- ✓ Any element a_{ij} is actively *modified* for p-steps where, p = min $\{(i\text{-}1), j\}$
- ✓ *Modification* formula: $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}$
- ✓ Summing over p-steps:

$$\sum_{k=1}^{p}\left[a_{ij}^{(k+1)} - a_{ij}^{(k)}\right] = -\sum_{k=1}^{p} l_{ik} a_{kj}^{(k)} \qquad p = \min\{(i-1), j\}$$

- ✓ For $i \leq j$, $p = i - 1$

$$a_{ij}^{(i)} - a_{ij}^{(1)} = -\sum_{k=1}^{i-1} l_{ik} a_{kj}^{(k)}$$

$$a_{ij}^{(1)} = a_{ij} = a_{ij}^{(i)} + \sum_{k=1}^{i-1} l_{ik} a_{kj}^{(k)}$$

Define: $l_{ii} = l_{jj} = 1$

$$\Longrightarrow \qquad a_{ij} = \sum_{k=1}^{i} l_{ik} a_{kj}^{(k)}$$

- ✓ For $i \leq j$, $p = i - 1$

$$a_{ij} = \sum_{k=1}^{i} l_{ik} a_{kj}^{(k)}$$

- ✓ For $j < i$, $p = j$

$$a_{ij}^{(j+1)} - a_{ij}^{(1)} = -\sum_{k=1}^{j} l_{ik} a_{kj}^{(k)}$$

- ✓ But $a_{ij}^{(j+1)} = 0$

$$\Rightarrow \qquad a_{ij}^{(1)} = a_{ij} = \sum_{k=1}^{j} l_{ik} a_{kj}^{(k)}$$

- ✓ Combining two statements:

$$a_{ij} = \sum_{k=1}^{p} l_{ik} a_{kj}^{(k)} \qquad p = \min\{i, j\}$$

$$a_{ij} = \sum_{k=1}^{p} l_{ik} a_{kj}^{(k)} \qquad p = \min\{i, j\}$$

- ✓ Define the elements of matrix ***L*** as: $l_{ik} = l_{ik}$
- ✓ Define the elements of matrix ***U*** as: $u_{kj} = a_{kj}^{(k)}$

$$\Rightarrow \qquad a_{ij} = \sum_{k=1}^{p} l_{ik} u_{kj} \qquad p = \min\{i, j\}$$

- ✓ This is equivalent to matrix multiplication: ***A* = *LU***

Can you see it?

$$a_{ij} = \sum_{k=1}^{p} l_{ik} u_{kj} \qquad p = \min\{i, j\}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$a_{11} = l_{11} u_{11} \qquad a_{12} = l_{11} u_{12} \qquad a_{13} = l_{11} u_{13}$$

$$a_{21} = l_{21} u_{11} \qquad a_{22} = l_{21} u_{12} + l_{22} u_{22}$$
$$a_{23} = l_{21} u_{13} + l_{22} u_{23}$$

$$a_{31} = l_{21} u_{11} \qquad a_{32} = l_{31} u_{12} + l_{32} u_{22}$$
$$a_{33} = l_{31} u_{13} + l_{32} u_{23} + l_{33} u_{33}$$

12 Unknowns and 9 equations! 3 free entries!

In general, n^2 equations and $n^2 + n$ unknows! n free entries!

Doolittle's Algorithm:

Define: $l_{ii} = l_{jj} = 1$

✓ The ***U*** matrix: $i \le j$

$$a_{ij} = a_{ij}^{(i)} + \sum_{k=1}^{i-1} l_{ik} a_{kj}^{(k)} \qquad \Rightarrow \qquad a_{ij} = u_{ij} + \sum_{k=1}^{i-1} l_{ik} u_{kj}$$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \qquad i = 1,2, \ldots n; \qquad j = i, i+1, \ldots n$$

✓ The ***L*** matrix: $j < i$

$$a_{ij} = \sum_{k=1}^{j} l_{ik} a_{kj}^{(k)} \qquad \Rightarrow \qquad a_{ij} = \sum_{k=1}^{j} l_{ik} u_{kj}$$

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}}{u_{jj}} \qquad i = j+1, \ldots n; \quad j = 1,2, \ldots n$$

Crout's Algorithm:

$$a_{ij} = \sum_{k=1}^{p} l_{ik} u_{kj} \qquad p = \min\{i, j\}$$

Define: $u_{ii} = u_{jj} = 1$

✓ The $\boldsymbol{L}$ matrix: $j \le i$

$$a_{ij} = \sum_{k=1}^{j} l_{ik} u_{kj} \qquad \Rightarrow \qquad l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}$$

$$j = 1,2, \dots n; \qquad i = j, j+1, \dots n$$

✓ The $\boldsymbol{U}$ matrix: $i < j$

$$a_{ij} = \sum_{k=1}^{i} l_{ik} u_{kj} \qquad \Rightarrow \qquad u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}}{l_{ii}}$$

$$i = 1,2, \dots n; \qquad j = i+1, \dots n$$

Doolittle's Algorithm (3×3 example):

$$\begin{array}{c} \begin{array}{ccc} 2 & 4 & \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \end{array}$$

