NUMERICAL METHODS

$$U^{n+1} = U^n + \Delta t\, f(U^n)$$

$$\frac{\partial v}{\partial t} + V \cdot \nabla v = \nabla \cdot (k \nabla v) + g(v)$$

$$(\lambda + \mu)\nabla(\nabla \cdot u) + \mu \nabla^2 u = \alpha(3\lambda + 2\mu)\nabla T - \rho b$$

Lecture 5

$$\rho\left(\frac{\partial u}{\partial t} + V \cdot \nabla u\right) = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x$$

$$\nabla^2 u = f$$

Dr. P V Ramana

JACOBI ITERATION Method: Example 2

Given the system of equations

$$12x_1 + 3x_2 - 5x_3 = 1$$

$$x_1 + 5x_2 + 3x_3 = 28$$

$$3x_1 + 7x_2 + 13x_3 = 76$$

With an initial guess of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

The coefficient matrix is:

$$[A] = \begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix}$$

Will the solution converge using the **JACOBI ITERATION** method?

JACOBI ITERATION Method : Example 2

Checking if the coefficient matrix is diagonally dominant

$$[A]=\begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix}$$

$$|a_{11}| = |12| = 12 \geq |a_{12}| + |a_{13}| = |3| + |-5| = 8$$

$$|a_{22}| = |5| = 5 \geq |a_{21}| + |a_{23}| = |1| + |3| = 4$$

$$|a_{33}| = |13| = 13 \geq |a_{31}| + |a_{32}| = |3| + |7| = 10$$

The inequalities are all true and at least one row is *strictly* greater than:

Therefore: The solution should converge using the **JACOBI ITERATION** Method

JACOBI ITERATION Method : Example 2

Rewriting each equation

$$\begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 28 \\ 76 \end{bmatrix}$$

$$x_1 = \frac{1 - 3x_2 + 5x_3}{12}$$

$$x_2 = \frac{28 - x_1 - 3x_3}{5}$$

$$x_3 = \frac{76 - 3x_1 - 7x_2}{13}$$

With an initial guess of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$x_1 = \frac{1 - 3(0) + 5(1)}{12} = 0.50000$$

$$x_2 = \frac{28 - (0.5) - 3(1)}{5} = 4.9000$$

$$x_3 = \frac{76 - 3(0.50000) - 7(4.9000)}{13} = 3.0923$$

JACOBI ITERATION Method : Example 2

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$\mathrm{x}_1 = 0.50000$

$\mathrm{x}_2 = 4.9000$

$\mathrm{x}_3 = 3.0923$

The absolute relative approximate error

$$\left|\in_a\right|_1 = \left|\frac{0.50000 - 1.0000}{0.50000}\right| \times 100 = 67.662\%$$

$$\left|\in_a\right|_2 = \left|\frac{4.9000 - 0}{4.9000}\right| \times 100 = 100.00\%$$

$$\left|\in_a\right|_3 = \left|\frac{3.0923 - 1.0000}{3.0923}\right| \times 100 = 67.662\%$$

The maximum absolute relative error after the first iteration is 100%

JACOBI ITERATION Method : Example 2

After Iteration #1

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.5000 \\ 4.9000 \\ 3.0923 \end{bmatrix}$$

Substituting the x values into the equations

$$x_1 = \frac{1-3(4.9000)+5(3.0923)}{12} = 0.14679$$

$$x_2 = \frac{28-(0.14679)-3(3.0923)}{5} = 3.7153$$

$$x_3 = \frac{76-3(0.14679)-7(4.900)}{13} = 3.8118$$

After Iteration #2

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.14679 \\ 3.7153 \\ 3.8118 \end{bmatrix}$$

JACOBI ITERATION Method : Example 2

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.14679 \\ 3.7153 \\ 3.8118 \end{bmatrix}$$

Iteration #2 absolute relative approximate error

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.5000 \\ 4.9000 \\ 3.0923 \end{bmatrix}$$

$$\left|\in_a\right|_1 = \left|\frac{0.14679 - 0.50000}{0.14679}\right| \times 100 = 240.62\%$$

$$\left|\in_a\right|_2 = \left|\frac{3.7153 - 4.9000}{3.7153}\right| \times 100 = 31.887\%$$

$$\left|\in_a\right|_3 = \left|\frac{3.8118 - 3.0923}{3.8118}\right| \times 100 = 18.876\%$$

The maximum absolute relative error after the first iteration is 240.62%.

