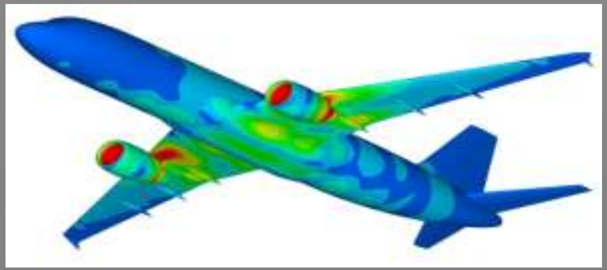
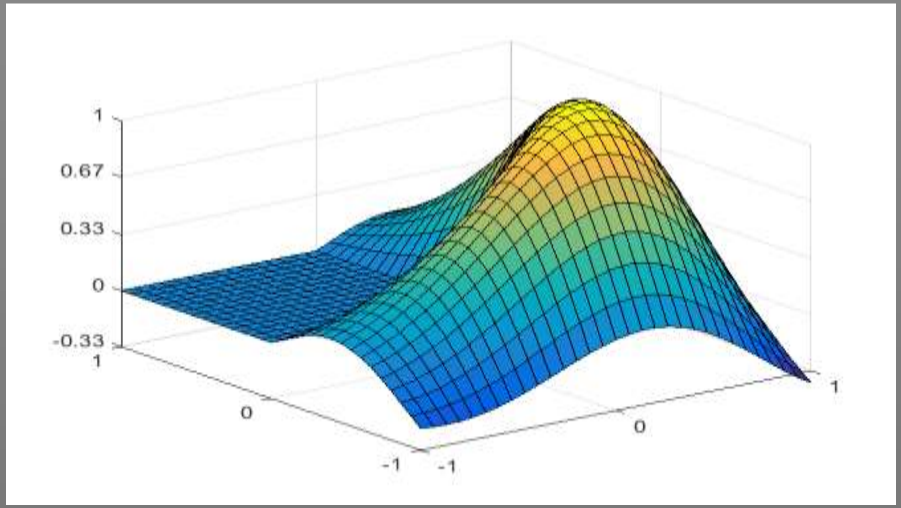
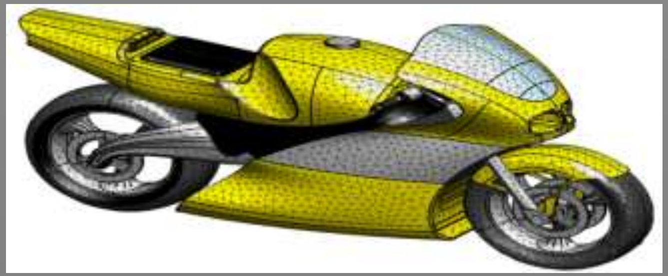
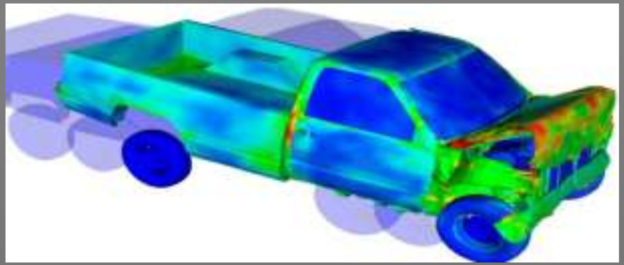
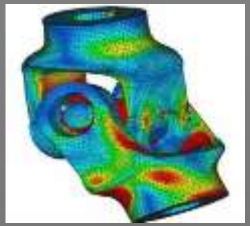
Linear Systems of Equations

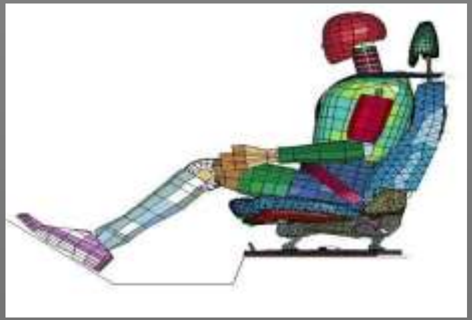












Dr. Himanshu Pathak

himanshu@iitmandi.ac.in

Systems of Equations

- Recall that a system of two linear equations in two variables may be written in the general form

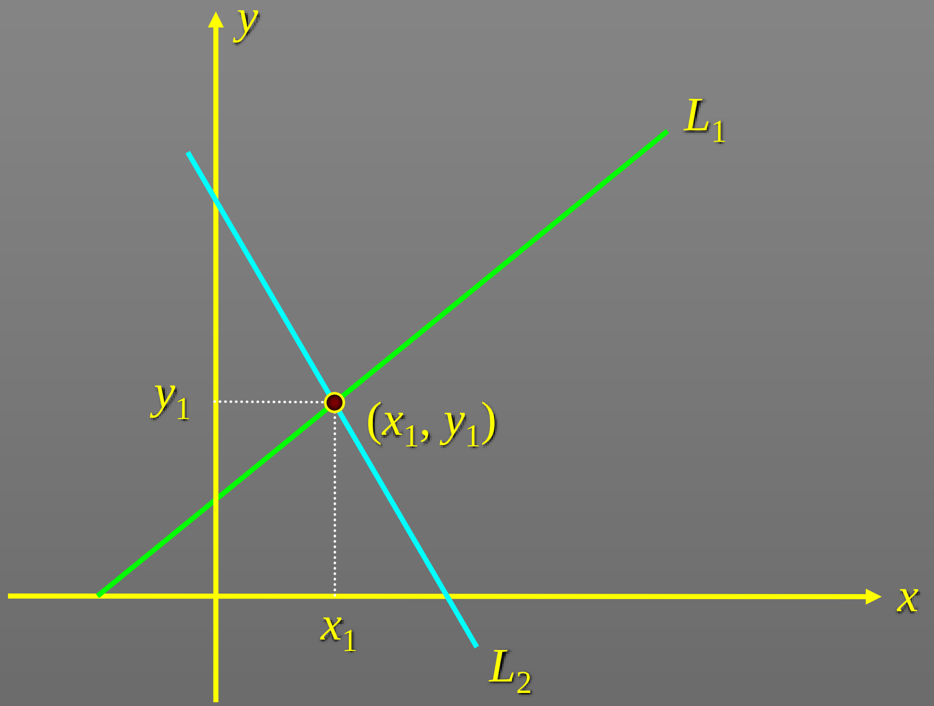
$$\begin{aligned} ax + by &= h \\ cx + dy &= k \end{aligned}$$

where a, b, c, d, h, and k are real numbers and neither a and b nor c and d are both zero.

- Recall that the graph of each equation in the system is a straight line in the plane, so that geometrically, the solution to the system is the point(s) of intersection of the two straight lines L_1 and L_2, represented by the first and second equations of the system.

Systems of Equations

- Given the two straight lines L_1 and L_2, **one and only one** of the following may occur:
  1. L_1 and L_2 intersect at **exactly one point**.

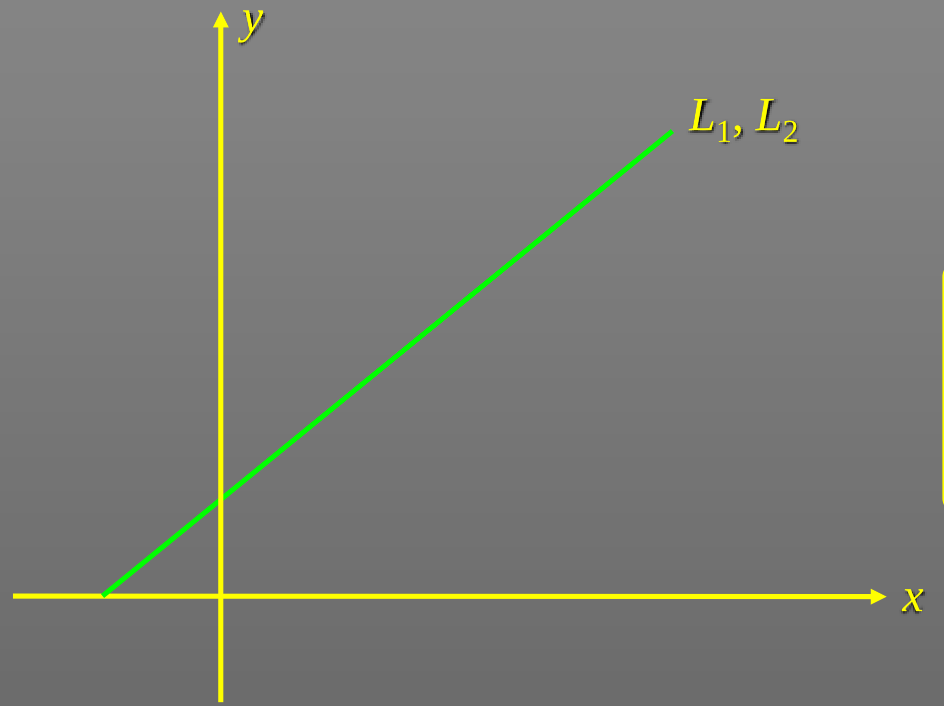


Unique solution
(x_1, y_1)

Systems of Equations

- Given the two straight lines L_1 and L_2, one and only one of the following may occur:

