Operation count for gaussian elimination

Consider an $n \times n$ matrix and an element 'x' in the ith row and column:

$$\begin{array}{cc} & n\times n \quad \text{ith column} \\ \text{ith row} & \begin{bmatrix} \cdot & \cdot & \cdots & \cdots \\ \vdots & \cdot & & \\ \vdots & & x & \\ \vdots & & & \cdot \; \ddots \\ \vdots & & & \cdot \end{bmatrix} \end{array}$$

The total number of operations required for gaussian elimination from element x is:

1. $(n-i)$ multiplications of row i by $(n-i)$ elements under x.
2. $(n-i)$ additions of multiplied row i members to rows underneath it.

total $= (n-i)^2$ operations

$\therefore$ Total number of operations for every element along diagonal is:

$$T = \sum_{k=1}^{n-1} (n-i)^2$$

recall summation identity $\sum_{k=1}^{n} k^2 = \frac{1}{6} n(2n+1)(n+1)$ — same

$\therefore$ The order of the gaussian elimination algo is $O(n^3)$.

Operation count for forward and backward substitutions

Now we see why solving $LUX = B$ is faster than solving $AX = B$ using gaussian elimination.

consider –

$$\therefore \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & \cdots & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{J=i+1}^{n} (a_{iJ} x_J) \right) \quad — ①$$

$\therefore$ # of operations for backward operation per row $= n-i+1$ multiplications + additions for eq ①

# of rows $\rightarrow n$, $\therefore$ we do all these multiplications and additions for each of the n rows. The operation counts for each row keep summing up:

$$\therefore \text{Operations count} = \sum_{i=1}^{n} (n-i+1)$$

same form as $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

$\therefore$ Order: $O(n^2)$

$\therefore$ LU decomposition then solution of $LUX = B$ is faster than gaussian elimination for solving $AX = B$