- ✓ The $\boldsymbol{L}$ matrix: $j < i$
- ✓ $l_{11} = l_{22} = l_{33} = 1$

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}}{u_{jj}} \qquad i = j + 1, \ldots n; \quad j = 1,2, \ldots n - 1$$

- ✓ $j = 1,\ i = 2, 3$: $l_{21} = \frac{a_{21}}{u_{11}},\ l_{31} = \frac{a_{31}}{u_{11}}$
- ✓ $j = 2,\ i = 3$: $l_{32} = \frac{a_{32} - l_{31} u_{12}}{u_{22}}$

$$\begin{array}{c} 1 \\ 3 \\ 5 \end{array} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

- ✓ The $\boldsymbol{U}$ matrix: $i \le j$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \qquad i = 1,2, \ldots n; \qquad j = i, i + 1, \ldots n$$

- ✓ $i = 1,\ j = 1, 2, 3$: $u_{11} = a_{11},\ u_{12} = a_{12},\ u_{13} = a_{13}$
- ✓ $i = 2,\ j = 2, 3$: $u_{22} = a_{22} - l_{21} u_{12},\ u_{23} = a_{23} - l_{21} u_{13}$
- ✓ $i = 3,\ j = 3$: $u_{33} = a_{23} - l_{31} u_{13} - l_{32} u_{23}$

Crout's Algorithm (3×3 example):

Define: $u_{ii} = u_{jj} = 1$

$$\begin{array}{ccc} 1 & 3 & 5 \\ \multicolumn{3}{c}{\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}} \end{array}$$

✓ The ***L*** matrix: $j \le i$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj} \quad j = 1,2,\ldots n; \qquad i = j, j+1, \ldots n$$

$$\begin{array}{c} 2 \\ 4 \\ \\ \end{array} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

✓ The ***U*** matrix: $i < j$

✓ $u_{11} = u_{22} = u_{33} = 1$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}}{l_{ii}} \quad i = 1,2,\ldots n-1; \qquad j = i+1, \ldots n$$

Verify this computation sequence!

LU Theorem:

Let A_k be a sequence of matrices formed by first k rows and k columns of a $n \times n$ square matrix A. If det $(A_k) \neq 0$ for $k = 1, 2, \ldots (n\text{-}1)$, then there exist an upper triangular matrix U and a lower triangular matrix L such that, $A = LU$. Furthermore, if the diagonal elements of either L or U are unity, i.e. l_{ii} or $u_{ii} = 1$ for $i = 1,2, \ldots n$, both L and U are unique.

A_k

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2k} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} & \cdots & a_{kn} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix}$$

For $k = 1$, the theorem is trivially valid!

A_{k-1} b

$$A_k = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1,k-1} & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2,k-1} & a_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{k-1,1} & a_{k-1,2} & \cdots & a_{k-1,k-1} & a_{k-1,k} \\ \hline a_{k1} & a_{n2} & \cdots & a_{kj} & a_{kk} \end{array}\right]$$

c^T

Let's assume that, the theorem is valid for (k - 1) and prove it for k!

$$\boldsymbol{A_k = L_k U_k}$$

$$\begin{bmatrix} \boldsymbol{A_{k-1}} & \boldsymbol{b} \\ \boldsymbol{c^T} & a_{kk} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L_{k-1}} & \boldsymbol{0} \\ \boldsymbol{l^T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{U_{k-1}} & \boldsymbol{u} \\ \boldsymbol{0} & u_{kk} \end{bmatrix}$$

- ✓ $\boldsymbol{L_{k\text{-}1} U_{k\text{-}1} = A_{k\text{-}1}}$: exists uniquely (by assumption). Also note that the following is valid, $\det(\boldsymbol{A_{k\text{-}1}}) = \det(\boldsymbol{L_{k\text{-}1}}).\det(\boldsymbol{U_{k\text{-}1}}) \neq 0$
- ✓ $\boldsymbol{L_{k\text{-}1} u = b}$: Since $\det(\boldsymbol{L_{k\text{-}1}}) \neq 0$, the triangular system has a unique solution for the vector $\boldsymbol{u}$
- ✓ $\boldsymbol{l^T U_{k\text{-}1} = c^T}$ or $\boldsymbol{U_{k\text{-}1}{}^T l = c}$: Since $\det(\boldsymbol{U_{k\text{-}1}}) \neq 0$, the triangular system has a unique solution for the vector $\boldsymbol{l}$
- ✓ $\boldsymbol{l^T u} + u_{kk} = a_{kk}$: Since $\boldsymbol{l}$ and $\boldsymbol{u}$ are unique, u_{kk} is unique.

Condition for existence: **$\det(A_{k\text{-}1}) = \det(L_{k\text{-}1}).\det(U_{k\text{-}1}) \neq 0$**

Diagonalization (*LDU* theorem):

Let $\boldsymbol{A}$ be a $n \times n$ invertible matrix then there exists a decomposition of the form $\boldsymbol{A} = \boldsymbol{LDU}$ where, $\boldsymbol{L}$ is a $n \times n$ lower triangular matrix with diagonal elements as 1, $\boldsymbol{U}$ is a $n \times n$ upper triangular matrix with diagonal elements as 1, and $\boldsymbol{D}$ is a $n \times n$ diagonal matrix.