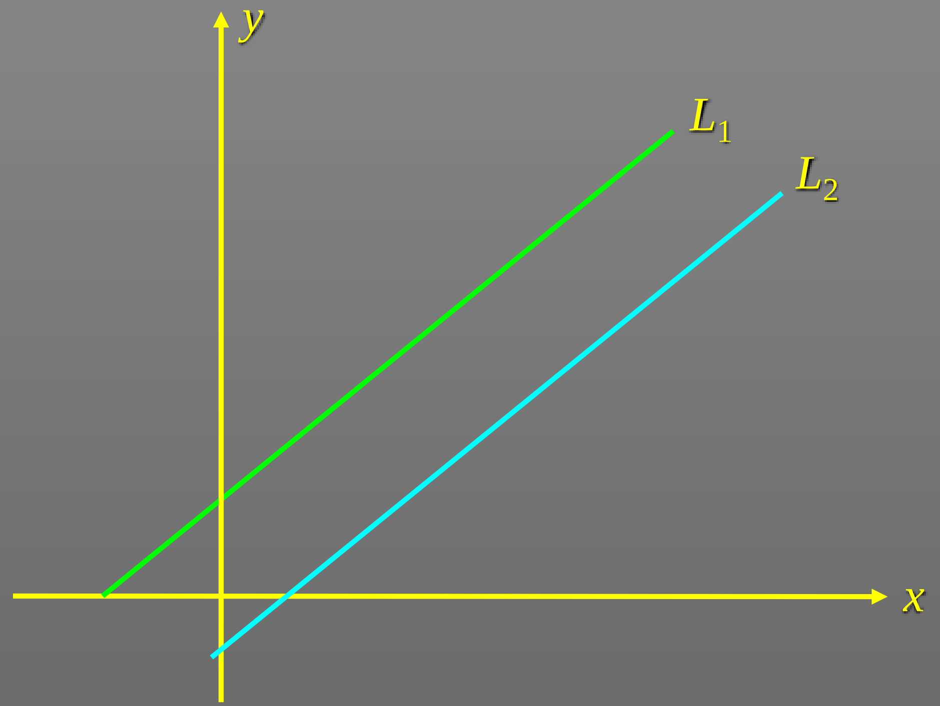
  2. L_1 and L_2 are coincident.



Infinitely many solutions

Systems of Equations

- Given the two straight lines L_1 and L_2, **one and only one** of the following may occur:
  3. L_1 and L_2 are **parallel**.



No solution

Example:

A System of Equations With Exactly One Solution

- Consider the system

$$2x - y = 1$$
$$3x + 2y = 12$$

- Solving the first equation for y in terms of x, we obtain

$$y = 2x - 1$$

- Substituting this expression for y into the second equation yields

$$3x + 2(2x - 1) = 12$$
$$3x + 4x - 2 = 12$$
$$7x = 14$$
$$x = 2$$

Example:
A System of Equations With Exactly One Solution

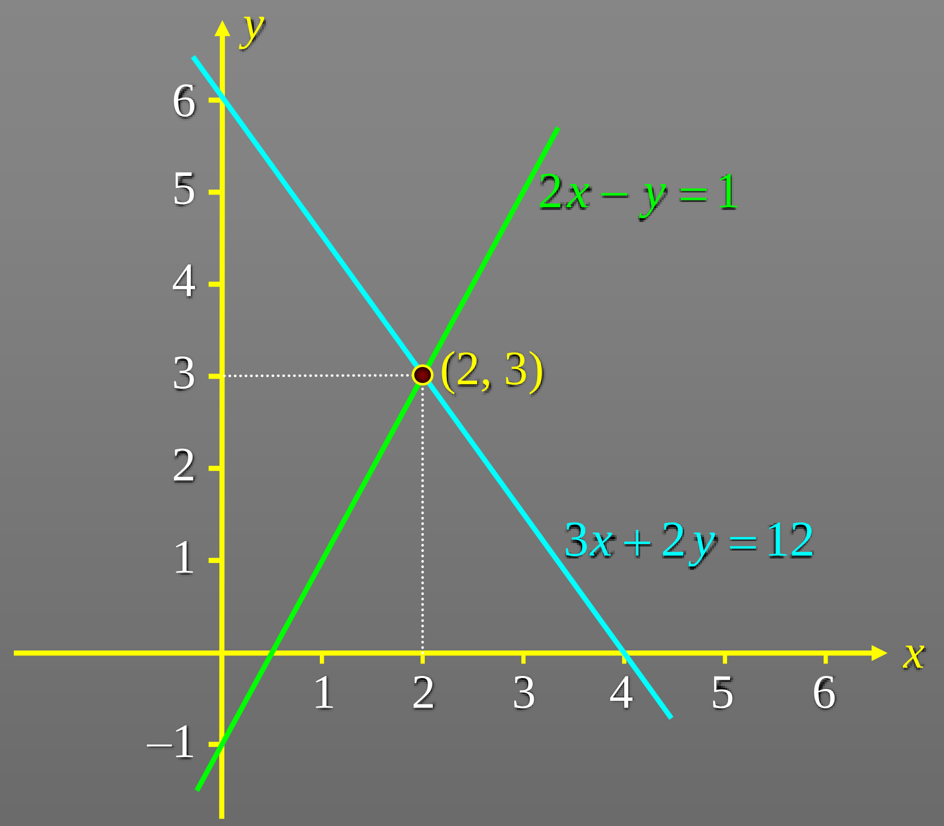
- Finally, substituting this value of x into the expression for y obtained earlier gives

$$\begin{aligned} y &= 2x - 1 \\ &= 2(2) - 1 \\ &= 3 \end{aligned}$$

- Therefore, the unique solution of the system is given by $x = 2$ and $y = 3$.

Example:
A System of Equations With Exactly One Solution

- **Geometrically, the two lines represented by the two equations that make up the system intersect at the point (2, 3):**



Example:
A System of Equations With Infinitely Many Solutions

- Consider the system

$$2x - y = 1$$
$$6x - 3y = 3$$

- **Solving** the **first equation** for y in terms of x, we obtain

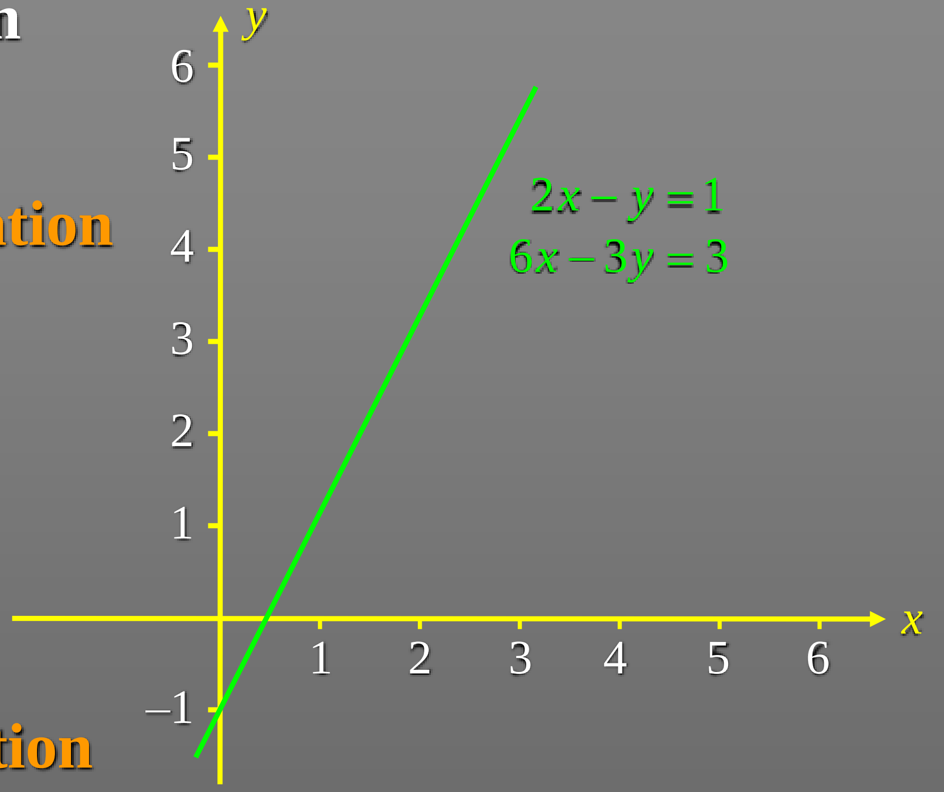
$$y = 2x - 1$$

- **Substituting** this expression for y into the **second equation** yields

$$6x - 3(2x - 1) = 3$$
$$6x - 6x + 3 = 3$$
$$0 = 0$$

which is a **true statement**.

- This result follows from the fact that the **second equation** is **equivalent** to the **first**.



