ESO 208A: Computational Methods in Engineering

System of Linear Equations

Saumyen Guha

Department of Civil Engineering
IIT Kanpur

System of Linear Equations:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$\boldsymbol{A}$ is a square matrix of size $n \times n$

$\boldsymbol{x}$ and $\boldsymbol{b}$ are vectors of size n

Background assumed (MTH 102): various types of matrices (orthogonal, orthonormal, positive-definite, etc.); matrix operations (transpose, adjoint, multiplication, determinant); vector space; column space; null space; rank; conditions for existence of solution; uniqueness of solution; eigenvalues, eigenvectors; diagonalization; vector and matrix norms; etc.

Consider the matrix:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix}$$

- ✓ Elements of matrix $\boldsymbol{A}$ are a_{ij}; $i, j = 1, 2, \ldots n$.
- ✓ Total n^2 elements.
- ✓ If most of the elements are non-zero, the matrix is *dense* or *full*. Otherwise, it is *sparse*.
- ✓ Sparse matrices may have *banded* structure.

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

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

Methods for Solution of the System of Equations:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

- ✓ Direct Methods: one obtains the exact solution (ignoring the round-off errors) in a finite number of steps. These group of methods are more efficient for dense and banded matrices.
  - ✓ Gauss Elimination; Gauss-Jordon Elimination
  - ✓ LU-Decomposition
  - ✓ Thomas Algorithm (for tri-diagonal banded matrix)
- ✓ Iterative Methods: solution is obtained through successive approximation. Number of computations is a function of desired accuracy/precision of the solution and are not known apriori. More efficient for sparse matrices.
  - ✓ Jacobi Iterations
  - ✓ Gauss Seidal Iterations with Successive Over/Under Relaxation

Consider the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is a lower-triangular matrix:

$$\begin{bmatrix} a_{11} & 0 & \dots & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

The solution is:

$$x_1 = \frac{b_1}{a_{11}}; x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}; x_3 = \frac{b_3 - (a_{31}x_1 + a_{32}x_2)}{a_{33}}$$

The *Forward Substitution Algorithm*:

$$x_1 = \frac{b_1}{a_{11}}; x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}}; \quad i = 2, 3, \dots n$$

Consider the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is an upper-triangular matrix:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

The solution is:

$$x_n = \frac{b_n}{a_{nn}}; \qquad x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}};$$

$$x_{n-2} = \frac{b_{n-2} - \left(a_{n-2,n-1}x_{n-1} + a_{n-2,n}x_n\right)}{a_{n-2,n-2}}$$

The *Back Substitution Algorithm*:

$$x_n = \frac{b_n}{a_{nn}}; x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}; \quad i = (n-1), (n-2), \dots 3, 2, 1$$

Gauss Elimination for the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Approach: Operating on rows of matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$, transform the matrix $\boldsymbol{A}$ to an upper triangular matrix. Solve the system using *Back substitution algorithm*.

Indices:

- i: Row index
- j: Column index
- k: Step index

Gauss Elimination for the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Step 1: $k = 1$

Define multiplication factors: $l_{i1} = \frac{a_{i1}}{a_{11}}$

Compute: $a_{ij} = a_{ij} - l_{i1}\, a_{1j}$; $\quad b_i = b_i - l_{i1}\; b_1$ for

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

Gauss Elimination for the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Step 2: $k = 2$

Define multiplication factors: $l_{i2} = \frac{a_{i2}}{a_{22}}$

Compute: $a_{ij} = a_{ij} - l_{i2}\, a_{2j}$; $b_i = b_i - l_{i2}\; b_2$ for $i = 3, 4, \ldots .n$ and $j = 3, 4, \ldots\ldots n$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Step 1: $k = 1$

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

$i = 2, 3, \dots .n$ and $j = 2, 3, \dots\dots n$

Step 2: $k = 2$

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

$i = 3, 4, \dots .n$ and $j = 3, 4, \dots\dots n$

Step k: $k = k$

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

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

Matrix after the k^{th} Step:

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

We only need to perform steps up to $k = n - 1$ in order to make the matrix upper triangular