(Can you prove it? also done in MTH 102)

Example of a 3×3 matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_{11} = u_{11} & 0 & 0 \\ 0 & d_{22} = u_{22} & 0 \\ 0 & 0 & d_{33} = u_{33} \end{bmatrix} \begin{bmatrix} 1 & {}^{u_{12}}/_{u_{11}} & {}^{u_{13}}/_{u_{11}} \\ 0 & 1 & {}^{u_{23}}/_{u_{22}} \\ 0 & 0 & 1 \end{bmatrix}$$

For symmetric matrix: $\boldsymbol{U} = \boldsymbol{L}^T$ and $\boldsymbol{A} = \boldsymbol{LDL}^T$

Note that the entries of the diagonal matrix $\boldsymbol{D}$ are the *pivots*!

- ✓ For positive definite matrices, *pivots* are positive!
- ✓ Therefore, a diagonal matrix $\boldsymbol{D}$ containing the *pivots* can be factorized as: $\boldsymbol{D} = \boldsymbol{D}^{1/2}\boldsymbol{D}^{1/2}$
- ✓ Example of a 3 × 3 matrix

$$\begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{d_{11}} & 0 & 0 \\ 0 & \sqrt{d_{22}} & 0 \\ 0 & 0 & \sqrt{d_{33}} \end{bmatrix} \begin{bmatrix} \sqrt{d_{11}} & 0 & 0 \\ 0 & \sqrt{d_{22}} & 0 \\ 0 & 0 & \sqrt{d_{33}} \end{bmatrix}$$

- ✓ For *symmetric positive definite* matrices: $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\boldsymbol{T}} = \boldsymbol{L}\,\boldsymbol{D}^{1/2}\boldsymbol{D}^{1/2}\,\boldsymbol{L}^{\boldsymbol{T}}$
- ✓ However, $\boldsymbol{D}^{1/2}\boldsymbol{L}^{\boldsymbol{T}} = (\mathbf{L}\boldsymbol{D}^{1/2})^{\boldsymbol{T}}$. Denote: $\boldsymbol{L}\,\boldsymbol{D}^{1/2} = \boldsymbol{L}_{\mathbf{1}}$
- ✓ Therefore, $\boldsymbol{A} = \boldsymbol{L}_{\mathbf{1}}\boldsymbol{L}_{\mathbf{1}}^{\boldsymbol{T}}$. This is also a *LU-Decomposition* where one needs to evaluate only one triangular matrix $\boldsymbol{L}_{\mathbf{1}}$.

Cholesky Algorithm (for symmetric positive definite matrices):

$$a_{ij} = \sum_{k=1}^{p} l_{ik} u_{kj} \qquad p = \min\{i, j\}$$

$\boldsymbol{A} = \boldsymbol{L}\,\boldsymbol{L}^{\boldsymbol{T}}$ .where $\boldsymbol{U} = \boldsymbol{L}^{\boldsymbol{T}}$. Elements of matrix $\boldsymbol{L}$ are to be evaluated!

- ✓ Diagonal elements of the $\boldsymbol{L}$ matrix: $j = i = p$, $l_{jk} = u_{kj}$

$$a_{jj} = \sum_{k=1}^{j} l_{jk} u_{kj} \qquad \Rightarrow \qquad a_{jj} = l_{jj}^2 + \sum_{k=1}^{j-1} l_{jk}^2$$

$$l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 \right)^{1/2} \qquad j = 1,2, \dots n$$

- ✓ Off-diagonal elements of the $\boldsymbol{L}$ matrix: $j < i$, $p = j$, $l_{jk} = u_{kj}$

$$a_{ij} = l_{ij} l_{jj} + \sum_{k=1}^{j-1} l_{ik} l_{jk} \qquad \Rightarrow \qquad l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}}$$

$$j = 1,2, \dots n; \qquad i = j + 1, \dots n$$

If you already have a *LU-decomposition* available for matrix ***A***:

- ✓ Can you use it to compute the inverse?
- ✓ Can you use it compute the determinant?

Banded Matrix

$$
\begin{array}{cc}
 & \leftarrow \; a \; \rightarrow \\
\begin{array}{c} \uparrow \\ \\ b \\ \downarrow \\ \\ \\ \\ \end{array} &
\begin{bmatrix}
\times & \times & \times & 0 & 0 & 0\ldots & 0 \\
\times & \times & \times & \times & 0 & 0\ldots & 0 \\
\times & \times & \times & \times & \times & 0\ldots & 0 \\
\times & \times & \times & \times & \times & \times\ldots & 0 \\
0 & \times & \times & \times & \times & \times\ldots & 0 \\
0 & 0 & \times & \times & \times & \times\ldots & 0 \\
. & . & . & . & . & . & . \\
 & & & & & &
\end{bmatrix}
\end{array}
$$