Example:
A System of Equations That Has No Solution

- Consider the system

$$\begin{aligned} 2x - y &= 1 \\ 6x - 3y &= 12 \end{aligned}$$

- **Solving** the **first equation** for y in terms of x, we obtain

$$y = 2x - 1$$

- **Substituting** this expression for y into the **second equation** yields

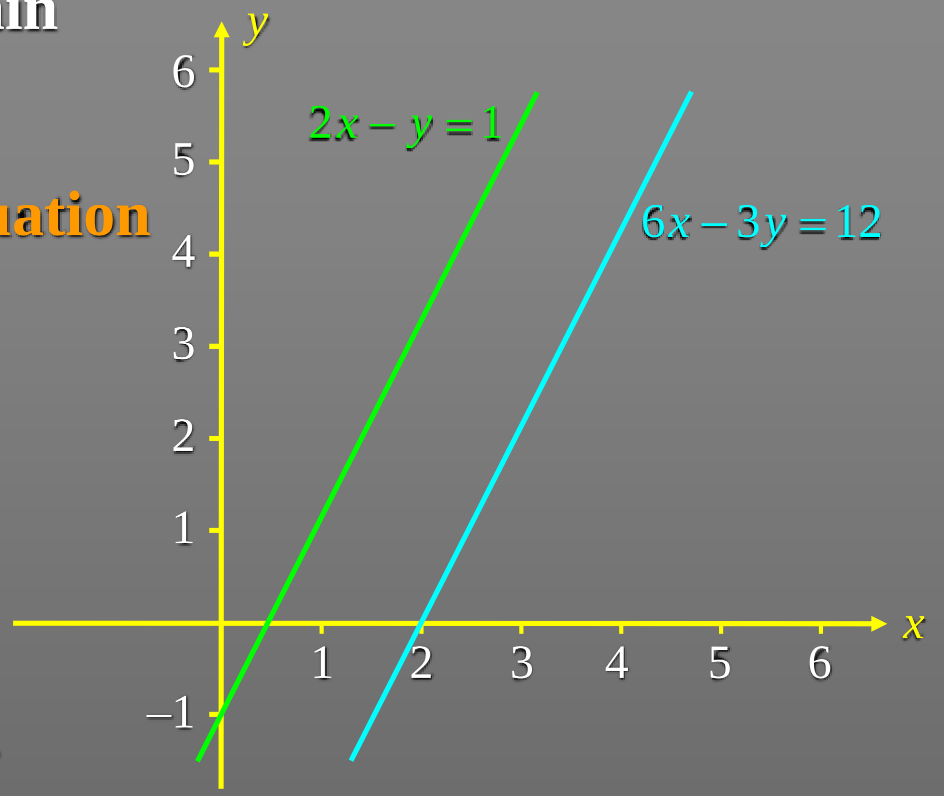
$$6x - 3(2x - 1) = 12$$

$$6x - 6x + 3 = 12$$

$$0 = 9$$

which is **clearly impossible**.

- Thus, there is **no solution** to the system of equations.



The Gauss Elimination Method

- The Gauss elimination method is a technique for solving systems of linear equations of any size.
- The operations of the Gauss-elimination method are
  1. Convert into upper triangle matrix.
  2. Get one of known value from last equation.
  3. Substitute the known variable into upper equations (Back Substitution).

Example

- **Solve the following system of equations:**

$$\begin{aligned} 2x + 4y + 6z &= 22 \\ 3x + 8y + 5z &= 27 \\ -x + y + 2z &= 2 \end{aligned}$$

<u>Solution</u>

- **First, we transform this system into an equivalent system in which the coefficient of x in the first equation is 1:**

$$\begin{aligned} 2x + 4y + 6z &= 22 \\ 3x + 8y + 5z &= 27 \\ -x + y + 2z &= 2 \end{aligned}$$

Multiply the equation by 1/2

Example

- Solve the following system of equations:

$$\begin{aligned} 2x + 4y + 6z &= 22 \\ 3x + 8y + 5z &= 27 \\ -x + \quad y + 2z &= \quad 2 \end{aligned}$$

<u>**Solution**</u>

- **First, we transform this system into an equivalent system in which the coefficient of x in the first equation is 1:**

$$\begin{aligned} x + 2y + 3z &= 11 \longleftarrow \\ 3x + 8y + 5z &= 27 \\ -x + \quad y + 2z &= \quad 2 \end{aligned}$$

Multiply the first equation by 1/2

Example

- Solve the following system of equations:

$$
\begin{aligned}
2x + 4y + 6z &= 22 \\
3x + 8y + 5z &= 27 \\
-x + \ \ y + 2z &= \ \ 2
\end{aligned}
$$

Solution

- Next, we eliminate the variable x from all equations except the first:

$$
\begin{aligned}
x + 2y + 3z &= 11 \\
3x + 8y + 5z &= 27 \\
-x + \ \ y + 2z &= \ \ 2
\end{aligned}
$$

Replace by the sum of
-3 X the first equation
$+$ the second equation

Example

- Solve the following system of equations:

$$
\begin{aligned}
2x + 4y + 6z &= 22 \\
3x + 8y + 5z &= 27 \\
-x + y + 2z &= 2
\end{aligned}
$$

<u>Solution</u>

- Next, we eliminate the variable x from all equations except the first:

$$
\begin{aligned}
x + 2y + 3z &= 11 \\
2y - 4z &= -6 \\
-x + y + 2z &= 2
\end{aligned}
$$

Replace by the sum of
$-3 \times$ the first equation
$+$ the second equation

Example

- Solve the following system of equations:

$$\begin{aligned} 2x + 4y + 6z &= 22 \\ 3x + 8y + 5z &= 27 \\ -x + y + 2z &= 2 \end{aligned}$$

<u>Solution</u>

- Next, we eliminate the variable x from all equations except the first:

$$\begin{aligned} x + 2y + 3z &= 11 \\ 2y - 4z &= -6 \\ -x + y + 2z &= 2 \end{aligned}$$

Replace by the sum
of the first equation
+ the third equation

Example

- Solve the following system of equations:

$$\begin{aligned} 2x + 4y + 6z &= 22 \\ 3x + 8y + 5z &= 27 \\ -x + \quad y + 2z &= \quad 2 \end{aligned}$$

Solution

- Next, we eliminate the variable x from all equations except the first:

$$\begin{aligned} x + 2y + 3z &= 11 \\ 2y - 4z &= -6 \\ 3y + 5z &= 13 \end{aligned}$$

← Replace by the sum of the first equation + the third equation

Example

- **Solve the following system of equations:**

$$\begin{aligned} 2x + 4y + 6z &= 22 \\ 3x + 8y + 5z &= 27 \\ -x + y + 2z &= 2 \end{aligned}$$

Solution

- **Then we transform so that the coefficient of y in the second equation is 1:**

$$\begin{aligned} x + 2y + 3z &= 11 \\ 2y - 4z &= -6 \\ 3y + 5z &= 13 \end{aligned}$$

Multiply the second equation by 1/2

Example

- Solve the following system of equations:

$$
\begin{aligned}
2x + 4y + 6z &= 22 \\
3x + 8y + 5z &= 27 \\
-x + \quad y + 2z &= \ \ 2
\end{aligned}
$$

<u>**Solution**</u>

- Then we transform so that the coefficient of y in the **second equation** is 1:

$$
\begin{aligned}
x + 2y + 3z &= 11 \\
y - 2z &= -3 \\
3y + 5z &= 13
\end{aligned}
$$

Multiply the second equation by 1/2

Example

- Solve the following system of equations:

$$
\begin{aligned}
2x + 4y + 6z &= 22 \\
3x + 8y + 5z &= 27 \\
-x + y + 2z &= 2
\end{aligned}
$$

Solution

- We now eliminate y from third equation:

$$
\begin{aligned}
x + 2y + 3z &= 11 \\
y - 2z &= -3 \\
3y + 5z &= 13 \quad \longleftarrow
\end{aligned}
$$

Replace by the sum of
the third equation +
(–3) × the second equation

Example

- **Solve the following system of equations:**

$$
\begin{aligned}
2x + 4y + 6z &= 22 \\
3x + 8y + 5z &= 27 \\
-x + y + 2z &= 2
\end{aligned}
$$

Solution

- **We now eliminate y from third equation:**

$$
\begin{aligned}
x + 2y + 3z &= 11 \\
y - 2z &= -3 \\
11z &= 22
\end{aligned}
$$

Example

- Solve the following system of equations:

$$\begin{aligned} 2x + 4y + 6z &= 22 \\ 3x + 8y + 5z &= 27 \\ -x + y + 2z &= 2 \end{aligned}$$

<u>Solution</u>

- Thus, after back substitution the solution to the system is $x = 3$, $y = 1$, and $z = 2$.

$$\begin{aligned} x \qquad\qquad &= 3 \\ y \qquad &= 1 \\ z &= 2 \end{aligned}$$

Example

- **Solve the following system of equations:**

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}; \quad b=\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}; \quad x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Step: 1

- **Pivot is a_{11};**

multiply first row by (- a_{21}/ a_{11}) and add the results to second row

$$m=\left(-a_{21}/a_{11}\right); a_{21}\Leftarrow 0; a_{22}\Leftarrow a_{22}+ma_{12}; a_{23}\Leftarrow a_{23}+ma_{13}; a_{24}\Leftarrow a_{24}+ma_{14}; b_2\Leftarrow b_2+mb_1;$$

multiply first row by (- a_{31}/ a_{11}) and add the results to third row

$$m=\left(-a_{31}/a_{11}\right); a_{31}\Leftarrow 0; a_{32}\Leftarrow a_{32}+ma_{12}; a_{33}\Leftarrow a_{33}+ma_{13}; a_{34}\Leftarrow a_{34}+ma_{14}; b_3\Leftarrow b_3+mb_1;$$

multiply first row by (- a_{41}/ a_{11}) and add the results to forth row

$$m=\left(-a_{41}/a_{11}\right); a_{41}\Leftarrow 0; a_{42}\Leftarrow a_{42}+ma_{12}; a_{43}\Leftarrow a_{43}+ma_{13}; a_{44}\Leftarrow a_{44}+ma_{14}; b_4\Leftarrow b_4+mb_1;$$

Example

- **Solve the following system of equations:**

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}; \quad b=\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}; \quad x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Step: 2

- **Pivot is a_{22};**

multiply second row by (- a_{32}/ a_{22}) and add the results to third row

$$m=\left({}^{-a_{32}}\!\!/_{a_{22}}\right); a_{31} \Leftarrow 0; a_{32} \Leftarrow 0; a_{33} \Leftarrow a_{33}+ma_{23}; a_{34} \Leftarrow a_{34}+ma_{24}; b_3 \Leftarrow b_3+mb_2;$$

multiply second row by (- a_{42}/ a_{22}) and add the results to fourth row

$$m=\left({}^{-a_{42}}\!\!/_{a_{22}}\right); a_{41} \Leftarrow 0; a_{42} \Leftarrow 0; a_{43} \Leftarrow a_{43}+ma_{23}; a_{44} \Leftarrow a_{44}+ma_{24}; b_4 \Leftarrow b_4+mb_2;$$