This is much larger than the maximum absolute relative error obtained in iteration #1. Is this a problem?

JACOBI ITERATION Method: Example 2

Repeating more iterations, the following values are obtained

Iteration	a_1	$\lvert\varepsilon_a\rvert_1$	a_2	$\lvert\varepsilon_a\rvert_2$	a_3	$\lvert\varepsilon_a\rvert_3$
1	0.50000	67.662	4.900	100.00	3.0923	67.662
2	0.14679	240.62	3.7153	31.887	3.8118	18.876
3	0.74275	80.23	3.1644	17.409	3.9708	4.0042
4	0.94675	21.547	3.0281	4.5012	3.9971	0.65798
5	0.99177	4.5394	3.0034	0.82240	4.0001	0.07499
6	0.99919	0.74260	3.0001	0.11000	4.0001	0.00000

The solution obtained

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.99919 \\ 3.0001 \\ 4.0001 \end{bmatrix}$$

is close to the exact solution of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$$

JACOBI ITERATION Method

Consider 4x4 case

$$\begin{bmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 25 \\ -11 \\ 15 \end{bmatrix}$$

Example

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

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

$$\text{given} \quad x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} x_1^{(1)} &= (x_2^{(0)} - 2x_3^{(0)} + 6)/10 \\ x_2^{(1)} &= (x_1^{(0)} + x_3^{(0)} - 3x_4^{(0)} + 25)/11 \\ x_3^{(1)} &= (-2x_1^{(0)} + x_2^{(0)} + x_4^{(0)} - 11)/10 \\ x_4^{(1)} &= (-3x_2^{(0)} + x_3^{(0)} + 15)/(-8) \end{aligned}$$

$$x^{(1)} = \begin{bmatrix} 0.6000 \\ 2.2727 \\ -1.1000 \\ 1.8750 \end{bmatrix}$$

JACOBI ITERATION Method

$$
\begin{aligned}
x_1^{(k+1)} &= (\quad x_2^{(k)} - 2x_3^{(k)} + 6)/10 \\
x_2^{(k+1)} &= (\quad x_1^{(k)} + x_3^{(k)} - 3x_4^{(k)} + 25)/11 \\
x_3^{(k+1)} &= (-2x_1^{(k)} + x_2^{(k)} + x_4^{(k)} - 11)/10 \\
x_4^{(k+1)} &= (\quad -3x_2^{(k)} + x_3^{(k)} + 15)/(-8)
\end{aligned}
$$

Note that in the Jacobi iteration one does not use the most recently available information.

$$
\begin{aligned}
x_1^{(k+1)} &= (\quad x_2^{(k)} - 2x_3^{(k)} + 6)/10 \\
x_2^{(k+1)} &= (\quad x_1^{(k+1)} + x_3^{(k)} - 3x_4^{(k)} + 25)/11 \\
x_3^{(k+1)} &= (-2x_1^{(k+1)} + x_2^{(k+1)} + x_4^{(k)} - 11)/10 \\
x_4^{(k+1)} &= (\quad -3x_2^{(k+1)} + x_3^{(k+1)} + 15)/(-8)
\end{aligned}
$$

	K=1	K=2	K=3	K=4	K=5
x1	0.6000	1.0302	1.0066	1.0009	1.0001
x2	2.2730	2.0369	2.0036	2.0003	2.0000
x3	-1.1000	-1.0145	-1.0025	-1.0003	-1.0000
X4	1.8750	0.9843	0.9984	0.9998	1.0000
$\|r^{(k)}\|$	5.6930	0.4300	0.0662	0.0082	0.0009

$$
x^* = \begin{bmatrix} 1 \\ 2 \\ -1 \\ 1 \end{bmatrix}
$$

JACOBI ITERATION Method

$$\begin{bmatrix} a_{11} & \Lambda & a_{1n} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ a_{n1} & \Lambda & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \mathrm{M} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \mathrm{M} \\ b_n \end{bmatrix}$$

