ECE257 Numerical Methods and Scientific Computing

Linear Algebraic Equations
Today’s class:

• Linear Algebraic Equations
• Gaussian Elimination
Linear Algebraic Equations

- Solving for roots gave us solutions to equations of the form:
  \[ f(x) = 0 \]

- A more general problem would be to solve the following n equations simultaneously

\[
\begin{align*}
    f_1(x_1, x_2, \ldots, x_n) &= 0 \\
    f_2(x_1, x_2, \ldots, x_n) &= 0 \\
    &\vdots \\
    f_n(x_1, x_2, \ldots, x_n) &= 0
\end{align*}
\]
Linear algebraic systems

- A linear algebraic system is a system of equations where all the functions are linear

\[ \begin{align*}
    a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n - b_1 &= 0 \\
    a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n - b_2 &= 0 \\
    &\vdots \\
    a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n - b_n &= 0
\end{align*} \]
Linear algebraic systems

• Graphical solutions
  – Plot the functions and the solution is the intersection point of the functions
  – For second order linear systems, each equation is a line
  – For third order linear systems each equation is a plane
Linear Algebraic Systems

- Example:

\[ 3x_1 + 2x_2 = 18 \]
\[ -x_1 + 2x_2 = 2 \]
Linear Algebraic Systems

- Singular system (no solution)
Linear Algebraic Systems

- Singular system (infinite solutions)

From Numerical Methods for Engineers, Chapra and Canale, Copyright © The McGraw-Hill Companies, Inc.
Linear Algebraic Systems

- Ill-conditioned system

From *Numerical Methods for Engineers*, Chapra and Canale, Copyright © The McGraw-Hill Companies, Inc.
Linear algebraic systems

• Graphical methods work only for second and third order systems

• Not precise

• Useful visualization tool
Linear algebraic equations

- In matrix form

\[ AX = B \]

- where A is a \( n \times n \) matrix, and X and B are \( n \times 1 \) vectors.
Matrices

• Definitions:
  – Symmetric matrix
  – Diagonal matrix
  – Identity matrix (I)
Matrices

• Definitions

– Upper triangular

\[ a_{ij} = 0 \quad \text{if} \quad i > j \]

– Lower triangular

\[ a_{ij} = 0 \quad \text{if} \quad i < j \]

– Tridiagonal

\[ a_{ij} = 0 \quad \text{if} \quad |i - j| > 1 \]
Matrices

- Definitions
  - Banded
    \[ a_{ij} = 0 \quad \text{if} \quad |i - j| > k \]
  - Transpose
    \[ a_{ij}^T = a_{ji} \quad \forall i, j \]
Matrix Operations

- **Addition**

\[ C = A + B \implies c_{ij} = a_{ij} + b_{ij} \]

- **Subtraction**

\[ C = A - B \implies c_{ij} = a_{ij} - b_{ij} \]

- **Multiplication**

\[ C = AB \implies c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \]
Matrices

- Data representation
  - 2-D array
  - 1-D array
  - Array of pointers
  - Sparse matrices
Matrix multiplication

\[ C_{n \times l} = A_{n \times m} B_{m \times l} \]

double A[n][m];
double B[m][l];
double C[n][l];

for (int i=0; i<n; i++) {
    for (int j=0; j<l; j++) {
        double sum = 0;
        for (int k=0; k<m; k++) {
            sum = sum + A[i][k]*B[k][j];
        }
        C[i][j] = sum;
    }
}

\( n \times m \times l \) multiplications
Matrix operations

- Addition/Subtraction - $O(n^2)$
- Multiplication - $O(n^3)$
  - Strassen’s algorithm - $O(n^{\log_2 7})$
  - Coppersmith and Winograd $O(2.376)$
Inverse Matrices

- If $A$ is non-singular and square, then $A^{-1}$ is the inverse such that

$$A^{-1}A = I$$

$$A^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$
Linear algebraic equations

• In matrix form

\[ AX = B \]

• where A is a \( n \times n \) matrix, and X and B are \( n \times 1 \) vectors.

\[
\begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1n} \\
  a_{21} & a_{22} & \cdots & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{bmatrix}
=
\begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_n
\end{bmatrix}
\]
Linear algebraic equations

\[ AX = B \]

- We need to solve for \( X \)

\[ A^{-1}AX = A^{-1}B \]

\[ X = A^{-1}B \]
Linear Equations

- How do we get $A^{-1}$?
  - It is non-trivial
  - Not very efficient

- Usually use other methods to solve for $X$
  - Gaussian Elimination
  - LU Decomposition
Linear Algebraic Equations

- Solutions for second and third order systems
  - Graphical methods
  - Elimination of unknowns
  - Determinants, Cramer’s Rule
Determinants, Cramer’s Rule

• Given a second-order matrix $A$, the determinant $D$ is defined as follows:

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

• Given a third-order matrix $A$, the determinant $D$ is defined as follows:

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$
Determinants, Cramer’s Rule