Gauss Elimination Algorithm

Forward Elimination:

For k = 1, 2, …. (n - 1)

Define multiplication factors: $l_{ik} = \frac{a_{ik}}{a_{kk}}$

Compute: $a_{ij} = a_{ij}$- $l_{ik}\ a_{kj}$; $b_i = \ b_i - l_{ik}\ b_k$ for

i = k+1, k+2, ….n and j = k+1, k+2, ….n

Resulting System of equation is upper triangular. Solve it using the *Back-Substitution algorithm:*

$$x_n = \frac{b_n}{a_{nn}}; x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}; \ \ i = (n-1), (n-2), \ldots 3, 2, 1$$

Gauss Elimination with Augmented Matrix for the matrix equation ***Ax*** **=** ***b***:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} & a_{1,n+1} = b_1 \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} & a_{2,n+1} = b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} & a_{i,n+1} = b_i \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} & a_{n,n+1} = b_n \end{bmatrix}$$

Gauss Elimination Algorithm with Augmented Matrix

Forward Elimination:

For k = 1, 2, …. (n - 1)

Define multiplication factors: $l_{ik} = \frac{a_{ik}}{a_{kk}}$

Compute: $a_{ij} = a_{ij}$- $l_{ik}\ a_{kj}$

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

Resulting System of equation is upper triangular. Solve it using the *Back-Substitution algorithm:*

$$x_n = \frac{a_{n,n+1}}{a_{nn}} \qquad\qquad x_i = \frac{a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}$$

$$i = (n-1), (n-2), \ldots 3, 2, 1$$

Gauss Elimination with augmented matrix for multiple right hand side vectors:

$$\boldsymbol{Ax} = \boldsymbol{b}_1,\ \boldsymbol{Ax} = \boldsymbol{b}_2,\ \ldots\ \boldsymbol{Ax} = \boldsymbol{b}_m$$

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1j} & \ldots & a_{1n} & a_{1,n+1} = b_{11} & \ldots & a_{1,n+m} = b_{m1} \\ a_{21} & a_{22} & \ldots & a_{2j} & \ldots & a_{2n} & a_{2,n+1} = b_{12} & \ldots & a_{2,n+m} = b_{m2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \ldots & a_{ij} & \ldots & a_{in} & a_{i,n+1} = b_{1i} & \ldots & a_{i,n+m} = b_{mi} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nj} & \ldots & a_{nn} & a_{n,n+1} = b_{1n} & \ldots & a_{n,n+m} = b_{mn} \end{bmatrix}$$

Homework: Modify the algorithm for this case

Gauss Elimination: Counting Floating Point Operation

Forward Elimination:

For k = 1, 2, …. (n - 1) [At each k]

Multiplication factors: $l_{ik} = \frac{a_{ik}}{a_{kk}}$ [(n-k) ops]

Compute: $a_{ij} = a_{ij}$- $l_{ik}\ a_{kj}$; $b_i = \ b_i - l_{ik}\ b_k$ [2(n-k)(n-k+1) ops]

i = k+1, k+2, ….n and j = k+1, k+2, ….n

$$Ops = \sum_{k=1}^{n-1} [(n-k) + 2(n-k)(n-k+1)] = \frac{2n^3}{3} + \frac{n^2}{2} - \frac{7n}{6}$$

Back-Substitution algorithm:

$$x_n = \frac{b_n}{a_{nn}}; x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}; \ \ i = (n-1), (n-2), \dots 3, 2, 1$$

$$Ops = 1 + \sum_{i=n-1}^{1} 2(n-i) = n^2 - n + 1$$

$$Total\ Ops = \frac{2n^3}{3} + \frac{n^2}{2} - \frac{7n}{6} + n^2 - n + 1 = \frac{2n^3}{3} + \frac{3n^2}{2} - \frac{13n}{6} + 1$$

- ✓ For large n: Number of Floating Point Operations required to solve a system of equation using Gauss elimination is $\sim 2n^3/3$ (*of the order of*)

- ✓ When is the *Gauss Elimination algorithm* going to fail ?