Gauss-Seidel iteration for general n:

```
for i = 1 : n
    $x_i^{(k+1)} = \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}$
end
```

MATLAB CODE

Ex:

Write a Matlab function for JI

```
function [sol,X]=gs(A,b,x0)
n=length(b);
maxiter=10;
x=x0;
for k=1:maxiter
for i=1:n
    sum1=0;
    for j=1:i-1
        sum1=sum1+A(i,j)*x(j);
    end
     sum2=0;
    for j=i+1:n
        sum2=sum2+A(i,j)*x(j);
    end
    x(i)=(b(i)-sum1-sum2)/A(i,i)
end
X(1:n,k)=x;
end
sol=x;
```

JACOBI ITERATION Method

iteration for general n:

$\mathrm{for\ i} = 1 : \mathrm{n}$

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}$$

end

Iterative Methods

- ❑ **Jacobi Iteration Method**
- ❑ **Gauss – Siedel Method**

Gauss – Siedel Method

GS Iterative methods provide an **alternative** to the *elimination methods*.

$$Ax = b \qquad A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad D = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}$$

$$[D + (A - D)]x = b \quad \Rightarrow \quad Dx = b - (A - D)x \quad \Rightarrow \quad x = D^{-1}[b - (A - D)x]$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1/a_{11} & 0 & 0 \\ 0 & 1/a_{22} & 0 \\ 0 & 0 & 1/a_{33} \end{bmatrix} * \left(\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} - \begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right)$$

$$x_1^k = \frac{b_1 - a_{12}x_2^{k-1} - a_{13}x_3^{k-1}}{a_{11}} \qquad x_2^k = \frac{b_2 - a_{21}x_1^{k-1} - a_{23}x_3^{k-1}}{a_{22}} \qquad x_3^k = \frac{b_3 - a_{31}x_1^{k-1} - a_{32}x_2^{k-1}}{a_{33}}$$

Choose an initial guess (i.e. all zeros) and Iterate until the equality is satisfied.
No guarantee for convergence! Each iteration takes O(n^2) time!

Gauss – Siedel Method

- The *Gauss-Seidel* method is a commonly used *iterative method*.
- It is same as **Jacobi technique** except with one important difference:

  A newly computed *x* value (say x_k) is substituted in the subsequent equations (equations *k+1*, *k+*2, …, *n*) **in the same iteration**.

Example: Consider the *3x3* system below:

$$x_1^{new} = \frac{b_1 - a_{12}x_2^{old} - a_{13}x_3^{old}}{a_{11}}$$

$$x_2^{new} = \frac{b_2 - a_{21}x_1^{new} - a_{23}x_3^{old}}{a_{22}}$$

$$x_3^{new} = \frac{b_3 - a_{31}x_1^{new} - a_{32}x_2^{new}}{a_{33}}$$

$$\{X\}_{old} \leftarrow \{X\}_{new}$$

- First, choose initial guesses for the x’s.
- A simple way to obtain initial guesses is to assume that they are all **zero**.
- Compute **new** $\mathbf{x_1}$ using the previous iteration values.
- **New** $\boldsymbol{x_1}$ is substituted in the equations to calculate x_2 and x_3
- The process is repeated for $\mathbf{x_2}$, $\mathbf{x_3}$, …