• Using determinants to solve a linear system

• Cramer’s rule
  – Replace a column of coefficients in matrix $A$ with the $B$ vector and find determinant

\[
\begin{align*}
  x_1 &= \frac{b_1 a_{12} a_{13}}{D}, \\
  x_2 &= \frac{a_{11} b_1 a_{13}}{D}, \\
  x_3 &= \frac{a_{11} a_{12} b_1}{D}
\end{align*}
\]
Cramer’s rule example

\[3x_1 + 2x_2 = 18\]
\[-x_1 + 2x_2 = 2\]

\[D = \begin{vmatrix} 3 & 2 \\ -1 & 2 \end{vmatrix} = 3 \cdot 2 - (-1) \cdot 2 = 8\]

\[x_1 = \frac{18 \cdot 2 - 2 \cdot 2}{D} = 4, \quad x_2 = \frac{3 \cdot 2 - (-1) \cdot 18}{D} = 3\]
Gaussian Elimination

• Extension of elimination of unknowns as a systematic algorithm

• Two steps
  – Forward elimination
  – Back substitution
Gaussian Elimination

- **Forward elimination**

  
  \[ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \]

  \[ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \]

  \[ \vdots \]

  \[ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \]

- **Eliminate** \( x_1 \) **from row 2**

  \[ \text{– Multiply row 1 by } \frac{a_{21}}{a_{11}} \]

  
  \[ \frac{a_{21}}{a_{11}}a_{11}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \ldots + \frac{a_{21}}{a_{11}}a_{1n}x_n = \frac{a_{21}}{a_{11}}b_1 \]
Gaussian Elimination

• Eliminate $x_1$ from row 2
  – Subtract row 1 from row 2

\[
(a_{21} - a_{21})x_1 + \left( a_{22} - \frac{a_{21}}{a_{11}} a_{12} \right) x_2 + \ldots + \left( a_{2n} - \frac{a_{21}}{a_{11}} a_{1n} \right) x_n = b_2 - \frac{a_{21}}{a_{11}} b_1
\]

\[
a'_{22} x_2 + a'_{23} x_3 + \ldots + a'_{2n} x_n = b'_2
\]

• Eliminate $x_1$ from all other rows in the same way

• Then eliminate $x_2$ from rows 3-n and so on
Gaussian Elimination

- Forward elimination

\[
\begin{align*}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a'_{22}x_2 + a'_{23}x_3 + \ldots + a'_{2n}x_n &= b'_2 \\
a''_{33}x_3 + \ldots + a''_{2n}x_n &= b''_3 \\
& \vdots \\
a_{nn}^{(n-1)}x_n &= b_{n}^{(n-1)}
\end{align*}
\]

- Back substitute to solve for \( x \)

\[
x_n = \frac{b_{n}^{(n-1)}}{a_{nn}^{(n-1)}}
\]
Gaussian Elimination

• Back substitution

\[ a_{n-1n-1}^{(n-2)} x_{n-1} + a_{n-1n}^{(n-2)} x_n = b_{n-1}^{(n-2)} \]
\[ a_{n-1n-1}^{(n-2)} x_{n-1} + a_{n-1n}^{(n-2)} \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} = b_{n-1}^{(n-2)} \]
\[ b_{n-1}^{(n-2)} - a_{n-1n}^{(n-2)} \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \]
\[ x_{n-1} = \frac{b_{n-1}^{(n-2)} - a_{n-1n}^{(n-2)} b_n^{(n-1)} a_{nn}^{(n-1)}}{a_{n-1n-1}^{(n-2)}} \]

• In general,

\[ b_{i-1}^{(i-2)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j \]
\[ x_i = \frac{b_{i-1}^{(i-2)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}} \]
Gaussian Elimination

\[
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & c_1 \\
a_{21} & a_{22} & a_{23} & c_2 \\
a_{31} & a_{32} & a_{33} & c_3 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & c_1 \\
da_{22} & a_{23} & c_2 \\
a_{33} & c_3 \\
\end{bmatrix}
\]

Forward elimination

\[
x_3 = \frac{c_3''}{a_{33}}
\]

\[
x_2 = \frac{c_2' - a_{23}'x_3}{a_{22}'}
\]

\[
x_1 = \frac{c_1' - a_{12}'x_2 - a_{13}'x_3}{a_{11}'}
\]

Back substitution

From *Numerical Methods for Engineers*, Chapra and Canale, Copyright © The McGraw-Hill Companies, Inc.
Gaussian elimination

(a) \[ \sum_{k=1}^{n-1} (n-k) \text{ divisions} \]

\[ \text{DO } k = 1, n - 1 \]
\[ \text{DO } i = k + 1, n \]
\[ \text{factor} = a_{i,k} / a_{k,k} \]
\[ \text{DO } j = k + 1 \text{ to } n \]
\[ a_{i,j} = a_{i,j} - \text{factor} \cdot a_{k,j} \]
\[ \text{END DO} \]

\[ \text{SUM} = 0 \]
\[ \text{DO } j = i + 1, n \]
\[ \text{SUM} = \text{SUM} + a_{i,j} \cdot x_j \]
\[ \text{END DO} \]

