Lecture 18

Iterative solution of linear systems*

Newton refinement

Conjugate gradient method

Review of Part II

Methods and Formulas

Basic Matrix Theory:

Identity matrix: $AI = A$, $IA = A$, and $I\mathbf{v} = \mathbf{v}$
Inverse matrix: $AA^{-1} = I$ and $A^{-1}A = I$
Norm of a matrix: $\|A\| \equiv \max_{\|\mathbf{v}\|=1} \|A\mathbf{v}\|$
A matrix may be singular or nonsingular. See Lecture 10.

Solving Process:

Learn the exact Gaussian Elimination algorithm:
Row $j \mapsto$ Row j - (ratio) Row i
Gaussian Elimination this way produces LU decomposition
Row Pivoting (bigger absolute number on top)
Back Substitution

Condition number:

$$\text{cond}(A) \equiv \max\left(\frac{\|\delta\mathbf{x}\|/\|\mathbf{x}\|}{\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}} \right) = \max\left(\frac{\text{Relative error of output}}{\text{Relative error of inputs}} \right).$$

A big condition number is bad; in engineering it usually results from poor design.

LU factorization:

The LU factorization is a by-product of Gaussian Elimination (if done with the correct algorithm).

$$PA = LU.$$

Solving steps:
Multiply by P: $\mathbf{d} = P\mathbf{b}$
Forwardsolve: $L\mathbf{y} = \mathbf{d}$
Backsolve: $U\mathbf{x} = \mathbf{y}$

Eigenvalues and eigenvectors:

A nonzero vector $\mathbf{v}$ is an eigenvector and a number λ is its eigenvalue if

$$A\mathbf{v} = \lambda\mathbf{v}.$$

Characteristic equation: $\det(A - \lambda I) = 0$
Equation of the eigenvector: $(A - \lambda I)\mathbf{v} = \mathbf{0}$
Residual for an approximate eigenvector-eigenvalue pair: $r = \|A\mathbf{v} - \lambda\mathbf{v}\|$

Complex eigenvalues:

Occur in conjugate pairs: $\lambda_{1,2} = \alpha \pm i\beta$
and eigenvectors must also come in conjugate pairs: $\mathbf{w} = \mathbf{u} \pm i\mathbf{v}$.

Vibrational modes:

Eigenvalues are frequencies squared. Eigenvectors represent modes.

Power Method:

- Repeatedly multiply $\mathbf{x}$ by A and divide by the element with the largest absolute value.
- The element of largest absolute value converges to largest absolute eigenvalue.
- The vector converges to the corresponding eigenvector.
- Convergence assured for a real symmetric matrix, but not for an arbitrary matrix, which may not have real eigenvalues at all.

Inverse Power Method:

- Apply power method to A^{-1}.
- Use solving rather than the inverse.
- If λ is an eigenvalue of A then $1/\lambda$ is an eigenvalue for A^{-1}.
- The eigenvectors for A and A^{-1} are the same.

Symmetric and Positive definite:

- Symmetric: $A = A'$.
- If A is symmetric its eigenvalues are real.
- Positive definite: $A\mathbf{x} \cdot \mathbf{x} > 0$.
- If A is positive definite, then its eigenvalues are positive.

QR method (Not covered in MATH 3600 at Ohio):

- Transform A into H the Hessian form of A.
- Decompose H in QR.
- Multiply Q and R together in reverse order to form a new H.
- Repeat
- The diagonal of H will converge to the eigenvalues of A.

Matlab

Matrix arithmetic:

```
A = [ 1  3 -2 5 ;  -1  -1 5 4 ; 0 1 -9  0]
```
.... Manually enter a matrix.
```
u = [ 1  2  3  4]'
A*u
B = [3 2 1; 7 6 5; 4 3 2]
B*A
```
.... multiply B times A.
```
2*A
```
.... multiply a matrix by a scalar.
```
A + A
```
.... add matrices.
```
A + 3
```
.... add 3 to every entry of a matrix.
```
B.*B
```
.... component-wise multiplication.
```
B.^3
```
.... component-wise exponentiation.

Special matrices:

```
I = eye(3)
```
.... identity matrix
```
D = ones(5,5)
O = zeros(10,10)
C = rand(5,5)
```
.... random matrix with uniform distribution in $[0, 1]$.
```
C = randn(5,5)
```
.... random matrix with normal distribution.
```
hilb(6)
pascal(5)
```

General matrix commands:

```
size(C)
```
.... gives the dimensions $(m \times n)$ of A.
```
norm(C)
```
.... gives the norm of the matrix.
```
det(C)
```
.... the determinant of the matrix.
```
max(C)
```
.... the maximum of each row.
```
min(C)
```
.... the minimum in each row.
```
sum(C)
```
.... sums each row.
```
mean(C)
```
.... the average of each row.
```
diag(C)
```
.... just the diagonal elements.
```
inv(C)
```
.... inverse of the matrix.
```
C'
```
.... transpose of the matrix.

Matrix decompositions:

```
[L U P] = lu(C)
[Q R] = qr(C)
H = hess(C)
```
`H = hess(C)` .... transform into a Hessian tri-diagonal matrix, which has the same eigenvalues as A.

Part III

Functions and Data