Lecture 11 - Gauss Elimination

* AMA / Quiz / Prayer

I. Gauss Elimination Algorithm

* Last time we reviewed how to solve a linear system by hand in a systematic way. that way is an algorithm called Gauss Elimination.

* Lets review the algorithm for the general $n \times n$ matrix:

$$\underline{\underline{A}} \cdot \underline{x} = \underline{b}$$

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & \cdots & a_{1,n-1} \\ \vdots & & & \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

A. Forward Elimination

* The first step is forward elimination to make $\underline{\underline{A}}$ into an upper triangular matrix $\underline{\underline{U}}$.

Procedure:

- Start in column 0 (eliminate all elements below diagonal, a_{00})
- Find ratio of $\frac{a_{10}}{a_{11}} \times (-1)$. $\quad a_{10} \rightarrow a_{n-1,0}$
- Multiply all elements in row=1 by ratio (and b_1!) and add to row 1 (and b_1!)
- Repeat for all rows to n-1
- Repeat for all columns ($a_{11}, a_{22}, \cdots, a_{n-2,n-2}$)

* We can write this more precisely in index notation:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right) a_{kj}^{(k)}$$

How many times updated.

$a_{ij}^{(k)}$ — row, col

for: $k = 0, 1, \ldots, n-2$ ← column we are eliminating

$i = k+1, k+2, \ldots, n-1$ ← row we are doing

$j = k, k+1, \ldots, n-1$ ← elements in the row

(only need $k+1, k+2 \ldots, n$ but k gives the '0')

don't forget b:

$$b_i^{(k+1)} = b_i^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right) b_k^{(k)}$$

* This is a triple nested loop!

* the end result is $\underline{\underline{U}} \cdot \underline{x} = \underline{b}'$

$$\begin{bmatrix} a_{00}^{(0)} & a_{01}^{(0)} & \cdots & a_{0,n-1}^{(0)} \\ 0 & a_{11}^{(1)} & \cdots & a_{1,n-1}^{(1)} \\ & \vdots & & \\ 0 & 0 & & a_{n-1,n-1}^{(n-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_0^{(0)} \\ b_1^{(1)} \\ \vdots \\ b_{n-1}^{(n-1)} \end{bmatrix}$$

$\underline{\underline{U}}$ $\qquad$ $\underline{b}'$

Activity

* Forward Elimination in Python.

B. Backward substitution

* the second step is backward substitution, to find all of the x_i's.

* let's review the algorithm for the general nxn matrix:

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ 0 & a_{11} & \cdots & a_{1,n-1} \\ \vdots & & & \\ 0 & 0 & \cdots & a_{n-1,n-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

* I dropped the superscript for clarity & convenience

Procedure:

- start at the bottom (row = n-1). Solve for x_{n-1}.
- Move to row = n-2. Move know x_i's and coefficients (a_{ij}) to the right-hand side (RHS)
- Divide by diagonal (a_{ii}) to solve for x_i
- Repeat, moving up the rows until reach the top.

* We can write this more precisely in index notation:

$$x_{n-1} = \frac{b_{n-1}}{a_{n-1,n-1}}$$

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n-1} a_{ij} x_j \right)$$

for $i = n-2, n-3, \cdots, 0$

Activity

* Backward Substitution in Python

* Put both Forward Elimination and Backward Substitution together to solve a problem.