(b) \[ \sum_{i=1}^{n-1} (n-i) \text{ multiplications} \]

\[ x_n = b_n / a_{n,n} \]
\[ \text{DO } i = n - 1, 1, -1 \]
\[ \text{SUM} = 0 \]
\[ \text{DO } j = i + 1, n \]
\[ \text{SUM} = \text{SUM} + a_{i,j} \cdot x_j \]
\[ \text{END DO} \]

\[ x_i = (b_i - \text{SUM}) / a_{i,i} \]
\[ \text{END DO} \]
Gaussian elimination

- Elimination step

\[
\begin{align*}
\sum_{k=1}^{n-1} & \left[ (n-k) + (n-k)(n-k) + (n-k) \right] \\
\sum_{k=1}^{n-1} & (n-k) \left[ 1 + (n-k) + 1 \right] \\
\sum_{k=1}^{n-1} & \left( n(n+2) - (2n+2k + k^2) \right) \\
& n(n+2)(n-1) - 2(n+1) \frac{(n-1)n}{2} + \frac{n(n-1)(2n-1)}{6} \\
& O(n^3)
\end{align*}
\]
Gaussian elimination

- Back substitution step

\[\begin{align*}
1 + \sum_{i=1}^{n-1} (n-i) + (n-1) \\
n + n(n-1) - \sum_{i=1}^{n-1} i \\
n + n(n-1) - \frac{n(n-1)}{2} \\
\frac{n(n+1)}{2} \\
O(n^2)
\end{align*}\]
Gaussian Elimination

- Things to worry about
  - Division by zero
  - Round-off error
  - Ill-conditioned system
Gaussian elimination

- Ill-conditioned system example

\[
\begin{align*}
x_1 + 2x_2 &= 10 \\
1.1x_1 + 2x_2 &= 10.4
\end{align*}
\]

\[
x_1 = \frac{2(10) - 2(10.4)}{1(2) - 2(1.1)} = 4
\]

\[
x_2 = \frac{1(10.4) - 1.1(10)}{1(2) - 2(1.1)} = 3
\]
Gaussian elimination

- Ill-conditioned system example

\[
\begin{align*}
x_1 + 2x_2 &= 10 \\
1.05x_1 + 2x_2 &= 10.4
\end{align*}
\]

\[
x_1 = \frac{2(10) - 2(10.4)}{1(2) - 2(1.05)} = 8
\]

\[
x_2 = \frac{1(10.4) - 1.05(10)}{1(2) - 2(1.05)} = 1
\]
Gaussian elimination

- If the determinant is close to zero, the system is ill-conditioned
- However, the magnitude of the determinant can change by multiplying by a constant
Gaussian elimination with pivoting

• Basic idea is to remove divide by zero if $a_{11}$ is zero

• Swap the row with the largest element with the top row
Gaussian elimination with pivoting

\[
\begin{align*}
0.0003x_1 + 3.0000x_2 &= 2.0001 \\
1.0000x_1 + 1.0000x_2 &= 1.0000 \\
\downarrow \\
1.000x_1 + \frac{3.0000}{0.0003}x_2 &= \frac{2.0001}{0.0003} \\
1.0000x_1 + 1.0000x_2 &= 1.0000 \\
\downarrow \\
9999x_2 &= 6666 \\
\downarrow \\
x_2 &= \frac{2}{3} \\
x_1 &= \frac{2.0001 - 3\frac{2}{3}}{0.0003}
\end{align*}
\]
Gaussian elimination with pivoting

\[
\begin{align*}
1.0000x_1 + 1.0000x_2 &= 1.0000 \\
0.0003x_1 + 3.0000x_2 &= 2.0001 \\
\downarrow \\
0.0003x_1 + 0.0003(1.0000)x_2 &= 0.0003(1.0000) \\
0.0003x_1 + 3.0000x_2 &= 2.0001 \\
\downarrow \\
2.9997x_2 &= 1.9998 \\
\downarrow \\
x_2 &= \frac{2}{3} \\
x_1 &= \frac{1-\frac{2}{3}}{1}
\end{align*}
\]
Gaussian elimination with scaling

- It is sometimes useful to scale the equations so that the largest coefficient in any row is 1

- Example

  \[ 2x_1 + 100,000x_2 = 100,000 \]
  \[ x_1 + x_2 = 2 \]
  \[ 2x_1 + 100,000x_2 = 100,000 \]
  \[ -49999x_2 = -49998 \]
  \[ x_2 = 1.00 \]
  \[ x_1 = 0.00 \]
Gaussian elimination with scaling

• Example

\[
0.00002x_1 + x_2 = 100,000
\]
\[
x_1 + x_2 = 2
\]
\[
x_1 + x_2 = 2
\]
\[
.99998x_2 = 1
\]
\[
x_2 = 1.00
\]
\[
x_1 = 1.00
\]
Next Lecture

• LU Decomposition
• Read Chapter 10
• HW3 due 9/30
• Exam 1 on Tuesday 10/04