ME 3001 Lecture, Systems of Linear Equations
The Gaussian Elimination Algorithm

What is Matrix Multiplication

- Consider 2 conformable matrices F and G with elements f_{ij} and g_{ij}.
- Matrix Multiplication gives the product matrix E with elements e_{ij}.

$$E = F \times G \qquad\qquad e_{ij} = \Sigma_{k=1}^{n} f_{ik} \times g_{kj}$$

- Lets do an example with a few small matrices.

$$\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{bmatrix} \times \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \\ g_{31} & g_{32} \end{bmatrix}$$

- A closer look at E

$$\begin{bmatrix} e_{11} & e_{12} \\ \\ e_{21} & e_{22} \end{bmatrix} = \left[\begin{array}{c|c} f_{11}g_{11} + f_{12}g_{21} + f_{13}g_{31} & f_{11}g_{12} + f_{12}g_{22} + f_{13}g_{32} \\ \\ f_{21}g_{11} + f_{22}g_{21} + f_{23}g_{31} & f_{21}g_{12} + f_{22}g_{22} + f_{23}g_{32} \end{array}\right]$$

- So what is n ?

A Programming Exercise - Matrix Multiplication

- $e_{ij} = \Sigma_{k=1}^{n} f_{ik} \times g_{kj}$
- What does that equation mean?
- How can we write a *General Solution Technique*?

Simple Example (3x3)

Gaussian Elimination

This is a 2 part process

- **Step 1:** Forward Elimination of Unknowns
  - Eliminate x_1 from equations 2 to n
    - Eliminate x_1 from equation 2
      - define the eliminating factor f_{21} as a_{21}/a_{11}
      - redefine a_{21} as $a_{21} - a_{11} * f_{21}$
      - redefine a_{22} as $a_{22} - a_{12} * f_{21}$

        . . .
      - redifine a_{2n} as $a_{2n} - a_{1n} * factor$

    - Eliminate x_1 from equation 3
      - define the eliminating factor f_{31} as a_{31}/a_{11}
      - redefine a_{31} as $a_{31} - a_{11} * f_{31}$
      - redefine a_{32} as $a_{32} - a_{12} * f_{31}$

        . . .
      - redefine a_{3n} as $a_{3n} - a_{1n} * f_{31}$

  - Eliminate x_2 from equations 3 to n

    - Eliminate x_2 from equation 3
      - define the eliminating factor f_{32} as a_{32}/a_{22}
      - redefine a_{32} as $a_{32} - a_{22} * f_{32}$
      - redefine a_{33} as $a_{33} - a_{23} * f_{32}$

        . . .
      - redefine a_{3n} as $a_{3n} - a_{2n} * f_{32}$

    . . .
  - Eliminate x_{n-1} from equation n
    - define the eliminating factor $f_{n,n-1}$ as $a_{n,n-1}/a_{n-1,n-1}$
    - redefine $a_{n,n-1}$ as $a_{n,n-1} - a_{n-1,n-1} * f_{n,n-1}$

- **Step 2:** Backwards Subsitution
  - Solve Equations n through 1
    - Solve for x_n as $\frac{b_n}{a_{n,n}}$
    - Solve for x_{n-1} as $\frac{b_{n-1}-(a_{n-1,n}x_n)}{a_{n-1,n-1}}$
    - Solve for x_{n-2} as $\frac{b_{n-2}-(a_{n-2,n-1}x_{n-1})-(a_{n-2,n}x_n)}{a_{n-2,n-2}}$

      .

      .

      .
    - Solve for x_1 as $\frac{b_1-(a_{12}x_2)-...-(a_{1,n-1}x_{n-1})-(a_{1,n}x_n)}{a_{1,1}}$

- Summary

- **The Forward Eimination Algorithm:**

for k *from* 1 *to* n-1

for i *from* k+1 *to* n

fact= $a_{i,k}/a_{k,k}$

for j *from* k *to* n

$$a_{i,j} = a_{i,j} - \text{fact} \times a_{k,j}$$

end

$b_i = b_i - \text{fact} \times b_k$

end

end

- **The Backwards Substitution Algorithm:**

$x_n = b_n/a_{n,n}$

for i *from* n-1 *to* 1

$$x_i = (b_i - \sum_{j=i+1}^{n} (a_{i,j} x_j))/a_{i,i}$$

end

REMINDER - Homework 2 is due Wed. Sep. 26

REMINDER - MATLAB script from today's lecture will be posted on ilearn.