Band Width = $a + b$ - 1

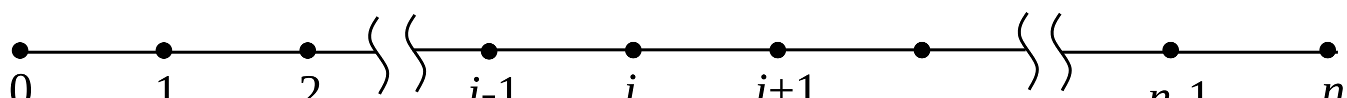
A system of equation with Tri-diagonal coefficient matrix. Total number of elements = n^2. Non-zero elements = 3n-2

$$
\begin{bmatrix}
d_1 & u_1 & 0 & \bullet & 0 & 0 \\
l_2 & d_2 & u_2 & \bullet & 0 & 0 \\
0 & l_3 & d_3 & \bullet & 0 & 0 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
0 & 0 & 0 & l_{n-1} & d_{n-1} & u_{n-1} \\
0 & 0 & 0 & 0 & l_n & d_n
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ \bullet \\ x_{n-1} \\ x_n
\end{bmatrix}
=
\begin{bmatrix}
b_1 \\ b_2 \\ b_3 \\ \bullet \\ b_{n-1} \\ b_n
\end{bmatrix}
$$

Tri-Diagonal Matrix: Origin

$$p(x)y'' + q(x)y' + r(x)y = s(x)$$

$$y(0) = a \text{ and } y(l) = b$$



Node Numbers:-	0	1	2	i-1	i	$i+1$	n-1	n
x-values:-	x_0	x_1	x_2	x_{i-1}	x_i	x_{i+1}	x_{n-1}	x_n
y-values:-	y_0	y_1	y_2	y_{i-1}	y_i	y_{i+1}	y_{n-1}	y_n

Thomas Algorithm (for Tridiagonal)

$$\begin{bmatrix} d_1 & u_1 & 0 & \bullet & 0 & 0 \\ l_2 & d_2 & u_2 & \bullet & 0 & 0 \\ 0 & l_3 & d_3 & \bullet & 0 & 0 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & l_{n-1} & d_{n-1} & u_{n-1} \\ 0 & 0 & 0 & 0 & l_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \bullet \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \bullet \\ b_{n-1} \\ b_n \end{bmatrix}$$

- No need to store $n^2 + n$ elements!
- Store only $4n$ elements in the form of four vectors l, d, u and b
- i^{th} equation is: $l_i x_{i-1} + d_i x_i + u_i x_{i+1} = b_i$
- Notice: $l_1 = u_n = 0$

Thomas Algorithm

- Initialize two new vectors α and β as $\alpha_1 = d_1$ and $\beta_1 = b_1$
- Take the first two equation and eliminate x_1 :

$$\begin{aligned} \alpha_1 x_1 + u_1 x_2 \qquad\qquad &= \beta_1 \\ l_2 x_1 + d_2 x_2 + u_2 x_3 &= b_2 \end{aligned}$$

- Resulting equation is: $\alpha_2 x_2 + u_2 x_3 = \beta_2$

where,

$$\alpha_2 = d_2 - \left(\frac{l_2}{\alpha_1}\right) u_1 \qquad \beta_2 = b_2 - \left(\frac{l_2}{\alpha_1}\right) \beta_1$$

- Similarly, we can eliminate x_2, x_3 ……

Thomas Algorithm

- At the i^{th} step:

$$\alpha_{i-1} x_{i-1} + u_{i-1} x_i = \beta_{i-1}$$

$$l_i x_{i-1} + d_i x_i + u_i x_{i+1} = b_i$$

- Eliminate $x_{i\text{-}1}$ to obtain: $\alpha_i x_i + u_i x_{i+1} = \beta_i$

where,

$$\alpha_i = d_i - \left(\frac{l_i}{\alpha_{i-1}}\right) u_{i-1} \qquad \beta_i = b_i - \left(\frac{l_i}{\alpha_{i-1}}\right) \beta_{i-1}$$

Thomas Algorithm

- Last two equations are:

$$\alpha_{n-1}x_{n-1} + u_{n-1}x_n = \beta_{n-1}$$
$$l_n x_{n-1} + d_n x_n = b_n$$

- Eliminate $x_{n\text{-}1}$ to obtain: $\alpha_n x_n = \beta_n$

$$\alpha_n = d_n - \left(\frac{l_n}{\alpha_{n-1}}\right)u_{n-1} \qquad \beta_n = b_n - \left(\frac{l_n}{\alpha_{n-1}}\right)\beta_{n-1}$$

Thomas Algorithm

- Given: four vectors *l*, *d*, *u* and *b*
- Generate: two vectors α and β as

$$\alpha_1 = d_1 \text{ and } \beta_1 = b_1$$

$$\alpha_i = d_i - \left(\frac{l_i}{\alpha_{i-1}}\right) u_{i-1} \qquad \beta_i = b_i - \left(\frac{l_i}{\alpha_{i-1}}\right) \beta_{i-1}$$

$$i = 2, 3, \ldots.. n$$

- Solution:

$$x_n = \frac{\beta_n}{\alpha_n} \qquad x_i = \frac{\beta_i - u_i x_{i+1}}{\alpha_i}$$