Example

- **Solve the following system of equations:**

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}; \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}; \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Step: 3

- **Pivot is a_{33};**

multiply third row by (- a_{43}/ a_{33}) and add the results to fourth row

$$m = \left({-a_{43}} \middle/ {a_{33}} \right); a_{41} \Leftarrow 0; a_{42} \Leftarrow 0; a_{43} \Leftarrow 0; a_{44} \Leftarrow a_{44} + m a_{34}; b_4 \Leftarrow b_4 + m b_3;$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Example

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Step: 4

- **Back Substitution;**

$$x_4 = {-b_4}/{a_{44}}$$

$$x_3 = \frac{b_3 - a_{34}x_4}{a_{33}}$$

$$x_2 = \frac{b_2 - a_{23}x_3 - a_{24}x_4}{a_{22}}$$

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3 - a_{14}x_4}{a_{11}}$$

MATLAB© Script

Simplest Gauss Elimination Program within MATLAB to solve linear system of equations

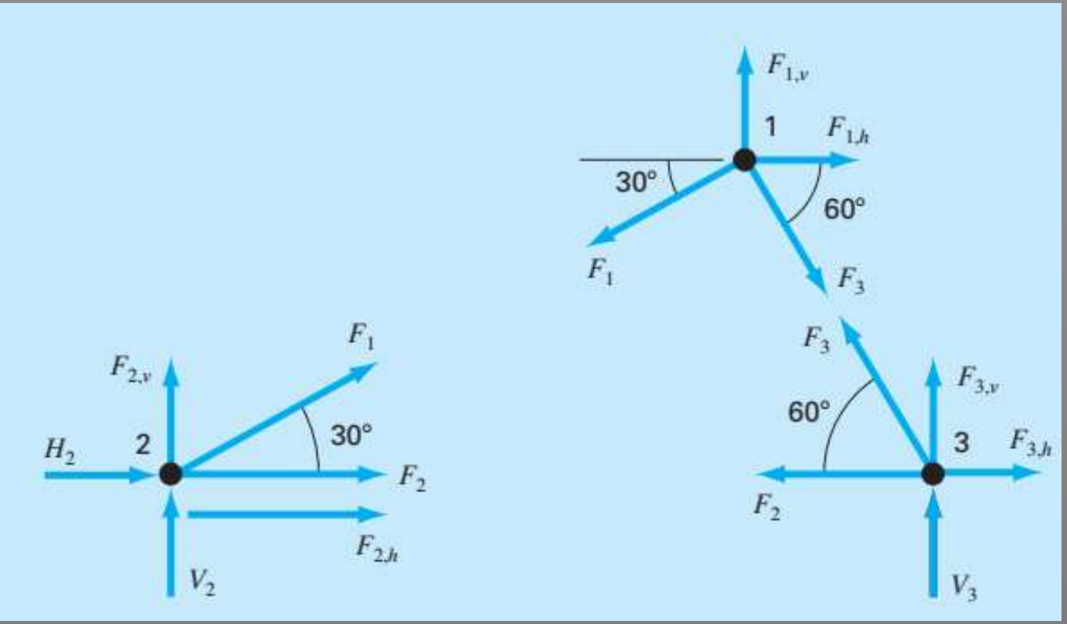
- **Coefficient Matrices**

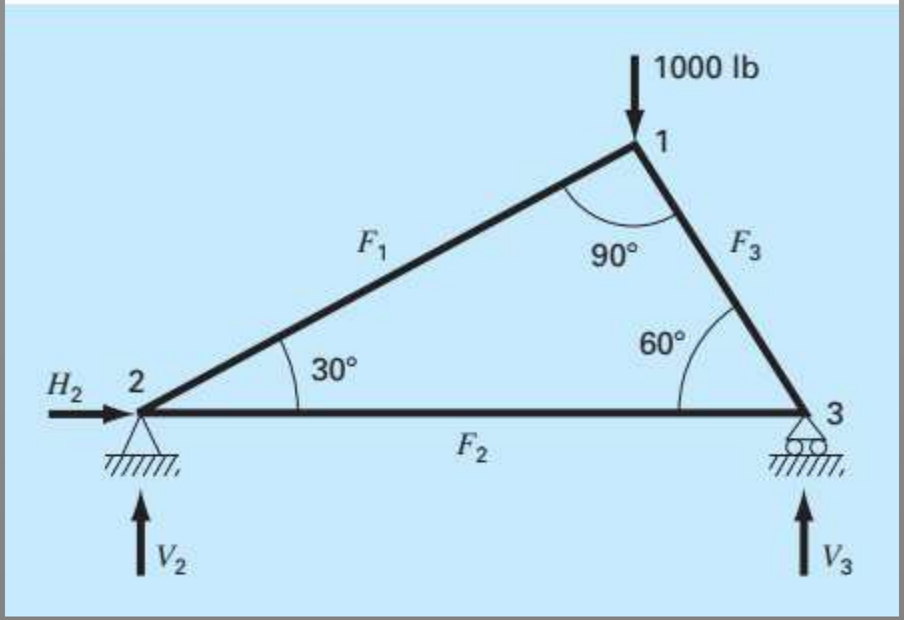
$$A = \begin{bmatrix} 3 & -1 & 2 \\ 1 & 1 & 2 \\ 2 & -2 & -1 \end{bmatrix}; \quad b = \begin{bmatrix} 7 \\ 9 \\ -5 \end{bmatrix}; \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

```
[n, nn] = size(A);                 % Check the size of coefficient matrix
C = [A b];                         % Create Augmented matrix
for i = 1: n-1                     % Loop over pivot row (i)
        for k = i+1 : n            % Elimination on row k
                m(k, i) = - C(k, i) / C(i, i);              % Coefficient for elimination
                C(k, :) = C(k, :) + m(k, i) * C(i, :);      % Elimination done at kth row
        end
end
x(n, 1) = C(n, n+1) / C(n, n);                              % Last variable
for i = n-1: -1: 1                                          % Loop for back substitution
        x(i, 1) = (C(i, n+1) – C(i, i+1:n) * x(i+1: n, 1)) / C(i, i);        % Back substitution
end
X=rref(C) % Solution from inbuilt function to check correctness of result
```

Implementation

- **Analysis of a Statically Determinate Truss**





At node 1,

$$\Sigma F_H = 0 = -F_1 \cos 30^\circ + F_3 \cos 60^\circ + F_{1,h}$$
$$\Sigma F_V = 0 = -F_1 \sin 30^\circ - F_3 \sin 60^\circ + F_{1,v}$$

At node 2,

$$\Sigma F_H = 0 = F_2 + F_1 \cos 30^\circ + F_{2,h} + H_2$$
$$\Sigma F_V = 0 = F_1 \sin 30^\circ + F_{2,v} + V_2$$

At node 3,

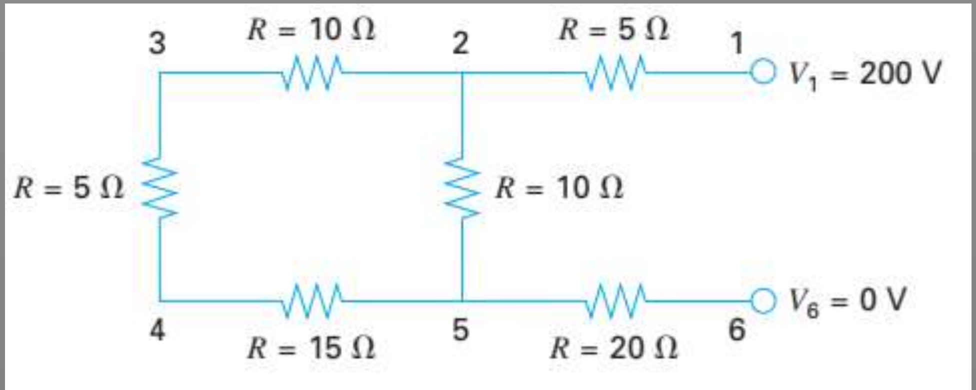
$$\Sigma F_H = 0 = -F_2 - F_3 \cos 60^\circ + F_{3,h}$$
$$\Sigma F_V = 0 = F_3 \sin 60^\circ + F_{3,v} + V_3$$

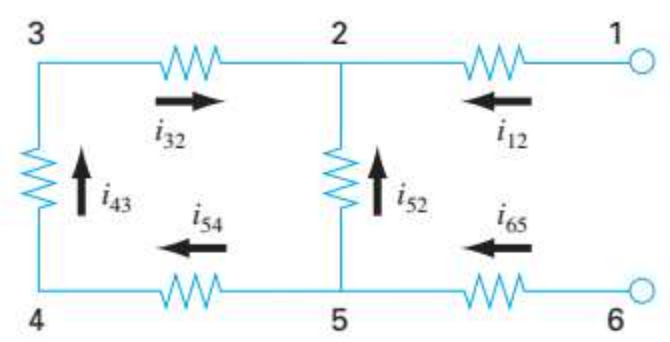
$$\begin{bmatrix} 0.866 & 0 & -0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0.866 & 0 & 0 & 0 \\ -0.866 & -1 & 0 & -1 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & -0.866 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ H_2 \\ V_2 \\ V_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -1000 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

$$F_1 = -500 \qquad F_2 = 433 \qquad F_3 = -866$$
$$H_2 = 0 \qquad V_2 = 250 \qquad V_3 = 750$$

Implementation

- **Analysis of a Resistor Circuit**





Kirchhoff's current rule:

$$
\begin{aligned}
i_{12} + i_{52} + i_{32} &= 0 \\
i_{65} - i_{52} - i_{54} &= 0 \\
i_{43} - i_{32} &= 0 \\
i_{54} - i_{43} &= 0
\end{aligned}
$$

