Revision

System of linear equations

$$Ax = b$$

1. Direct Methods

Gauss Elimination } Scaling
Gauss Jordon } partial pivoting

Comparison of two algorithms

- Computational time

[Computational / Algorithmic Complexity]

- Speed of computer
- Programming Language

} $\rightarrow$ No of basic steps

- Data – value $\rightarrow$ Worst case
  – Size $\rightarrow$ Asymptotic analysis

"Big O" notation

- If computational time is sum of multiple terms, only largest is considered
- If the remaining term is a product keep / drop constant depending on application

Gauss Elimination - $O\left(\frac{2n^3}{3}\right)$

Gauss Jordan - $O(n^3)$

LU Decomposition

$A \rightarrow LU$

Motivation - In many engineering problems (design) one needs to study the performance of the system under different conditions

A — characteristic of the system

b — external forcing

i.e A — constant

b — multiple

GE/GJ : You need to solve the problem independently

LU - It provides an alternative by which you can avoid repeated operations on the same coefficient matrix

$$A = LU$$

$$Ax = b$$

$$LUx = b$$

$$Ly = b$$ - determine y by forward substitution

$$Ux = y$$ - determine x by backward substitution

$$\underset{n\times n}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}} = \underset{\frac{n(n+1)}{2}}{\begin{bmatrix} l_{11} & 0 & 0 & \cdots \\ l_{21} & l_{22} & 0 & 0 \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}} \underset{\frac{n(n+1)}{2}}{\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ 0 & & \ddots & \\ & & & u_{nn} \end{bmatrix}}$$

$$= \begin{bmatrix} l_{11}u_{1} & 0 & 0 & - & - \\ l_{21}u_{11} & l_{21}u_{12} + l_{22}u_{22} & - & - & - \\ & & & & \end{bmatrix}$$

$$l_{11}u_1 = a_{11}$$

$$l_{21}u_{11} = a_{21}$$

n^2 - equations

$n^2 + n$ - unknowns

No unique solution for l_{ij} & u_{ij}

If we can fix 'n' terms, we will get a unique solution

LU Decomposition theorem

If A is a square matrix of size $n \times n$ and if $\det(A) \neq 0$

Then there exists a lower triangular matrix (L) and an upper triangular matrix (U) such that $A = LU$

Further, if the diagonal elements of either L or U are unity,

i.e. l_{ii} or $u_{ii} = 1$ $\quad i = 1, 2, \ldots n$

the both L and U are unique

How to get elements of L & U

1. Gauss Elimination
2. Doolittle method

$\Big\}$ $l_{ii} = 1$

3. Crout method — $u_{ii} = 1$
4. Thomas algorithm — tri-diagonal matrices
5. Cholesky algorithm — Positive definite matrix

Gauss Elimination for LU decomposition

$$GE \begin{cases} A \longrightarrow U \\ \text{multiplication factors } \quad \underline{\underline{l_{ij}}} = \dfrac{a_{ij}}{a_{ii}} \end{cases}$$

Example

$$\begin{bmatrix} 2 & 3 \\ 8 & 5 \end{bmatrix}$$

$$l_{21} = \frac{8}{2} = 4$$

$$\begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 0 & -7 \end{bmatrix}$$

U

Previous Example

$$A = \begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{bmatrix} \qquad b = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix}$$

multiplication factors

$l_{21} = 4/2 = 2$

$l_{31} = -2/2 = -1$

$l_{32} = -3$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & -5 \end{bmatrix}$$

$$Ly = b$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -3 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 5 \\ 3 - 10 = -7 \\ 1 - (-1 \times 5) - (-3 \times -7) = -15 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ -15 \end{bmatrix}$$

$$Ux = y$$

$$y = \begin{bmatrix} 5 \\ -7 \\ -15 \end{bmatrix}$$

$$Ux = y$$

$$\begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -7 \\ -15 \end{bmatrix}$$

$$x_3 = 3$$

$$x_2 = 2$$

$$x_1 = 1$$

'L' is like a "recorder" of the operations to be applied on 'b'

Comparison of GE & LU

	Forward elimination	Backward substitution
GE $\rightarrow$	$O(n^3)$	$O(n^2)$
LU $\rightarrow$	$O(n^3)$	$O(n^2)$

$n^2 + n^2$

r - equations

GE $\quad O(r n^3)$

LU $\quad O(n^3 + r n^2) \sim O(n^3)$

Example – Inverse of a matrix

$$A \underline{x}_1 = \underline{b}_1 \qquad \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

$$A x_2 = b_2 \qquad \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

$$A \underbrace{\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}}_{A^{-1}} = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}$$

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

$\underline{\underline{GE}}$ $\quad O(n4)$

LU $\quad O(n^3 + n^3) \sim O(n^3)$

MATLAB for estimation inverse of a matrix

Example

$$A = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} 0.4 & 0.1 \\ -0.2 & 0.3 \end{bmatrix}$$

$$D = 10$$

$$\overset{L}{\begin{bmatrix} 1 & 0 \\ 2/3 & 1 \end{bmatrix}} \overset{U}{\begin{bmatrix} 3 & 1 \\ 0 & 10/3 \end{bmatrix}}$$

$$Ly = b$$

$$\begin{bmatrix} 1 & 0 \\ 2/3 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow y = \begin{bmatrix} 1 \\ -2/3 \end{bmatrix}$$

$$Ux = y$$

$$\begin{bmatrix} 3 & 1 \\ 0 & 10/3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2/3 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix}$$

2. CROUT METHOD

[Compact methods]

$U_{ii} = 1$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & & 0 \\ l_{21} & l_{22} & & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & 1 & u_{23} & \cdots & u_{2n} \\ 0 & & \ddots & & \\ & & & & 1 \end{bmatrix}$$

$a_{11} = l_{11}$

$a_{21} = l_{21}$ $\Rightarrow$ $l_{i1} = a_{i1}$ $\quad i = 1, 2, \dots n$

$a_{12} = l_{11} u_{12} \Rightarrow u_{12} = \frac{a_{12}}{l_{11}}$

$a_{13} = l_{11} u_{13} \Rightarrow u_{13} = \frac{a_{13}}{l_{11}}$ $\Rightarrow u_{1j} = \frac{a_{1j}}{l_{11}}$ $\quad j = 2 \dots n$

$$\begin{bmatrix} l_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ l_{21} & - & - & \cdot & u_{2n} \\ \vdots & & & & \\ l_{n1} & - & - & & l_{nn} \end{bmatrix}$$

$a_{22} = l_{21} u_{12} + l_{22}$

$\Rightarrow l_{22} = a_{22} - l_{21} u_{12}$

For $j = 2, 3, \dots n-1$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \qquad i = j, j+1, \dots n$$

$$u_{jk} = \frac{a_{jk} - \sum_{i=1}^{j-1} l_{ji} u_{ik}}{l_{jj}} \qquad k = j+1, \dots n$$

end

$$l_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn}$$

Comparison of GE and Crout method for LU decomposition

1. No of operations $O(n^3)$
2. Storage requirement
3. Pivoting is slightly more involved

Doolittle method

$l_{ii} = 1$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \\ & & & u_{nn} \end{bmatrix}$$