For $k = 1, 2, \ldots. (n - 1)$

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

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

- ✓ If a_{kk} is zero at any step! The a_{kk}'s are called "*Pivots*" or "*Pivotal Element*"
- ✓ If this happens at some step, does that mean the system cannot be solved by Gauss Elimination?
- ✓ We will come back to this question after we have covered all the direct methods!

Gauss-Jordon Elimination for the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Approach: Operating on rows of matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$, transform the matrix $\boldsymbol{A}$ to an identity matrix. The vector $\boldsymbol{b}$ transforms into the solution vector.

Indices:

- i: Row index
- j: Column index
- k: Step index

Gauss-Jordan Elimination for the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Step 1: $k = 1$

$$a_{1j} = \frac{a_{1j}}{a_{11}}; b_1 = \frac{b_1}{a_{11}}; \quad j = 1, 2, \ldots n$$

$a_{ij} = a_{ij} - a_{i1}\, a_{1j}; \quad b_i = b_i - a_{i1} b_1$ for

$i = 2, 3, \ldots .n$ $(\neq 1)$ and $j = 1, 2, 3, \ldots \ldots n$

Gauss-Jordan Elimination for the matrix equation $\boldsymbol{Ax} = \boldsymbol{b}$:

$$\begin{bmatrix} 1 & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Step 2: $k = 2$

$$a_{2j} = \frac{a_{2j}}{a_{22}}; b_2 = \frac{b_2}{a_{22}}; \; j = 2, \ldots n$$

$a_{ij} = a_{ij} - a_{i2}\, a_{2j}$; $\quad b_i = \; b_i - a_{i2} \; b_2$ for

$i = 1, 3, \ldots .n \; (\neq 2)$ and $j = 2, 3, \ldots\ldots n$

$$\begin{bmatrix} 1 & 0 & \dots & a_{1j} & \dots & a_{1n} \\ 0 & 1 & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Step 1: $k = 1$

$$a_{1j} = \frac{a_{1j}}{a_{11}}; \qquad b_1 = \frac{b_1}{a_{11}}; \qquad j = 1, 2, \dots n$$

$a_{ij} = a_{ij} - a_{i1}\, a_{1j}$; $\quad b_i = \; b_i - a_{i1} b_1$ for $i = 2, 3, \dots .n \; (\neq 1)$ and $j = 1, 2, 3, \dots\dots n$

Step 2: $k = 2$

$$a_{2j} = \frac{a_{2j}}{a_{22}}; \qquad b_2 = \frac{b_2}{a_{22}}; \qquad j = 2, 3, \dots n$$

$a_{ij} = a_{ij} - a_{i2}\, a_{2j}$; $\quad b_i = \; b_i - a_{i2}\; b_2$ for $i = 1, 3, \dots .n \; (\neq 2)$ and $j = 2, 3, \dots\dots n$

Step k: $k = k$

$$a_{kj} = \frac{a_{kj}}{a_{kk}}; \qquad b_k = \frac{b_k}{a_{kk}}; \qquad j = k, \dots n$$

$a_{ij} = a_{ij} - a_{ik}\, a_{kj}$; $\quad b_i = \; b_i - a_{ik} b_k$ for $i = 1, 2, 3, \dots .n \; (\neq k)$ and $j = k, \dots\dots n$

$$\begin{bmatrix} 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Gauss-Jordon Algorithm:

For k = 1, 2, …n

$$a_{kj} = \frac{a_{kj}}{a_{kk}}; \qquad b_k = \frac{b_k}{a_{kk}}; \quad j = k, \dots n$$

$a_{ij} = a_{ij} - a_{ik}\, a_{kj}$; $\quad b_i = \ b_i - a_{ik} b_k$ for i = 1, 2, 3, ….n ($\neq k$) and $j = k, \dots n$

Final $\boldsymbol{b}$ vector is the solution.