$$i = n\text{-}1 \ldots\ldots 3, 2, 1$$

- FP operations: $8(n\text{-}1) + 3(n\text{-}1) + 1 = 11n - 10$

Thomas Algorithm: Example

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{Bmatrix}$$

$$\alpha_1 = d_1; \alpha_i = d_i - \frac{l_i}{\alpha_{i-1}} u_{i-1}; \beta_1 = b_1; \beta_i = b_i - \frac{l_i}{\alpha_{i-1}} \beta_{i-1} \qquad x_n = \frac{\beta_n}{\alpha_n}; x_i = \frac{\beta_i - u_i x_{i+1}}{\alpha_i}$$

$$\alpha_1 = 2; \alpha_2 = d_2 - \frac{l_2}{\alpha_1} u_1 = 2 - \frac{-1}{2}(-1) = \frac{3}{2}; \alpha_3 = 2 - \frac{-1}{3/2}(-1) = \frac{4}{3}; \alpha_4 = 1 - \frac{-1}{4/3}(-1) = \frac{1}{4}$$

$$\beta_1 = 0; \beta_2 = b_2 - \frac{l_2}{\alpha_1} \beta_1 = 0 - \frac{-1}{2} 0 = 0; \beta_3 = 1 - \frac{-1}{3/2} 0 = 1; \beta_4 = 0 - \frac{-1}{4/3} 1 = \frac{3}{4}$$

$$x_4 = \frac{\beta_4}{\alpha_4} = 3; x_3 = \frac{\beta_3 - u_3 x_4}{\alpha_3} = \frac{1-(-1)3}{4/3} = 3; x_2 = \frac{0-(-1)3}{3/2} = 2; x_1 = \frac{0-(-1)2}{2} = 1$$

ESO 208A: Computational Methods in Engineering

Pivoting, Scaling and Equilibration, Perturbation Analysis

Saumyen Guha

Department of Civil Engineering

IIT Kanpur

Recall Gauss Elimination:

Step 1: $k = 1$

$l_{i1} = \frac{a_{i1}}{a_{11}}$; $a_{ij} = a_{ij}$- $l_{i1}\ a_{1j}$; $b_i = b_i - l_{i1}\ b_1$

$i = 2, 3, \ldots n$ and $j = 2, 3, \ldots\ldots n$

Step 2: $k = 2$

$l_{i2} = \frac{a_{i2}}{a_{22}}$; $a_{ij} = a_{ij}$- $l_{i2}\ a_{2j}$; $b_i = b_i - l_{i2}\ b_2$

$i = 3, 4, \ldots n$ and $j = 3, 4, \ldots\ldots n$

Step k: $k = k$

$l_{ik} = \frac{a_{ik}}{a_{kk}}$; $a_{ij} = a_{ij}$- $l_{ik}\ a_{kj}$; $b_i = b_i - l_{ik}\ b_k$ for

$i = k+1, k+2, \ldots n$ and $j = k+1, k+2, \ldots n$

- ✓ At each k, the all the elements in rows $> k$ are multiplied by l_{ik}.
- ✓ Roundoff errors will be magnified and eventually may grow out of bound if $l_{ik} > 1$ (*i.e.*, algorithm become unstable)
- ✓ Condition for stability: $l_{ik} \leq 1$ (for all k)
- ✓ This will happen only if a_{kk} is the largest element in the k^{th} column for rows $\geq k$
- ✓ If not, make it! (This operation is called *pivoting*)

Partial Pivoting: Matrix at the start of the k^{th} Step:

$$\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1k} & a_{1k+1} & \cdots & a_{1n} \\
0 & a_{22} & a_{23} & a_{24} & \cdots & a_{2k} & a_{2k+1} & \cdots & a_{2n} \\
0 & 0 & a_{33} & a_{34} & \cdots & a_{3k} & a_{3k+1} & \cdots & a_{3n} \\
0 & 0 & 0 & a_{44} & \cdots & a_{4k} & a_{4k+1} & \cdots & a_{4n} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & a_{kk} & a_{kk+1} & \cdots & a_{kn} \\
0 & 0 & 0 & 0 & \cdots & a_{k+1k} & a_{k+1k+1} & \cdots & a_{k+1n} \\
\cdots & \cdots & \cdots & \cdots & \cdots & a_{pk} & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & a_{nk} & a_{nk+1} & \cdots & a_{nn}
\end{bmatrix}$$

Row Exchange

Before operations of the k^{th} step,

- ✓ Search for $\max_{k \le i \le n} |a_{ik}|$. Let's say, it occurs at $i = p$
- ✓ interchange k^{th} row with the p^{th} row
- ✓ Interchange the right hand side vector b_k with b_p, if not working with the augmented matrix.)