Gauss – Siedel Method

$$\begin{aligned} a_{11}\boxed{x_1} + a_{12}x_2 + \Lambda + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}\boxed{x_2} + \Lambda + a_{2n}x_n &= b_2 \\ &\mathrm{M} \\ a_{n1}x_1 + a_{n2}x_2 + \Lambda + a_{nn}\boxed{x_n} &= b_n \end{aligned}$$

$$x^0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \mathrm{M} \\ x_n^0 \end{bmatrix}$$

$$x_1^1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2^0 - \Lambda - a_{1n}x_n^0)$$

$$x_i^{k+1} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^{n} a_{ij}x_j^k\right]$$

$$x_2^1 = \frac{1}{a_{22}}(b_2 - a_{21}x_1^0 - a_{23}x_3^0 - \Lambda - a_{2n}x_n^0)$$

$$x_n^1 = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^0 - a_{n2}x_2^0 - \Lambda - a_{nn-1}x_{n-1}^0)$$

Gauss – Siedel Method

$x^{k+1}=Ex^{k}+f$ iteration for Jacobi method

A can be written as A=L+D+U (*not decomposition*)

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

$Ax=b \Rightarrow (L+D+U)x=b$

$Dx^{k+1} =-(L+U)x^{k}+b$

$x^{k+1}=-D^{-1}(L+U)x^{k}+D^{-1}b$

$E=-D^{-1}(L+U)$

$f=D^{-1}b$

$$x_i^{k+1} = \frac{1}{a_{ii}}\left[b_i - \underbrace{\sum_{j=1}^{i-1} a_{ij}x_j^{k}}_{Lx^{k}} - \underbrace{\sum_{j=i+1}^{n} a_{ij}x_j^{k}}_{Ux^{k}} \right]$$

Dx^{k+1}

JACOBI ITERATION & GAUSS SEIDEL Method

$$x_1^{new} = \frac{b_1 - a_{12}x_2^{old} - a_{13}x_3^{old} \ldots}{a_{11}}$$

$$x_2^{new} = \frac{b_2 - a_{21}x_1^{old} - a_{23}x_3^{old} \ldots}{a_{22}}$$

$$x_3^{new} = \frac{b_3 - a_{31}x_1^{old} - a_{32}x_2^{old} \ldots}{a_{33}}$$

.

.

$$x_n^{new} = \frac{b_n - a_{n1}x_1^{old} - a_{n2}x_2^{old} \ldots.}{a_{nn}}$$

$$\{X\}_{old} \Rightarrow \{x_1, x_2 \ldots\ldots.. x_n\}_{old}$$

$$x_1^{new} = \frac{b_1 - a_{12}x_2^{old} - a_{13}x_3^{old} \ldots.}{a_{11}}$$

$$x_2^{new} = \frac{b_2 - a_{21}x_1^{new} - a_{23}x_3^{old} \ldots.}{a_{22}}$$

$$x_3^{new} = \frac{b_3 - a_{31}x_1^{new} - a_{32}x_2^{new} \ldots.}{a_{33}}$$

.

.

$$x_{nn}^{new} = \frac{b_n - a_{n1}x_1^{new} - a_{n2}x_2^{new} \ldots.}{a_{nn}}$$

