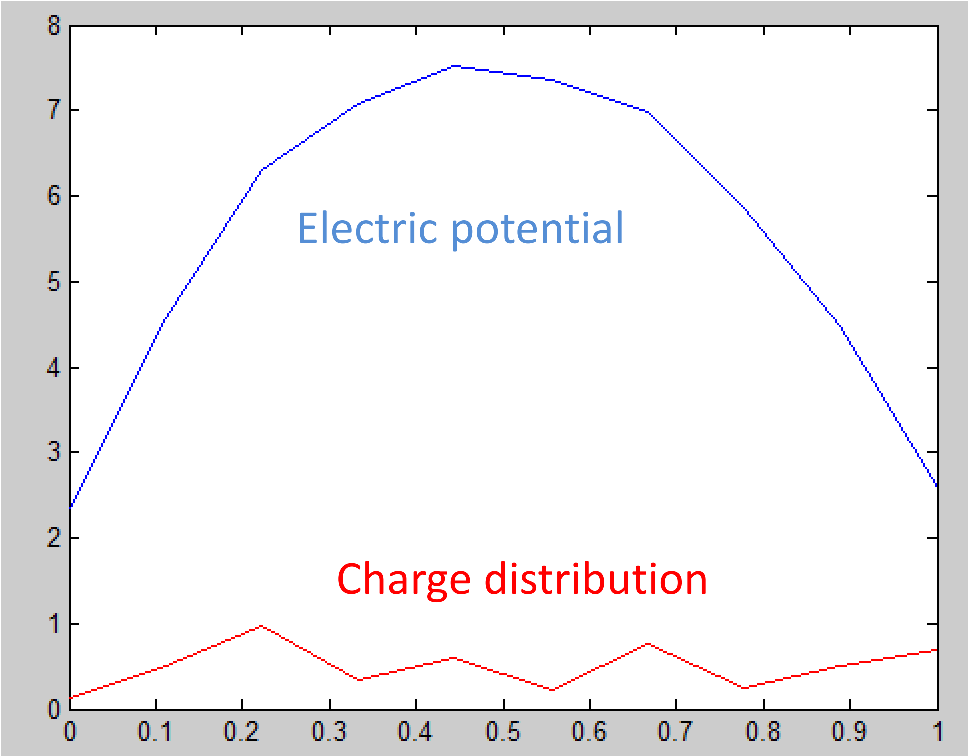
Electrostatic Potential Solution: 1D

- Use backslash operator to solve linear algebra 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
```

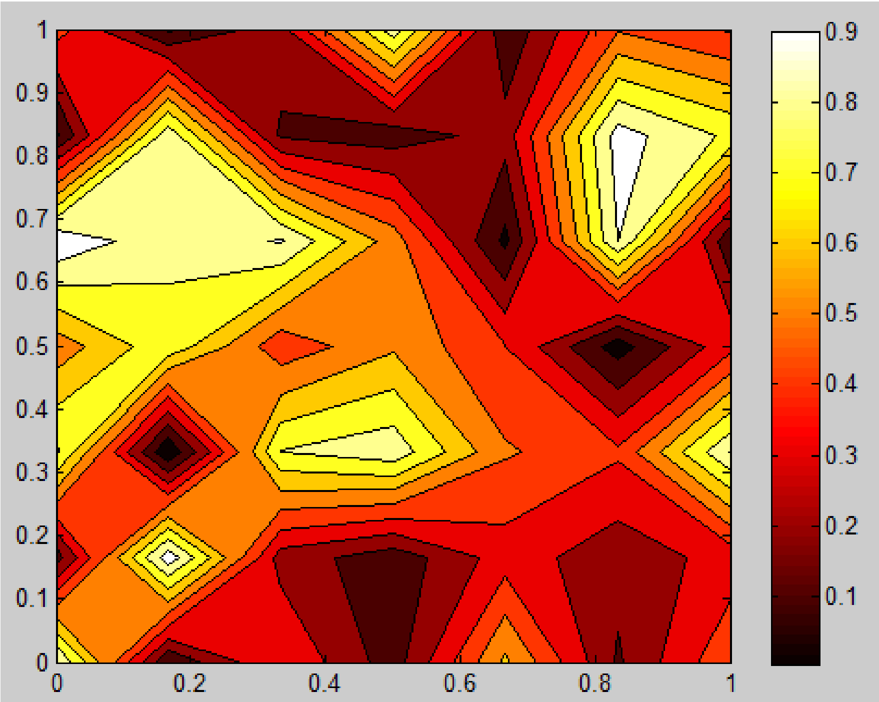


Electrostatic Potential Solution: 2D

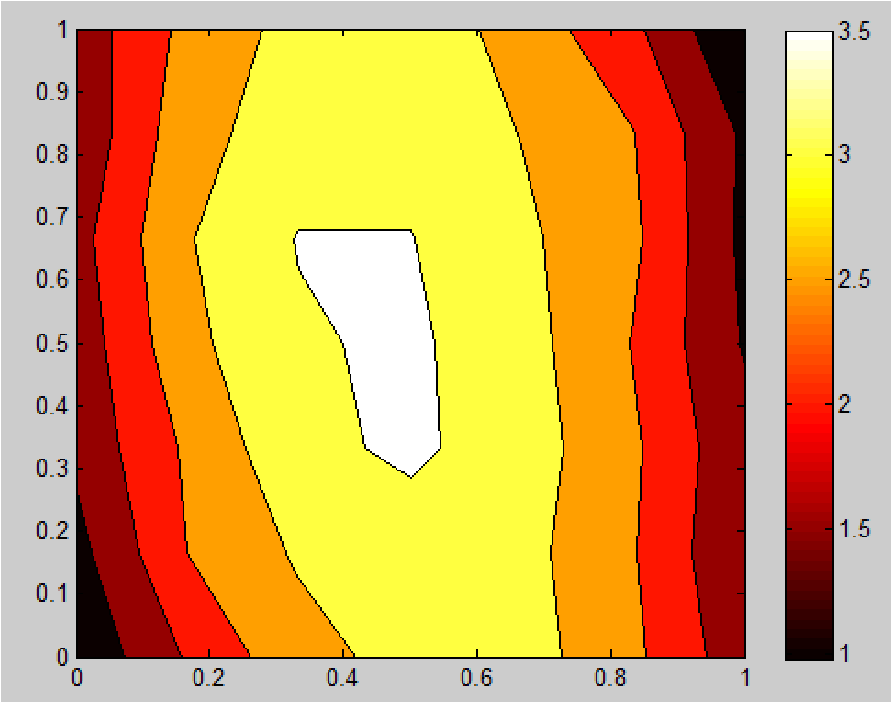
- Connections in two directions creates a total of 5 non-vanishing diagonals in our linear algebra problem:

$$\begin{pmatrix} -4 & 1 & 0 & & 1 & 0 & 0 \\ 1 & -4 & 1 & \cdots & 0 & 1 & 0 \\ 0 & 1 & -4 & & 0 & 0 & 1 \\ & \vdots & & \ddots & & \vdots & \\ 1 & 0 & 0 & & -4 & 1 & 0 \\ 0 & 1 & 0 & \cdots & 1 & -4 & 1 \\ 0 & 0 & 1 & & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_{N-2} \\ \varphi_{N-1} \\ \varphi_N \end{pmatrix} = \frac{h^2}{\epsilon_o} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_{N-2} \\ \rho_{N-1} \\ \rho_N \end{pmatrix}$$

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:

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

- 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:

```
>> 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
     1     2     1     0     0     0
     0     1     2     1     0     0
     0     0     1     2     1     0
     0     0     0     1     2     1
     0     0     0     0     1     2
```

Spin Array Solution: 1D

```
>> [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    3.2470         0         0         0         0
         0         0    2.4450         0         0         0
         0         0         0    1.5550         0         0
         0         0         0         0    0.7530         0
         0         0         0         0         0    0.1981
```

Spin Array Solution: 2D

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

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

- Use Schrodinger equation to set up eigenproblem:

$$\mathcal{H}\Psi = E\Psi$$

- Choose basis spinor wavefunction :

$$\Psi = (\sigma_1, \sigma_2, \cdots, \sigma_N)^T$$

Spin Array Solution: 2D

- In MATLAB, use the following:

```
>> 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

```
>> [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
```

Next Class

- Is on Wednesday, Jan. 30
- Will discuss numerical tools for simulating eigenproblems further