Full Pivoting: Matrix at the start of the k^{th} Step:

Column Exchange

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1k} & a_{1k+1} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & a_{24} & \cdots & a_{2k} & a_{2k+1} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & a_{34} & \cdots & a_{3k} & a_{3k+1} & \cdots & a_{3n} \\ 0 & 0 & 0 & a_{44} & \cdots & a_{4k} & a_{4k+1} & \cdots & a_{4n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & a_{kk} & a_{kk+1} & \cdots & a_{kn} \\ 0 & 0 & 0 & 0 & \cdots & a_{k+1k} & a_{k+1k+1} & \cdots & a_{k+1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & a_{pq} & \cdots \\ 0 & 0 & 0 & 0 & \cdots & a_{nk} & a_{nk+1} & \cdots & a_{nn} \end{bmatrix}$$

Row Exchange

Before operations of the k^{th} step,

- ✓ Search for $\max\limits_{\substack{k\le i\le n \\ k\le j\le n}} |a_{ij}|$. Let's say, it occurs at $i = p; j = q$
- ✓ interchange k^{th} row with p^{th} row, k^{th} column with the q^{th} column
- ✓ Interchange the right hand side vector b_k with b_p, if not working with the augmented matrix.)
- ✓ Column interchange → renaming of variables $k \leftrightarrow q$

Perturbation Analysis

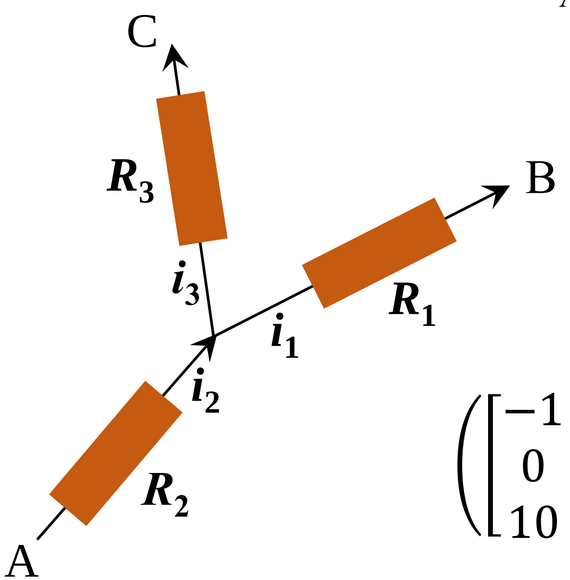
Consider the system of equation $\boldsymbol{Ax} = \boldsymbol{b}$

Question: If small perturbation is given in the matrix $\boldsymbol{A}$ and/or the vector $\boldsymbol{b}$, what is the effect on the solution vector $\boldsymbol{x}$? Alternatively, how sensitive is the solution to small perturbations in the coefficient matrix and the forcing function.

Example:

$R_1 = 10.0 \pm 1.2\ \Omega$; $R_2 = 15.0 \pm 1.8\ \Omega$; $R_3 = 25.0 \pm 2.7\ \Omega$

$V_A - V_C = 100.0 \pm 11.0$ V; $V_A - V_B = 60.0 \pm 9.0$ V



Let's denote the resulting perturbation in the solution vector $\boldsymbol{x}$ as $\boldsymbol{\delta x}$

$$\left(\underbrace{\begin{bmatrix} -1 & 1 & -1 \\ 0 & 15 & 25 \\ 10 & 15 & 0 \end{bmatrix}}_{\boldsymbol{A}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \pm 1.8 & \pm 2.7 \\ \pm 1.2 & \pm 1.8 & 0 \end{bmatrix}}_{\boldsymbol{\delta A}}\right)\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 100 \\ 60 \end{bmatrix}}_{\boldsymbol{b}} + \underbrace{\begin{bmatrix} 0 \\ \pm 11 \\ \pm 9 \end{bmatrix}}_{\boldsymbol{\delta b}}$$

Vector Norms:

A vector norm is a measure (in some sense) of the size of a vector

- ✓ Properties of Vector Norm:
  - ✓ $\|\boldsymbol{x}\| > 0$ for $\boldsymbol{x} \neq \mathbf{0}$; $\|\boldsymbol{x}\| = 0$ iff $\boldsymbol{x} = \mathbf{0}$
  - ✓ $\|\alpha\boldsymbol{x}\| = |\alpha|\|\boldsymbol{x}\|$ for a scalar α
  - ✓ $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$

- ✓ L_p-Norm of a vector $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|_p = \left(|x_1|^p + |x_2|^p \ldots + |x_n|^p\right)^{1/p}$$

- ✓ Example Norms:
  - ✓ $p = 1$: sum of the absolute values
  - ✓ $p = 2$: Euclidean norm
  - ✓ $p \to \infty$: maximum absolute value, $\|\boldsymbol{x}\|_\infty = \max\limits_{0 \leq i \leq n} |x_i|$

Matrix Norms:

A matrix norm is a measure of the size of a matrix

- ✓ Properties of Matrix norm:
  - ✓ $\|\boldsymbol{A}\| > 0$ for $\boldsymbol{A} \neq \boldsymbol{0}$; $\|\boldsymbol{A}\| = 0$ iff $\boldsymbol{A} = \boldsymbol{0}$
  - ✓ $\|\alpha\boldsymbol{A}\| = |\alpha|\|\boldsymbol{A}\|$ for a scalar α
  - ✓ $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$
  - ✓ $\|\boldsymbol{AB}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\|$
  - ✓ $\|\boldsymbol{Ax}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{x}\|$ for consistent matrix and vector norms

- ✓ L_p Norm of a matrix $\boldsymbol{A}$:

$$\|\boldsymbol{A}\|_p = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{Ax}\|_p}{\|\boldsymbol{x}\|_p}$$

