Lecture 20: Numerical Linear Algebra #4
Iterative methods and Eigenproblems

Outline

1) Motivation: beyond LU for \( Ax = b \)
   A little PDE's and sparse matrices
      A) Temperature Equation
      B) Poisson Equation
2) Splitting Methods
   A) General Overview
   B) Jacobi
   C) Gauss-Seidel
3) Iterative methods for Finding Eigenvalues
   A) Power Method
   B) Inverse Power Method
   C) Inverse Power Method with shifts

Motivation: Sparse matrices and numerical PDE's

Basic Problem: 2-D Temperature evolution on a square with Heating

TempBump.avi
Motivation: Sparse matrices and numerical PDE's

Basic Problem: 2-D Temperature evolution on a square with Heating

Steady Solution
Motivation: Sparse matrices and numerical PDE's

Basic Problem: Steady State Discrete Poisson's Equation by Finite Differences
Motivation: Sparse matrices and numerical PDE's

Basic Problem: Steady State Discrete Poisson's Equation by Finite Differences -- stencil Notation

Motivation: Sparse matrices and numerical PDE's

Basic Problem: Steady State Discrete Poisson's Equation by Finite Differences -- matrix -vector form
PDE's as linear algebra

Discrete Laplacian on a square -- spy(A)

```
width = 1; % size of domain
N = 100; % number of full grid points on a side
x = linspace(0,width,N);
[X,Y]=meshgrid(x,x);
dx = x(2)-x(1);

% set up square heat source
xmin=.3; xmax = .5;
 ymin=.5; ymax = .7;
rho=(X>=xmin).*(X<=xmax).*(Y>=ymin).*(Y<=ymax);

% set up steady state problem
A = delsq(numgrid('S',N)); % get laplacian matrix
index = [2:N-1]'; % subarray index
b = dx^2*reshape(rho(index,index),(N-2)^2,1); % rhs vector

x=A; % solution vector
T=zeros(size(X)); % solution array
T(index,index) = reshape(u,N-2,N-2);

% quick and dirty picture
figure;
imagesc(x,x,T); axis image; axis xy;
```
Iterative Methods for Linear systems

Idea: given a guess for $x$, can we feed it back into $Ax=b$ to get a better guess...if $A$ is sparse, then $Ax$ is cheap!

Overall schemes:
  a) Basic Splitting schemes (Jacobi, Gauss-Seidel, SOR)
  b) Advanced iterative methods (Preconditioned Krylov subspace Schemes)
  c) Multi-grid schemes

Simple Splitting Schemes:
  (warning...by themselves these schemes are not practical, but can be very powerful as "pre-conditioners" or as part of Multi-grid schemes)

Idea:
Iterative Methods for Linear systems

Example: Jacobi Iteration

Example: Gauss-Seidel Iteration
Iterative Methods for Linear systems

Convergence behavior of Splitting schemes

Iterative Methods for Finding Eigenvalues

Question: given a square matrix $A$. How do you find its largest eigenvalue (or spectral radius $\rho(A) = |\lambda|_{\text{max}}$)?

1) Get an upper bound using matrix norms

2) try Power method
Iterative Methods for Finding Eigenvalues

1) Naive Power method

Issues:
   1) if x is real, can never find complex eigenvalues
   2) if $\rho(A)>1$ then method rapidly diverges

Iterative Methods for Finding Eigenvalues

2) Normalized Power method

More issues:
   1) Can only find the largest eigenvalue
   2) can converge very slowly
Iterative Methods for Finding Eigenvalues

Convergence rate of normalized power method

Iterative Methods for Finding Eigenvalues

Inverse Power Method
Iterative Methods for Finding Eigenvalues

Inverse Power Method with shifts

Iterative Methods for Finding Eigenvalues

Inverse Power Method with shifts - Rayleigh Quotient acceleration