Kirchhoff's voltage rule:

$$
\begin{aligned}
-i_{54}R_{54} - i_{43}R_{43} - i_{32}R_{32} + i_{52}R_{52} &= 0 \\
-i_{65}R_{65} - i_{52}R_{52} - i_{12}R_{12} - 200 &= 0
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{aligned}
-15i_{54} - 5i_{43} - 10i_{32} + 10i_{52} &= 0 \\
-20i_{65} - 10i_{52} + 5i_{12} &= 200
\end{aligned}
$$

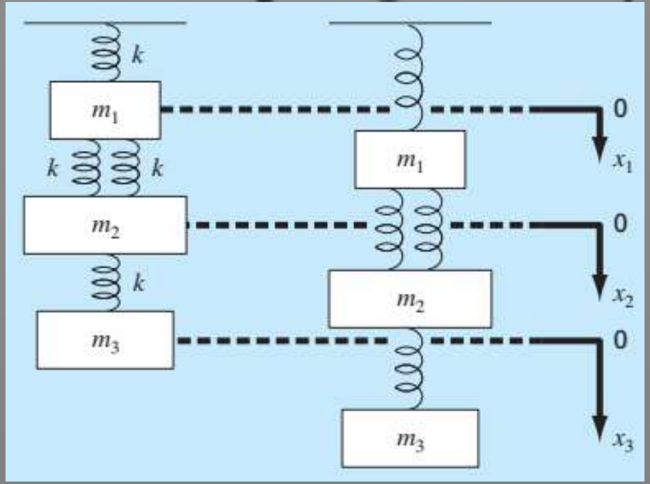
$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
0 & -1 & 0 & 1 & -1 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & -1 \\
0 & 10 & -10 & 0 & -15 & -5 \\
5 & -10 & 0 & -20 & 0 & 0
\end{bmatrix}
\begin{Bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \end{Bmatrix}
=
\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 200 \end{Bmatrix}
$$

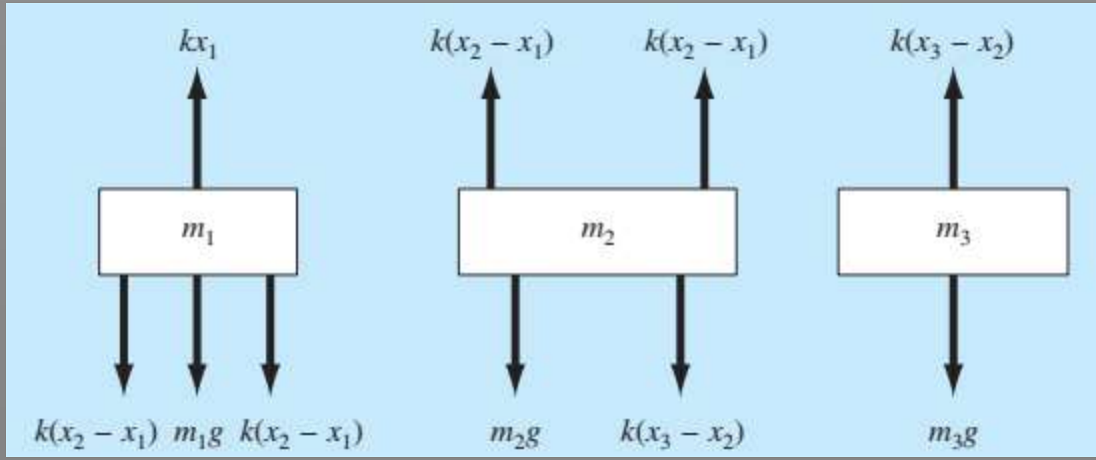
$$\Downarrow$$

$$
\begin{aligned}
i_{12} &= 6.1538 & i_{52} &= -4.6154 & i_{32} &= -1.5385 \\
i_{65} &= -6.1538 & i_{54} &= -1.5385 & i_{43} &= -1.5385
\end{aligned}
$$

Implementation

- **Analysis of a Spring-Mass System**





Newton's 2nd Law:

For 1st mass: $$m_1 \frac{d^2 x_1}{dt^2} = 2k(x_2 - x_1) + m_1 g - kx_1$$

For 2nd mass: $$m_2 \frac{d^2 x_2}{dt^2} = k(x_3 - x_2) + m_2 g - 2k(x_2 - x_1)$$

For 3rd mass: $$m_3 \frac{d^2 x_3}{dt^2} = m_3 g - k(x_3 - x_2)$$

$$\begin{array}{rcrcrcl} 3kx_1 & - & 2kx_2 & & & = & m_1 g \\ -2kx_1 & + & 3kx_2 & - & kx_3 & = & m_2 g \\ & - & kx_2 & + & kx_3 & = & m_3 g \end{array}$$

$$[K]\{X\} = \{W\}$$

Other Related Issues

- **If Pivot element is zero or near to zero**

$$\begin{aligned} 2x_2 + 3x_3 &= 8 \\ 4x_1 + 6x_2 + 7x_3 &= -3 \\ 2x_1 + x_2 + 6x_3 &= 5 \end{aligned}$$

- **Gauss Elimination with Row Pivoting**

$$\begin{aligned} 0.0001x + y &= 3 \\ x + 2y &= 5 \end{aligned} \qquad \begin{aligned} x + 2y &= 5 \\ 0.0001x + y &= 3 \end{aligned}$$

- **Round-off errors:** *It accumulated and propagated during the elimination process; Important when dealing with 100 or more equations.*
- **Ill-Conditioned Systems:** *Small changes in coeffcients result in large changes in the solution.*

  Determinant of system matrix is near to zero.

Improvement Techniques

- **Gauss Elimination with Row Pivoting**

$$\begin{aligned} 0.0001x + y &= 3 \\ x + 2y &= 5 \end{aligned} \qquad \begin{aligned} x + 2y &= 5 \\ 0.0001x + y &= 3 \end{aligned}$$

- **More significant number in solution**

$$\begin{aligned} 0.0003x_1 + 3.0000x_2 &= 2.0001 \\ 1.0000x_1 + 1.0000x_2 &= 1.0000 \end{aligned}$$

- **Gauss-Jordan:** *forward elimination for below diagonal elements backward elimination for above diagonal elements diagonal elements scaled to one*

(Reduced Row Echelon Form).

$$\begin{aligned} 3x_1 - x_2 + 2x_3 &= 7 \\ x_1 + x_2 + 2x_3 &= 9 \\ 2x_1 - 2x_2 - x_3 &= -5 \end{aligned} \Longrightarrow \begin{bmatrix} 3 & -1 & 2 & 7 \\ 1 & 1 & 2 & 9 \\ 2 & -2 & -1 & -5 \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 0 & 0 & b_1 \\ 0 & 1 & 0 & b_2 \\ 0 & 0 & 1 & b_3 \end{bmatrix}$$

Other Related Issues

- **Tridiagonal Systems (in FEM): Pivot element scaled to 1 preceding Gauss Elimination.**

$$
\begin{aligned}
2x_1 + 2x_2 \qquad\qquad\qquad &= 4 \\
2x_1 + 4x_2 + 4x_3 \qquad &= 6 \\
2x_2 + 3x_3 + 3x_4 &= 7 \\
2x_3 + 5x_4 &= 10
\end{aligned}
$$

- **Computational Effort: floating point operations (flops)**
- **Flops: number of executable operations in terms of multiplication and division.**

- **Gauss Elimination:**

$$flops \approx \frac{1}{3} n^3$$

LU Decomposition

- Gauss Elimination solves [A]{x}={B} by forward elimination and back substitution employed over Augmented matrix [A | B].

- It becomes insufficient when solving these equations for different values of {B}

- LU decomposition is very useful when the vector of variables {x} is estimated for different parameter vectors {B} since the forward elimination process employed over [A] only.

LU Decomposition

L: lower triangular matrix; U: upper triangular matrix
Then, [A]{X}={B} can be decomposed into two matrices [L] and [U] such that: [L][U]=[A]