- ✓ Spectral Radius: largest absolute eigenvalue of matrix $\boldsymbol{A}$ denoted by $\rho(\boldsymbol{A})$.
  - ✓ If there are m distinct eigenvalues of $\boldsymbol{A}$: $\rho(\boldsymbol{A}) = \max_{1\leq i\leq m} |\lambda_i|$
  - ✓ Lower bound of all matrix norms: $\rho(\boldsymbol{A}) \leq \|\boldsymbol{A}\|$
  - ✓ For any norm of matrix $\boldsymbol{A}$: $\rho(\boldsymbol{A}) = \lim_{n\to\infty} \|\boldsymbol{A}^n\|^{1/n}$

Matrix Norms:

- ✓ *Column-Sum* norm: $\|\boldsymbol{A}\|_1 = \max\limits_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|$

- ✓ *Spectral* norm: $\|\boldsymbol{A}\|_2 = \left(\max\limits_{1\le j\le n} |\lambda_j|\right)^{1/2}$ where, λ_j are the eigenvalues of the square symmetric matrix $\boldsymbol{A}^T\boldsymbol{A}$.

- ✓ *Row-Sum* norm: $\|\boldsymbol{A}\|_\infty = \max\limits_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$

- ✓ *Frobenius* norm: $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2} = \sqrt{\text{trace}(\boldsymbol{A}^T\boldsymbol{A})}$

Perturbation in matrix $\boldsymbol{A}$:

- ✓ System of equation: $(\boldsymbol{A} + \boldsymbol{\delta A})(\boldsymbol{x} + \boldsymbol{\delta x}) = \boldsymbol{b}$
- ✓ $\boldsymbol{A\delta x} + \boldsymbol{\delta A}(\boldsymbol{x} + \boldsymbol{\delta x}) = 0$ since, $\boldsymbol{Ax} = \boldsymbol{b}$
- ✓ $\boldsymbol{\delta x} = -\boldsymbol{A}^{-1}\boldsymbol{\delta A}(\boldsymbol{x} + \boldsymbol{\delta x})$
- ✓ Take the norms of vectors and matrices:

$$\|\boldsymbol{\delta x}\| = \left\|\boldsymbol{A}^{-1}\boldsymbol{\delta A}(\boldsymbol{x}+\boldsymbol{\delta x})\right\| \le \left\|\boldsymbol{A}^{-1}\right\|\|\boldsymbol{\delta A}\|\|\boldsymbol{x}+\boldsymbol{\delta x}\|$$

$$\le \left\|\boldsymbol{A}^{-1}\right\|\|\boldsymbol{\delta A}\|\|\boldsymbol{x}\| + \boxed{\left\|\boldsymbol{A}^{-1}\right\|\|\boldsymbol{\delta A}\|\|\boldsymbol{\delta x}\|}$$

Product of perturbation quantities

$$\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|} \le \|\boldsymbol{A}\|\left\|\boldsymbol{A}^{-1}\right\|\frac{\|\boldsymbol{\delta A}\|}{\|\boldsymbol{A}\|}$$

Perturbation in forcing vector *b*:

- ✓ System of equation: $\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\delta x}) = (\boldsymbol{b} + \boldsymbol{\delta b})$
- ✓ $\boldsymbol{A\delta x} = \boldsymbol{\delta b}$ since, $\boldsymbol{Ax} = \boldsymbol{b}$
- ✓ $\boldsymbol{\delta x} = - \boldsymbol{A}^{-1}\boldsymbol{\delta b}$
- ✓ Take the norms of vectors and matrices:

$$\|\boldsymbol{\delta x}\| = \left\|\boldsymbol{A}^{-1}\boldsymbol{\delta b}\right\| \le \left\|\boldsymbol{A}^{-1}\right\| \|\boldsymbol{\delta b}\| = \left\|\boldsymbol{A}^{-1}\right\| \|\boldsymbol{b}\| \frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|}$$

$$\le \left\|\boldsymbol{A}^{-1}\right\| \|\boldsymbol{A}\| \|\boldsymbol{x}\| \frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|}$$

$$\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|} \le \|\boldsymbol{A}\| \left\|\boldsymbol{A}^{-1}\right\| \frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|}$$

Condition Number:

- ✓ Condition number of a matrix $\boldsymbol{A}$ is defined as:

$$\mathcal{C}(\boldsymbol{A}) = \left\|\boldsymbol{A}^{-\mathbf{1}}\right\| \|\boldsymbol{A}\|$$

- ✓ $\mathcal{C}(\boldsymbol{A})$ is the proportionality constant relating relative error or perturbation in $\boldsymbol{A}$ and $\boldsymbol{b}$ with the relative error or perturbation in $\boldsymbol{x}$
- ✓ Value of $\mathcal{C}(\boldsymbol{A})$ depends on the norm used for calculation. Use the same norm for both A and A-1.
- ✓ If $\mathcal{C}(\boldsymbol{A}) \leq 1$ or of the order of 1, the matrix is *well-conditioned*.
- ✓ If $\mathcal{C}(\boldsymbol{A}) \gg 1$, the matrix is *ill-conditioned*.