If we work with the augmented matrix:

For k = 1, 2, …n

$$a_{kj} = \frac{a_{kj}}{a_{kk}} \qquad j = k, \dots n + 1$$

$a_{ij} = a_{ij} - a_{ik}\, a_{kj}$ $\quad i$ = 1, 2, 3, ….n ($\neq k$) and $j = k, \dots\dots n + 1$

$(n+1)^{\text{th}}$ column is the solution vector

- ✓ Homework: Calculate the number of floating point operation required for solution using the *Gauss-Jordon* Algorithm!

- ✓ When is the *Gauss-Jordon algorithm* going to fail ?

- ✓ Inverse of a matrix ($n \times n$) can be computed using the *Gauss-Jordon Algorithm*:
  - ✓ Augment an identity matrix of order n with the matrix to be inverted. Resulting matrix will be ($n \times 2n$)
  - ✓ Carry out the operations using *Gauss-Jordon* Algorithm
  - ✓ Original matrix will become an identity matrix and the augmented identity matrix will become its inverse!

$$
\left[\begin{array}{cccccc|cccccc}
a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} & 1 & 0 & \dots & 0 & \dots & 0 \\
a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} & 0 & 1 & \dots & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} & 0 & 0 & \dots & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} & 0 & 0 & \dots & 0 & \dots & 1
\end{array}\right]
$$

Gauss-Jordon Algorithm:

For $k = 1, 2, \ldots n$

$$a_{kj} = \frac{a_{kj}}{a_{kk}} \qquad j = k, \ldots 2n$$

$a_{ij} = a_{ij} - a_{ik}\, a_{kj}$ $\quad i = 1, 2, 3, \ldots .n\ (\neq k)$ and $j = k, \ldots\ldots 2n$

Can you see why this inversion algorithm works?

LU-Decomposition: A general method

$$\boldsymbol{Ax} = \boldsymbol{b}$$

- ✓ In most engineering problem, the matrix ***A*** remains constant while the vector ***b*** changes with time. The matrix ***A*** describes the system and the vector ***b*** describes the external forcing. e.g., all network problems (pipes, electrical, canal, road, reactors, etc.); structural frames; many financial analyses.
- ✓ If all ***b***'s are available together, one can solve the system by augmented matrix but in practice, they are not!
- ✓ Instead of performing $\sim n^3$ floating point operations to solve whenever a new ***b*** becomes available, it is possible to solve the system by performing $\sim n^2$ floating point operations if a ***LU Decomposition*** is available for matrix ***A***
- ✓ *LU*-decomposition requires $\sim n^3$ floating point operations!

*Consider the system: (**b** changes!)*

$$\boldsymbol{Ax} = \boldsymbol{b}$$

- ✓ Perform a decomposition of the form $\boldsymbol{A} = \boldsymbol{LU}$, where $\boldsymbol{L}$ is a *lower-triangular* and $\boldsymbol{U}$ is an *upper-triangular* matrix!
- ✓ *LU*-decomposition requires ~ n^3 floating point operations!
- ✓ For any given $\boldsymbol{b}$, solve $\boldsymbol{Ax} = \boldsymbol{LUx} = \boldsymbol{b}$
- ✓ This is equivalent to solving two triangular systems:
  - ✓ Solve $\boldsymbol{Ly} = \boldsymbol{b}$ using *forward substitution* to obtain $\boldsymbol{y}$ (~n^2 operations)
  - ✓ Solve $\boldsymbol{Ux} = \boldsymbol{y}$ using *back substitution* to obtain $\boldsymbol{x}$ (~n^2 operations)
- ✓ Most frequently used method for engineering applications!
- ✓ We will derive *LU-decomposition* from Gauss Elimination!

An example of gauss elimination (four decimal places shown):

$$\begin{bmatrix} 3 & -1 & 1 \\ -2 & -5 & 3 \\ 1 & 3 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -11 \\ 4 \end{bmatrix}$$

l_{21} = -2/3 = -0.6667

l_{31} = 1/3 = 0.3333

$$\begin{bmatrix} 3 & -1 & 1 \\ 0 & -5.6667 & 3.6667 \\ 0 & 3.3333 & -3.3333 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -9.6667 \\ 3.3333 \end{bmatrix}$$

l_{32} = -3.3333/5.6667 = -0.5882

$$\begin{bmatrix} 3 & -1 & 1 \\ 0 & -5.6667 & 3.6667 \\ 0 & 0 & -1.1765 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -9.6667 \\ -2.3529 \end{bmatrix}$$

At this point, you may solve the system using back-substitution to obtain $x_1 = 1$, $x_2 = 3$ and $x_3 = 2$.