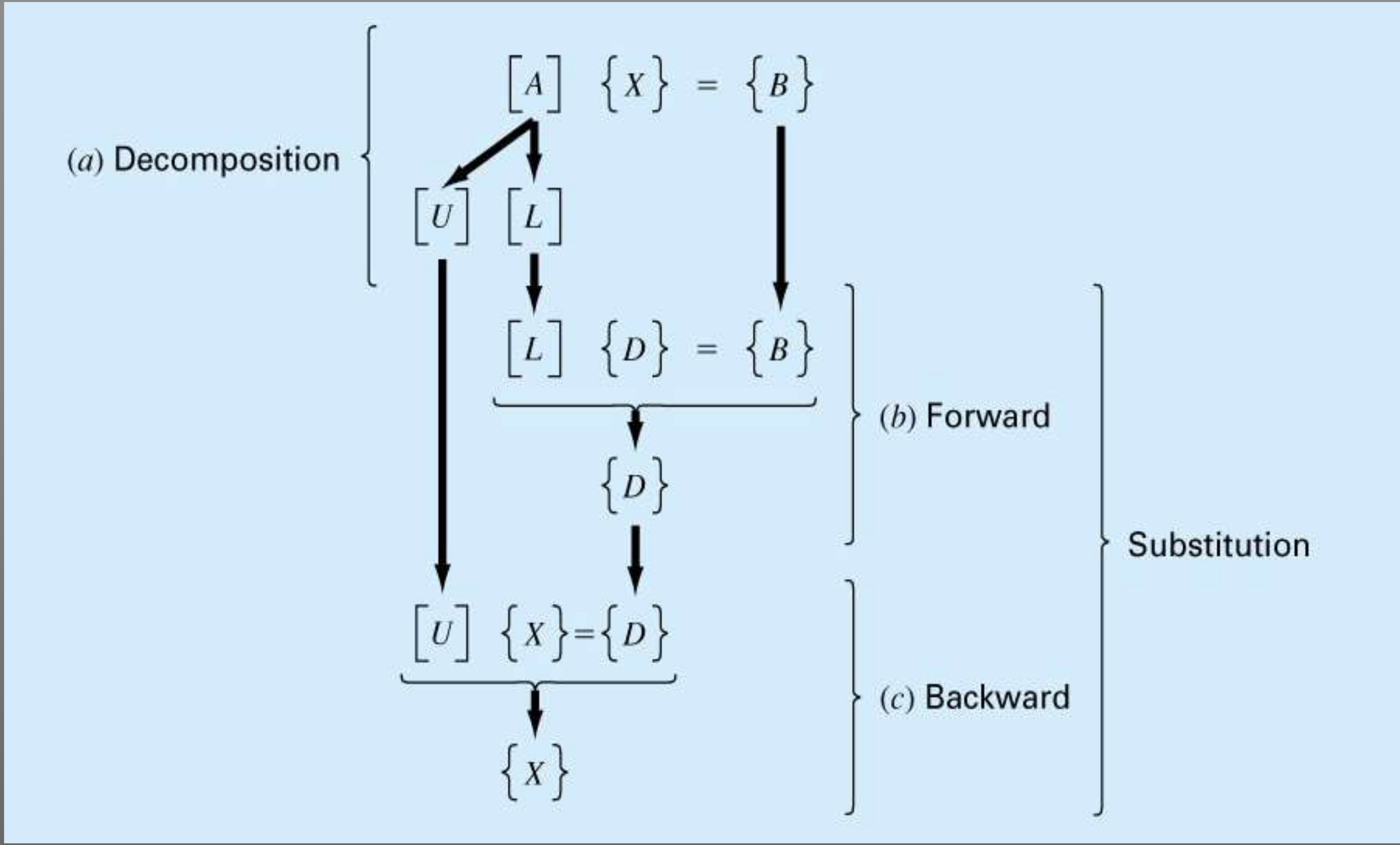
LU Decomposition

[A]{X}={B} can be solved by LU decomposition using following steps:

1. [L][U] = [A] → ([L] [U]) {X} = {B}
2. Consider: [U]{X} = {D}

   So, [L]{D} = {B}
3. [L] {D} = {B} is used to generate an intermediate vector {D} by forward substitution.
4. [U] {X} = {D} is used to get {X} by back substitution.

LU Decomposition



LU Decomposition

- **System of linear equations [A]{x}={B}**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

Step 1: Decomposition

$$[L][U] = [A]$$

$$l_{21} = \frac{a_{21}}{a_{11}}; \quad l_{31} = \frac{a_{31}}{a_{11}}$$

$$l_{32} = \frac{a_{32}}{a_{22}}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

LU Decomposition

Step 2: **Generate an intermediate vector {D} by forward substitution**

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \Longrightarrow$$

$$d_1 = b_1; \quad d_2 = b_2 - l_{21} d_1$$
$$d_3 = b_3 - l_{31} d_1 - l_{32} d_2$$

Step 3: **Get {X} by back substitution.**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} \Longrightarrow$$

$$x_3 = \frac{d_3}{a_{33}}; \quad x_2 = \frac{d_2 - a_{23} x_3}{a_{22}}$$
$$x_1 = \frac{d_1 - a_{12} x_2 - a_{13} x_3}{a_{11}}$$

MATLAB© Script

Simplest LU Decomposition Program within MATLAB

- **Linear Equation**

$$3x_1 - x_2 + 2x_3 = 7$$
$$x_1 + x_2 + 2x_3 = 9$$
$$2x_1 - 2x_2 - x_3 = -5$$

```
[n, m] = size(A);                 % Check the size of coefficient matrix
L = eye(n);                       % Initialize L matrix with diagonal 1
U = A;                            % Initially U matrix start with value of A matrix
for i = 1: n-1                    % Loop over pivot row (i)
        for j = i+1 : n           % Elimination on row j
                L(j, i) = U(j, i) / U(i, i);            % Coefficient for elimination
                U(j, :) = U(j, :) - L(j, i) * U(i, :);  % Elimination done at jth row
        end
end
```

MATLAB© Script

Simplest forward and backward substitution program to calculate intermediate matrix D

- **[L] {D} = {B}**

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

- **[U]{X} = {D}**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}$$

1. ***Forward substitution***

```
d(1, 1) = b(1, 1);                                       % First variable
for i = 2:n                                              % Loop for forward substitution
        d(i, 1) = b(i, 1) – L(i, 1:i-1) * d(1: i-1, 1);  % Forward substitution
end
```

2. ***Backward substitution***

```
x(n, 1) = d(n, 1) / U(n, n);                                          % Last variable
for i = n-1: -1: 1                                                    % Loop for back substitution
        x(i, 1) = (d(i, 1) – U(i, i+1:n) * x(i+1: n, 1)) / U(i, i);   % Back substitution
end
```

LU Decomposition for Matrix Inverse

- **Matrix**

$$[A] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

LU Decomposition

$$[U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \qquad [L] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix}$$

For first column calculation

- [L] {D} = {P}

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

$$P = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

- [U] {X} = {D}

Here, {X} is the first column for $[A]^{-1}$

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ -0.03333 \\ -0.1009 \end{Bmatrix}$$

LU Decomposition for Matrix Inverse

For second column calculation

- [L] {D} = {P}
- [U] {X} = {D}

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$$

$$P = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0.004944 & 0 \\ -0.00518 & 0.142903 & 0 \\ -0.01008 & 0.00271 & 0 \end{bmatrix}$$

Here, {X} is the second column for $[A]^{-1}$

For third column calculation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

$$P = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0.004944 & 0.006798 \\ -0.00518 & 0.142903 & 0.004183 \\ -0.01008 & 0.00271 & 0.09988 \end{bmatrix}$$

Introduction

- If systems of linear equations are very large, the computational effort of direct methods is prohibitively expensive.
- Three common classical iterative techniques for linear systems
  - The Jacobi method
  - Gauss-Seidel method
  - Successive Over Relaxation (SOR) method
- For systems that have coefficient matrices with the appropriate structure – especially large, sparse systems (many coefficients whose value is zero) – iterative techniques may be preferable.

$$
\begin{bmatrix}
4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4
\end{bmatrix}
\cdot
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \\ u_{15} \\ u_{16} \end{bmatrix}
=
\begin{bmatrix} 0.08 \\ 0.16 \\ 0.36 \\ 1.64 \\ 0.16 \\ 0.0 \\ 0.0 \\ 1.0 \\ 0.36 \\ 0 \\ 0 \\ 1.0 \\ 1.64 \\ 1.0 \\ 1.0 \\ 2.0 \end{bmatrix}
$$

Iterative Methods

- A system of equations should be converted into matrix form.

$$[A]\{x\}=\{b\}$$

- The diagonal variable will be evaluated with initial guess.

$$x_i^0=0$$

$$x_i^j=\sum_{i=1}^{n}\frac{a_{ik}}{a_{ii}}x_k^{j-1}+\frac{b_i}{a_{ii}} \quad \Leftrightarrow \quad x^j=Cx^{j-1}+d$$

Jacobi Iteration:

$$x_1^{(k)}=-\frac{a_{12}}{a_{11}}x_2^{(k-1)}-\frac{a_{13}}{a_{11}}x_3^{(k-1)}+\frac{b_1}{a_{11}}$$

$$x_2^{(k)}=-\frac{a_{21}}{a_{22}}x_1^{(k-1)}-\frac{a_{23}}{a_{22}}x_3^{(k-1)}+\frac{b_2}{a_{22}}$$

$$x_3^{(k)}=-\frac{a_{31}}{a_{33}}x_1^{(k-1)}-\frac{a_{32}}{a_{33}}x_2^{(k-1)}+\frac{b_3}{a_{33}}$$

Gauss Iteration:

$$x_1^{(k)}=-\frac{a_{12}}{a_{11}}x_2^{(k-1)}-\frac{a_{13}}{a_{11}}x_3^{(k-1)}+\frac{b_1}{a_{11}}$$

$$x_2^{(k)}=-\frac{a_{21}}{a_{22}}x_1^{(k)}-\frac{a_{23}}{a_{22}}x_3^{(k-1)}+\frac{b_2}{a_{22}}$$

$$x_3^{(k)}=-\frac{a_{31}}{a_{33}}x_1^{(k)}-\frac{a_{32}}{a_{33}}x_2^{(k-1)}+\frac{b_3}{a_{33}}$$

SOR Iteration:

$$x_1^{(new)}=(1-\omega)x_1^{(old)}+\frac{\omega}{a_{11}}(b_1-a_{12}x_2^{(old)}-a_{13}x_3^{(old)})$$

$$x_2^{(new)}=(1-\omega)x_2^{(old)}+\frac{\omega}{a_{22}}(b_2-a_{21}x_1^{(new)}-a_{23}x_3^{(old)})$$

$$x_3^{(new)}=(1-\omega)x_3^{(old)}+\frac{\omega}{a_{33}}(b_3-a_{31}x_1^{(new)}-a_{32}x_2^{(new)})$$

Jacobi Iterative Technique

- Consider the following set of equations.

$$
\begin{aligned}
10x_1 &- x_2 &+ 2x_3 & &= 6\\
-x_1 &+ 11x_2 &- x_3 &+ 3x_4 &= 25\\
2x_1 &- x_2 &+ 10x_3 &- x_4 &= -11\\
 & 3x_2 &- x_3 &+ 8x_4 &= 15
\end{aligned}
$$

- Convert the set Ax = b in the form of

  x = Cx + d.

$$
\begin{aligned}
x_1 &= & &\frac{1}{10}x_2 &-\frac{1}{5}x_3 & &+\frac{3}{5}\\
x_2 &= &\frac{1}{11}x_1 & &+\frac{1}{11}x_3 &-\frac{3}{11}x_4 &+\frac{25}{11}\\
x_3 &= &-\frac{1}{5}x_1 &+\frac{1}{10}x_2 & &+\frac{1}{10}x_4 &-\frac{11}{10}\\
x_4 &= & &-\frac{3}{8}x_2 &+\frac{1}{8}x_3 & &+\frac{15}{8}
\end{aligned}
$$

Jacobi Iterative Technique Contd..