Scaling and Equilibration:

- ✓ It helps to reduce the truncation errors during computation.
- ✓ Helps to obtain a more accurate solution for moderately ill-conditioned matrix.
- ✓ Example: Consider the following set of equation

$$\begin{bmatrix} 0.003 & 1.45 & 0.3 \\ 0.00002 & 0.0096 & 0.0021 \\ 0.0015 & 0.966 & 0.201 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 0.12 \\ 19 \end{bmatrix}$$

- ✓ Scale variable $x_1 = 10^3 \times x_1'$ and multiply the second equation by 100. Resulting equation is:

$$\begin{bmatrix} 3 & 1.45 & 0.3 \\ 2 & 0.96 & 0.21 \\ 1.5 & 0.966 & 0.201 \end{bmatrix} \begin{bmatrix} x_1' \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \\ 19 \end{bmatrix}$$

Scaling

- ✓ Vector $\boldsymbol{x}$ is replaced by $\boldsymbol{x'}$ such that, $\boldsymbol{x} = \boldsymbol{Sx'}$.
- ✓ $\boldsymbol{S}$ is a diagonal matrix containing the scale factors!
- ✓ For the example problem:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10^3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1' \\ x_2 \\ x_3 \end{bmatrix}$$

- ✓ $\boldsymbol{Ax} = \boldsymbol{b}$ becomes: $\boldsymbol{Ax} = \boldsymbol{ASx'} = \boldsymbol{A'x'} = \boldsymbol{b}$ where, $\boldsymbol{A'} = \boldsymbol{AS}$
- ✓ For the example problem:

$$\begin{bmatrix} 0.003 & 1.45 & 0.3 \\ 0.00002 & 0.0096 & 0.0021 \\ 0.0015 & 0.966 & 0.201 \end{bmatrix} \begin{bmatrix} 10^3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1.45 & 0.3 \\ 0.02 & 0.0096 & 0.0021 \\ 1.5 & 0.966 & 0.201 \end{bmatrix}$$

- ✓ Scaling operation is equivalent to post-multiplication of the matrix $\boldsymbol{A}$ by a diagonal matrix $\boldsymbol{S}$ containing the scale factors on the diagonal

Equilibration

- ✓ Equilibration is multiplication of one equation by a constant such that the values the coefficients become same order of magnitude as the other equations.
- ✓ The operation is equivalent to pre-multiplication by a diagonal matrix ***E*** on both sides of the equation.
- ✓ $\boldsymbol{Ax} = \boldsymbol{b}$ becomes: $\boldsymbol{EAx} = \boldsymbol{Eb}$
- ✓ For the example problem:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.003 & 1.45 & 0.3 \\ 0.00002 & 0.0096 & 0.0021 \\ 0.0015 & 0.966 & 0.201 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 11 \\ 0.12 \\ 19 \end{bmatrix}$$

$$\begin{bmatrix} 0.003 & 1.45 & 0.3 \\ 0.002 & 0.96 & 0.21 \\ 0.0015 & 0.966 & 0.201 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \\ 19 \end{bmatrix}$$

- ✓ Equilibration operation is equivalent to pre-multiplication of the matrix ***A*** and vector ***b*** by a diagonal matrix ***E*** containing the equilibration factors on the diagonal

Pivoting, Scaling and Equilibration

- ✓ Before starting the solution algorithm, take a look at the entries in $\boldsymbol{A}$ and decide on the scaling and equilibration factors. Construct matrices $\boldsymbol{E}$ and $\boldsymbol{S}$.
- ✓ Transform the set of equation $\boldsymbol{Ax} = \boldsymbol{b}$ to $\boldsymbol{EASx'} = \boldsymbol{Eb}$
- ✓ Solve the system of equation $\boldsymbol{A'x'} = \boldsymbol{b'}$ for $\boldsymbol{x'}$, where $\boldsymbol{A'} = \boldsymbol{EAS}$ and $\boldsymbol{b'} = \boldsymbol{Eb}$
- ✓ Compute: $\boldsymbol{x} = \boldsymbol{Sx'}$
- ✓ Gauss Elimination: perform *partial pivoting* at each step k
- ✓ For all other methods: perform *full pivoting* before the start of the algorithm to make the matrix diagonally dominant, as far as practicable!
- ✓ These steps will guarantee the best possible solution for all well-conditioned and mildly ill-conditioned matrices!
- ✓ However, none of these steps can transform an ill-conditioned matrix to a well-conditioned one.

Iterative Improvement by Direct Methods

- ✓ For moderately ill-conditioned matrices an approximate solution $\tilde{\boldsymbol{x}}$ to the set of equation $\boldsymbol{Ax} = \boldsymbol{b}$ can be improved through iterations using direct methods.
- ✓ Compute: $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\,\tilde{\boldsymbol{x}}$
- ✓ Recognize: $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\,\tilde{\boldsymbol{x}} + \boldsymbol{Ax} - \boldsymbol{b}$
- ✓ Therefore: $\boldsymbol{A}(\boldsymbol{x} - \tilde{\boldsymbol{x}}) = \boldsymbol{A}\Delta\boldsymbol{x} = \boldsymbol{r}$
- ✓ Compute: $\boldsymbol{x} = \tilde{\boldsymbol{x}} + \Delta\boldsymbol{x}$
- ✓ The iteration sequence can be repeated until $\| \Delta\boldsymbol{x} \| \leq \varepsilon$