Check the following matrix identity:

$$\begin{bmatrix} 1 & 0 & 0 \\ -0.6667 & 1 & 0 \\ 0.3333 & -0.5882 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 & 1 \\ 0 & -5.6667 & 3.6667 \\ 0 & 0 & -1.1765 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 1 \\ -2 & -5 & 3 \\ 1 & 3 & -3 \end{bmatrix}$$

One can derive the general algorithm of LU-Decomposition by carefully studying Gauss Elimination!

Matrix after the k^{th} Step:

$$
\begin{array}{cl}
\left[\begin{array}{ccccccccc}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1k} & a_{1k+1} & \cdots & a_{1n} \\
0 & a_{22} & a_{23} & a_{24} & \cdots & a_{2k} & a_{2k+1} & \cdots & a_{2n} \\
0 & 0 & a_{33} & a_{34} & \cdots & a_{3k} & a_{3k+1} & \cdots & a_{3n} \\
0 & 0 & 0 & a_{44} & \cdots & a_{4k} & a_{4k+1} & \cdots & a_{4n} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & a_{kk} & a_{kk+1} & \cdots & a_{kn} \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{k+1k+1} & \cdots & a_{k+1n} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{nk+1} & \cdots & a_{nn}
\end{array}\right]
&
\begin{array}{l}
\text{Changed 0-times} \\
\text{Changed 1-time} \\
\text{Changed 2-times} \\
\text{Changed 3-times} \\
\\
\text{Changed } (k\text{-}1)\text{-times} \\
\\
\\
\\
\end{array}
\end{array}
$$

Changed 1-time and became zero

Changed 2-times and became zero

Changed k- times and became zero

Gauss-Elimination Steps (*example* 4×4 *matrix*):

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} \\ a_{41}^{(1)} & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} \end{bmatrix} \xrightarrow{\textit{Step 1}} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & a_{42}^{(2)} & a_{43}^{(2)} & a_{44}^{(2)} \end{bmatrix}$$

$$\xrightarrow{\textit{Step 2}} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & a_{43}^{(3)} & a_{44}^{(3)} \end{bmatrix} \xrightarrow{\textit{Step 3}} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & 0 & a_{44}^{(4)} \end{bmatrix}$$

$$
\begin{array}{l}
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1k} & a_{1k+1} & \cdots & a_{1n} \\
0 & a_{22} & a_{23} & a_{24} & \cdots & a_{2k} & a_{2k+1} & \cdots & a_{2n} \\
0 & 0 & a_{33} & a_{34} & \cdots & a_{3k} & a_{3k+1} & \cdots & a_{3n} \\
0 & 0 & 0 & a_{44} & \cdots & a_{4k} & a_{4k+1} & \cdots & a_{4n} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & a_{kk} & a_{kk+1} & \cdots & a_{kn} \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{k+1k+1} & \cdots & a_{k+1n} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{nk+1} & \cdots & a_{nn}
\end{bmatrix}
\end{array}
\begin{array}{l}
\text{Changed 0-times} \\
\text{Changed 1-time} \\
\text{Changed 2-times} \\
\text{Changed 3-times} \\
\\
\text{Changed } (k\text{-1})\text{-times} \\
\\
\\
\\
\end{array}
$$

Changed 1-time and became zero

Changed 2-times and became zero

Changed k- times and became zero

For elements above and on the diagonal $i \le j$:

- a_{ij} is actively *modified* for the first $(i - 1)$ steps and remains constant for the rest $(n - i)$ steps

For elements below on the diagonal $j < i$:

- a_{ij} is actively *modified* for the first j steps and remains at zero for the rest $(n - j)$ steps

Combined statement:

Any element a_{ij} is actively *modified* for the first p steps where, $p = \min \{(i\text{-}1), j\}$