- **Start with an initial approximation of:** $x_1^{(0)} = 0,\ x_2^{(0)} = 0,\ x_3^{(0)} = 0 \text{ and } x_4^{(0)} = 0.$

$$
\begin{aligned}
x_1^{(1)} &= \frac{1}{10}x_2^{(0)} - \frac{1}{5}x_3^{(0)} + \frac{3}{5}\\
x_2^{(1)} &= \frac{1}{11}x_1^{(0)} + \frac{1}{11}x_3^{(0)} - \frac{3}{11}x_4^{(0)} + \frac{25}{11}\\
x_3^{(1)} &= -\frac{1}{5}x_1^{(0)} + \frac{1}{10}x_2^{(0)} + \frac{1}{10}x_4^{(0)} - \frac{11}{10}\\
x_4^{(1)} &= -\frac{3}{8}x_2^{(0)} + \frac{1}{8}x_3^{(0)} + \frac{15}{8}
\end{aligned}
$$

$$
\begin{aligned}
x_1^{(1)} &= \frac{1}{10}(0) - \frac{1}{5}(0) + \frac{3}{5}\\
x_2^{(1)} &= \frac{1}{11}(0) + \frac{1}{11}(0) - \frac{3}{11}(0) + \frac{25}{11}\\
x_3^{(1)} &= -\frac{1}{5}(0) + \frac{1}{10}(0) + \frac{1}{10}(0) - \frac{11}{10}\\
x_4^{(1)} &= -\frac{3}{8}(0) + \frac{1}{8}(0) + \frac{15}{8}
\end{aligned}
$$

$$x_1^{(1)} = 0.6000,\ x_2^{(1)} = 2.2727,$$
$$x_3^{(1)} = -1.1000 \text{ and } x_4^{(1)} = 1.8750$$

$$
\begin{aligned}
x_1^{(2)} &= \frac{1}{10}x_2^{(1)} - \frac{1}{5}x_3^{(1)} + \frac{3}{5}\\
x_2^{(2)} &= \frac{1}{11}x_1^{(1)} + \frac{1}{11}x_3^{(1)} - \frac{3}{11}x_4^{(1)} + \frac{25}{11}\\
x_3^{(2)} &= -\frac{1}{5}x_1^{(1)} + \frac{1}{10}x_2^{(1)} + \frac{1}{10}x_4^{(1)} - \frac{11}{10}\\
x_4^{(2)} &= -\frac{3}{8}x_2^{(1)} + \frac{1}{8}x_3^{(1)} + \frac{15}{8}
\end{aligned}
$$

Jacobi Iterative Technique Contd..

- **In the generalized way:**

$$
\begin{aligned}
x_1^{(k)} &= \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} + \frac{3}{5} \\
x_2^{(k)} &= \frac{1}{11}x_1^{(k-1)} + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11} \\
x_3^{(k)} &= -\frac{1}{5}x_1^{(k-1)} + \frac{1}{10}x_2^{(k-1)} + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10} \\
x_4^{(k)} &= -\frac{3}{8}x_2^{(k-1)} + \frac{1}{8}x_3^{(k-1)} + \frac{15}{8}
\end{aligned}
$$

Results of Jacobi Iteration:

k	0	1	2	3
$x_1^{(k)}$	**0.0000**	**0.6000**	**1.0473**	**0.9326**
$x_2^{(k)}$	**0.0000**	**2.2727**	**1.7159**	**2.0530**
$x_3^{(k)}$	**0.0000**	**-1.1000**	**-0.8052**	**-1.0493**
$x_4^{(k)}$	**0.0000**	**1.8750**	**0.8852**	**1.1309**

MATLAB© Script

Simplest MATLAB Program for Jacobi Iterative Method

```
clear all
clc
A = [2 -1 1;1 2 -1;1 -1 2];
b=[-1;6;-3];
x0=[0;0;0];
max=1000;
tol=1e-4;
[n,m]=sfize(A);
xold=x0;
C=-A;
for i=1:n
    C(i,i)=0;
end
for i=1:n
    C(i,:)=C(i,:)/A(i,i);
end
for i=1:n
    d(i,1)=b(i)/A(i,i);
end
i=1;
while i<=max
    xnew = C*xold+d;
    if norm(xnew-xold) <= tol
        x = xnew;
        disp('Jacobi method converge');
        disp([i   xnew'])
        return
    else
        xold = xnew;
    end
    i = i + 1;
end
x = xnew;
disp('Method did not Converge')
```

Gauss-Siedel Technique

- **Consider the following set of equations.**

$$\begin{aligned}
10x_1 - x_2 + 2x_3 &= 6 \\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\
2x_1 - x_2 + 10x_3 - x_4 &= -11 \\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}$$

- **Convert the set Ax = b in the form of x = Cx + d.**

$$\begin{aligned}
x_1 &= \frac{1}{10}x_2 - \frac{1}{5}x_3 + \frac{3}{5} \\
x_2 &= \frac{1}{11}x_1 + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11} \\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 + \frac{1}{10}x_4 - \frac{11}{10} \\
x_4 &= -\frac{3}{8}x_2 + \frac{1}{8}x_3 + \frac{15}{8}
\end{aligned}$$

Gauss-Siedel Technique Contd..

- **Start with an initial approximation of:**

$$x_1^{(0)} = 0,\ x_2^{(0)} = 0,\ x_3^{(0)} = 0 \text{ and } x_4^{(0)} = 0.$$

$$\begin{aligned}
x_1^{(1)} &= \qquad\quad \frac{1}{10}x_2^{(0)} - \frac{1}{5}x_3^{(0)} \qquad\quad + \frac{3}{5}\\
x_2^{(1)} &= \frac{1}{11}x_1^{(0)} \qquad\quad + \frac{1}{11}x_3^{(0)} - \frac{3}{11}x_4^{(0)} + \frac{25}{11}\\
x_3^{(1)} &= -\frac{1}{5}x_1^{(0)} + \frac{1}{10}x_2^{(0)} \qquad\quad + \frac{1}{10}x_4^{(0)} - \frac{11}{10}\\
x_4^{(1)} &= \qquad\quad -\frac{3}{8}x_2^{(0)} + \frac{1}{8}x_3^{(0)} \qquad\quad + \frac{15}{8}
\end{aligned}$$

$$\begin{aligned}
x_1^{(1)} &= \qquad\quad \frac{1}{10}x_2^{(0)} - \frac{1}{5}x_3^{(0)} \qquad\quad + \frac{3}{5}\\
x_2^{(1)} &= \frac{1}{11}x_1^{(1)} \qquad\quad + \frac{1}{11}x_3^{(0)} - \frac{3}{11}x_4^{(0)} + \frac{25}{11}\\
x_3^{(1)} &= -\frac{1}{5}x_1^{(1)} + \frac{1}{10}x_2^{(1)} \qquad\quad + \frac{1}{10}x_4^{(0)} - \frac{11}{10}\\
x_4^{(1)} &= \qquad\quad -\frac{3}{8}x_2^{(1)} + \frac{1}{8}x_3^{(1)} \qquad\quad + \frac{15}{8}
\end{aligned}$$

$$\begin{aligned}
x_1^{(k)} &= \qquad\quad \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} \qquad\quad + \frac{3}{5}\\
x_2^{(k)} &= \frac{1}{11}x_1^{(k)} \qquad\quad + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11}\\
x_3^{(k)} &= -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} \qquad\quad + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10}\\
x_4^{(k)} &= \qquad\quad -\frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} \qquad\quad + \frac{15}{8}
\end{aligned}$$

Gauss-Siedel Technique Contd..

Results of Gauss-Seidel Iteration:

(Blue numbers are for Jacobi iterations.)

The solution is: $x_1 = 1$, $x_2 = 2$, $x_3 = -1$, $x_4 = 1$

It required 15 iterations for Jacobi method and 7 iterations for Gauss-Seidel method to arrive at the solution with a tolerance of 0.00001.

k	0	1	2	3
$x_1^{(k)}$	0.0000	0.6000 0.6000	1.0300 1.0473	1.0065 0.9326
$x_2^{(k)}$	0.0000	2.3272 2.2727	2.0370 1.7159	2.0036 2.0530
$x_3^{(k)}$	0.0000	-0.9873 -1.1000	-1.0140 -0.8052	-1.0025 -1.0493
$x_4^{(k)}$	0.0000	0.8789 1.8750	0.9844 0.8852	0.9983 1.1309

MATLAB© Script

Simplest MATLAB Program for Gauss-Siedel Iterative Method

```
clear all
clc
A = [2 -1 1;1 2 -1;1 -1 2];
b=[-1;6;-3];
x0=[0;0;0];
max=1000;
tol=1e-4;
[n,m]=size(A);
x=x0;
C=-A;
for i=1:n
    C(i,i)=0;
end
for i=1:n
    C(i,:)=C(i,:)/A(i,i);
end
for i=1:n
    d(i,1)=b(i)/A(i,i);
end
i=1;
while i<=max
    xold=x;
    for j=1:n
        x(j) = C(j,:) * x + d(j);
    end
 if norm(x-xold) <= tol
disp(‘Gauss-Siedel method converge');
disp([i   x'])
        return
end
   i = i + 1;
end
disp('Method did not Converge')
```

Convergence

For any iterative numerical technique, each successive iteration results in a solution that moves progressively closer to the true solution. This is known as **convergence**.

$$\left| \frac{\max \left| \left(x_{new} - x_{old} \right) \right|}{x_{old}} \right| \le \varepsilon \qquad \left| \frac{\max \left| \left(x_{new} - x_{old} \right) \right|}{\sqrt{\sum \left(x_{new} - x_{old} \right)^2}} \right| \le \varepsilon$$

Jacobi

- For each row, magnitude of diagonal element of A should be greater than the sum of the magnitudes of the other elements in the row.
- Magnitude of the largest eigenvalue of the iterative matrix C should be less than 1.