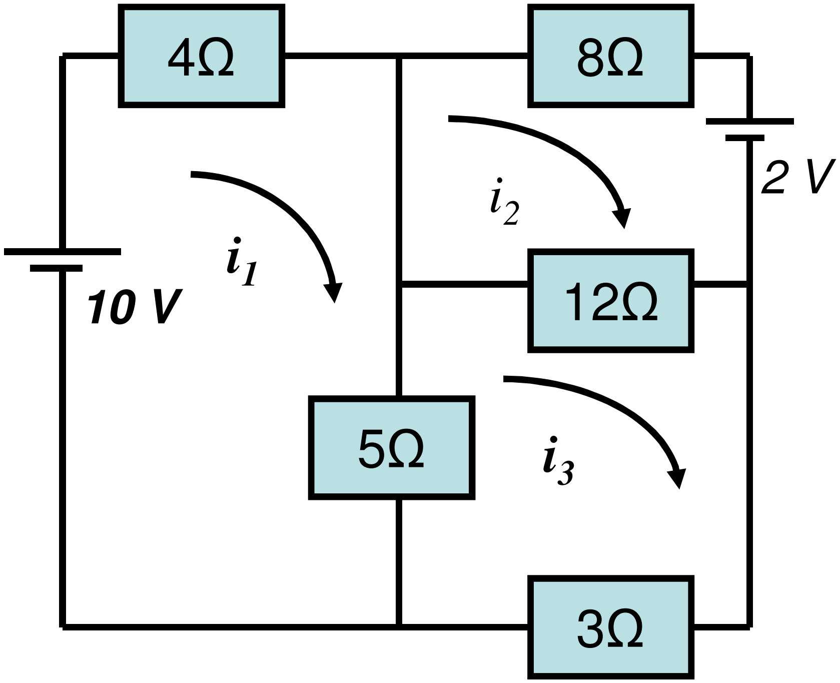
$$\{X\}_{old} \leftarrow \{X\}_{new}$$

Gauss – Siedel Method: Example 1

Consider a circuit shown in figure here; currents i_1, i_2, and i_3 are given by

$$4i_1+0i_2+5(i_1-i_3)=10$$
$$0i_1+8i_2+12(i_2-i_3)=-2$$
$$5(i_3-i_1)+12(i_3-i_2)+3i_3=0$$

$$9i_1+0i_2-5i_3=10$$
$$0i_1+20i_2-12i_3=-2$$
$$-5i_1-12i_2+20i_3=0$$



The matrix form is:

$$\begin{bmatrix} 9 & 0 & -5 \\ 0 & 20 & -12 \\ -5 & -12 & 20 \end{bmatrix}\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix}=\begin{bmatrix} 10 \\ -2 \\ 0 \end{bmatrix}$$

Notice that magnitude of any diagonal element is greater than the sum of the magnitudes of other elements in that row

A matrix with this property is said to be Diagonally dominant.

Gauss – Siedel Method: Example 1

The set of equations:

$$9i_1 + 0i_2 - 5i_3 = 10$$
$$0i_1 + 20i_2 - 12i_3 = -2$$
$$-5i_1 - 12i_2 + 20i_3 = 0$$

Let us write for i_1, i_2 and i_3 as

$$i_1 = (10 + 5i_3)/9 = 1.1111 + 0.5556i_2 \qquad (1)$$
$$i_2 = (-2 + 12i_3)/20 = -0.1000 + 0.6000i_2 \qquad (2)$$
$$i_3 = (5i_1 + 12i_3)/20 = 0.2500\, i_1 + 0.6000\, i_2 \qquad (3)$$

Let us make an initial guess as $i_1 = 0.0;$ $i_2 = 0.0$ *and* $i_3 = 0.0$

First iteration results: $i_1 = 1.1111;$ $i_2 = -0.1000$ *and* $i_3 = 0.0$

Gauss – Siedel Method: Example 1

$$i_1 = (10 + 5i_3)/9 = 1.1111 + 0.5556i_2 \qquad (1)$$

$$i_2 = (-2 + 12i_3)/20 = -0.1000 + 0.6000i_2 \qquad (2)$$

$$i_3 = (5i_1 + 12i_3)/20 = 0.2500\, i_1 + 0.6000\, i_2 \qquad (3)$$

First iteration results:	$i_1 = 1.1111;$	$i_2 = -0.1000$	*and*	$i_3 = 0.0$
2nd iteration results:	$i_1 = 1.1111;$	$i_2 = -0.1000$	*and*	$i_3 = 0.22$
3rd iteration results:	$i_1 = 1.23;$	$i_2 = 0.03$	*and*	$i_3 = 0.22$
4th iteration results:	$i_1 = 1.23\ ;$	$i_2 = 0.03$	*and*	$i_3 = 0.33$
5th iteration results:	$i_1 = 1.29;$	$i_2 = 0.1$	*and*	$i_3 = 0.33$
6th iteration results:	$i_1 = 1.29;$	$i_2 = 0.1$	*and*	$i_3 = 0.38$