Convergence Contd…

Gauss-Siedel

- If matrix A is real and symmetric, with positive diagonal elements, Gauss-Siedel will converge for any initial guess value; provided all eigenvalue of A are real and positive.
- Although the more rapid convergence of Gauss-Siedel is an advantage; Jacobi can be used in parallel computation.

SOR Iteration

- Relaxation (weighting) factor : ω
- Gauss-Siedel Method: $\omega = 1$
- Over relaxation: $1 < \omega < 2$
- Under relaxation: $0 < \omega < 1$

Relaxation Iteration:

$$\boxed{x_i^{new} = (1-\omega)x_i^{old} + \omega x_i^{new}}$$

SOR Iteration

GS Method $x_2^{new} = \left(b_2 - a_{21}x_1^{new} - a_{23}x_3^{old} - a_{24}x_4^{old}\right)/a_{22}$

SOR Method $x_2^{new} = (1-\omega)x_2^{old} + \omega x_2^{new}$

$$x_2^{new} = (1-\omega)x_2^{old} + \omega\left(b_2 - a_{21}x_1^{new} - a_{23}x_3^{old} - a_{24}x_4^{old}\right)/a_{22}$$

$$\Downarrow$$

$$x_1^{(new)} = (1-\omega)x_1^{(old)} + \omega(b_1 - a_{12}x_2^{(old)} - a_{13}x_3^{(old)} - a_{14}x_4^{(old)})/a_{11}$$

$$x_2^{(new)} = (1-\omega)x_2^{(old)} + \omega(b_2 - a_{21}x_1^{(new)} - a_{23}x_3^{(old)} - a_{24}x_4^{(old)})/a_{22}$$

$$x_3^{(new)} = (1-\omega)x_3^{(old)} + \omega(b_3 - a_{31}x_1^{(new)} - a_{32}x_2^{(new)} - a_{34}x_4^{(old)})/a_{33}$$

$$x_4^{(new)} = (1-\omega)x_4^{(old)} + \omega(b_4 - a_{41}x_1^{(new)} - a_{42}x_2^{(new)} - a_{43}x_3^{(new)})/a_{44}$$

SOR Iteration Example

- **Consider the following set of equations.**

$$\begin{aligned} 4x_1 \quad - x_2 \quad - x_3 &= -2 \\ 6x_1 \quad + 8x_2 \quad &= 45 \\ -5x_1 \quad + 12x_3 &= 80 \end{aligned}$$

$$\begin{cases} x_1 = (x_2 + x_3 - 2)/4 \\ x_2 = (45 - 6x_1)/8 \\ x_3 = (80 + 5x_1)/12 \end{cases}$$

Assume $x_1 = x_2 = x_3 = 0$, and $\omega = 1.2$

$$x_i = \omega\, x_i^{G-S} + (1-\omega)\, x_i^{old}; \;\; \text{G-S: Gauss-Seidel}$$

First iteration

$$\begin{cases} x_1 = (-0.2)\times 0 + 1.2\times(0+0-2)/4 = -0.6 \\ x_2 = (-0.2)\times 0 + 1.2\times\left[45 - 6\times(-0.6)\right]/8 = 7.29 \\ x_3 = (-0.2)\times 0 + 1.2\times\left[80 + 5\times(-0.6)\right]/12 = 7.7 \end{cases}$$

SOR Iteration Example Contd..

Second iteration

$$\begin{cases} x_1 = (-0.2)\times(-0.6) + 1.2\times(7.29 + 7.7 - 2)/4 = 4.017 \\ x_2 = (-0.2)\times(7.29) + 1.2\times(45 - 6(4.017))/8 = 1.6767 \\ x_3 = (-0.2)\times(7.7) + 1.2\times(80 + 5(4.017))/12 = 8.4685 \end{cases}$$

Third iteration

$$\begin{cases} x_1 = (-0.2)\times(4.017) + 1.2\times(1.6767 + 8.4685 - 2)/4 = 1.6402 \\ x_2 = (-0.2)\times(1.6767) + 1.2\times(45 - 6(1.6402))/8 = 4.9385 \\ x_3 = (-0.2)\times(8.4685) + 1.2\times(80 + 5(1.6402))/12 = 7.1264 \end{cases}$$

To achieve fast convergence, relaxation parameter (ω) should be optimum..

SOR Iteration Example Contd..

Result obtained at each iteration for ω=1.2

```
» [A,b]=Example

A =
 4     -1     -1
 6      8      0
-5      0     12

b =
    -2
    45
    80

» x0=[0 0 0]'
x0 =
     0
     0
     0

» tol=1.e-6
tol =
  1.0000e-006
```

```
» w=1.2; x =
        i         x1        x2        x3      ....
   1.0000    -0.6000    7.2900    7.7000
   2.0000     4.0170    1.6767    8.4685
   3.0000     1.6402    4.9385    7.1264
   4.0000     2.6914    3.3400    7.9204
   5.0000     2.2398    4.0661    7.5358
   6.0000     2.4326    3.7474    7.7091
   7.0000     2.3504    3.8851    7.6334
   8.0000     2.3855    3.8261    7.6661
   9.0000     2.3705    3.8513    7.6521
  10.0000     2.3769    3.8405    7.6580
  11.0000     2.3742    3.8451    7.6555
  12.0000     2.3753    3.8432    7.6566
  13.0000     2.3749    3.8440    7.6561
  14.0000     2.3751    3.8436    7.6563
  15.0000     2.3750    3.8438    7.6562
  16.0000     2.3750    3.8437    7.6563
  17.0000     2.3750    3.8438    7.6562
  18.0000     2.3750    3.8437    7.6563
  19.0000     2.3750    3.8438    7.6562
  20.0000     2.3750    3.8437    7.6563
  21.0000     2.3750    3.8438    7.6562

SOR method converged
```

SOR Iteration Example Contd..

Result obtained at each iteration for ω=1.5

```
» [A,b]=Example

A =
 4      -1      -1
 6       8       0
-5       0      12

b =
    -2
    45
    80

» x0=[0 0 0]'
x0 =
     0
     0
     0

» tol=1.e-6
tol =
  1.0000e-006
```

```
» w = 1.5; x =
          i        x1        x2        x3    ....
     1.0000   -0.7500    9.2812    9.5312
     2.0000    6.6797   -3.7178    9.4092
     3.0000   -1.9556   12.4964    4.0732
     4.0000    6.4414   -5.0572   11.9893
     5.0000   -1.3712   12.5087    3.1484
     6.0000    5.8070   -4.3497   12.0552
     7.0000   -0.7639   11.4718    3.4949
     8.0000    5.2445   -3.1985   11.5303
     9.0000   -0.2478   10.3155    4.0800
    10.0000    4.7722   -2.0890   10.9426
    11.0000    0.1840    9.2750    4.6437
    12.0000    4.3775   -1.1246   10.4141
    13.0000    0.5448    8.3869    5.1335
    14.0000    4.0477   -0.3097    9.9631
    15.0000    0.8462    7.6404    5.5473
     .........
    20.0000    3.3500    1.4220    9.0016
     .........
    30.0000    2.7716    2.8587    8.2035
     .........
    50.0000    2.4406    3.6808    7.7468
     .........
   100.0000    2.3757    3.8419    7.6573

SOR method did not converge
```

Diverged !!

MATLAB© Script

Simplest MATLAB Program for SOR Iterative Method

```
clear all
clc
A = [2 -1 1;1 2 -1;1 -1 2];
b=[-1;6;-3];
x0=[0;0;0];
max=1000;
tol=1e-4;
w = 1.2;
[n,m]=size(A);
xold=x0;
C=-A;
for i=1:n
    C(i,i)=0;
end
for i=1:n
    C(i,:)=C(i,:)/A(i,i);
end
for i=1:n
    d(i,1)=b(i)/A(i,i);
end
i=1;
while i<=max
    xold = x;
    for j=1:n
        x(j) = (1-w)*xold(j) + w*(C(j,:)*x+d(j));
    end
if norm(xold-x) <= tol
        disp('SOR method Converge');
        return
end
disp([i    x'])
i = i + 1;
end
disp('SOR did not Converge')
```

Convergence

- If $0<\omega<2$, SOR method converges for any initial vector if A matrix is symmetric and positive definite.
- If $\omega>2$, SOR method diverges.
- If $0<\omega<1$, SOR method converges but the convergence rate is slower (deceleration) than the Gauss-Seidel method.

Computation Cost

- The operation count for Gaussian Elimination or LU Decomposition was $O(n^3)$.
- For iterative methods, the number of scalar multiplications is $O(n^2)$ at each iteration.
- Iterative methods are well suited for sparse matrices.

THANK YOU

Questions??