Gauss – Siedel Method: Example 2

$$4X_1 + 2X_2 = 2$$

$$2X_1 + 10X_2 + 4X_3 = 6$$

$$4X_2 + 5X_3 = 5$$

Solution: $(X_1, X_2, X_3) = (0.41379, 0.17241, 0.86206)$

Gauss – Siedel Method: Example 2

$$\begin{bmatrix} 4 & 2 & 0 \\ 2 & 10 & 4 \\ 0 & 4 & 5 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 6 \\ 5 \end{Bmatrix}$$

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix} + \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 6 \\ 5 \end{Bmatrix}$$

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 6 \\ 5 \end{Bmatrix} - \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix}$$

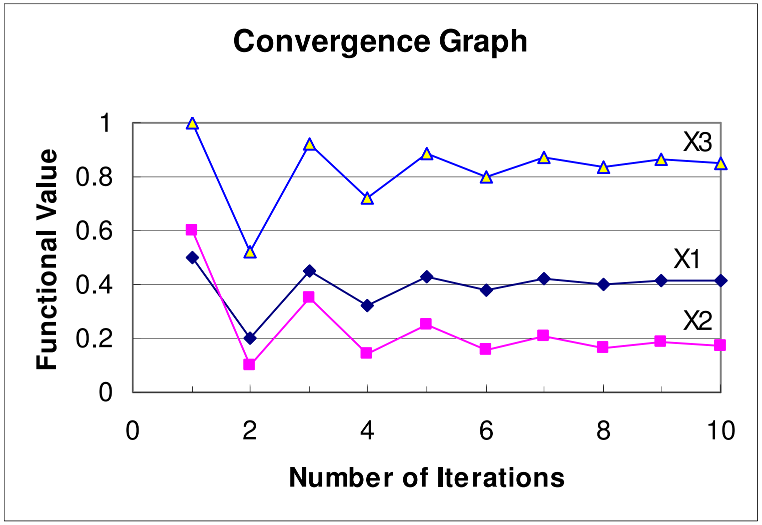
Gauss – Siedel Method: Example 2

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 6 \\ 5 \end{Bmatrix} - \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix}$$

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5 \end{bmatrix}^{-1} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$$

$$\begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \begin{Bmatrix} 2 \\ 6 \\ 5 \end{Bmatrix} - \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \end{Bmatrix}$$

Gauss – Siedel Method: Example 2



The actual Solution:

(X1 , X2 , X3) =

(0.41379, 0.17241, 0.86206)

Iteration	1	2	3	4	5	6	7
X_1	0.5	0.2	0.45	0.324	0.429	0.376	0.42
X_2	0.6	0.1	0.352	0.142	0.248	0.16	0.204
X_3	1	0.52	0.92	0.718	0.886	0.802	0.872

Gauss – Siedel Method: Example 3

Consider the following set of equations.

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

Convert the set Ax = b in the form of x = Tx + c.

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

Jacobi - Iterative Method: Example 3

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

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

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

$$
\begin{aligned}
x_1^{(1)} &= 0.6000, \\
x_2^{(1)} &= 2.2727, \\
x_3^{(1)} &= -1.1000 \\
x_4^{(1)} &= 1.8750
\end{aligned}
$$

Jacobi - Iterative Method : Example 3

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

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

Jacobi - Iterative Method : Example 3

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

Results:

iteration	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

Jacobi – Iterative Method: Example 4 (Gauss – Siedel)

A diverging case study:

$$\begin{bmatrix} -2 & 1 & 5 \\ 4 & -8 & 1 \\ 4 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15 \\ -21 \\ 7 \end{bmatrix} \qquad x^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \left\| b - Ax^0 \right\|_2 = 26.7395$$

The matrix is not diagonally dominant

$$\begin{aligned} x_1^1 &= \frac{-15 + x_2^0 + 5x_3^0}{2} &&= \frac{-15}{2} = -7.5 \\ x_2^1 &= \frac{21 + 4x_1^0 + x_3^0}{8} &&= \frac{21}{8} = 2.625 \\ x_3^1 &= 7 - 4x_1^0 + x_2^0 &&= 7.0 \end{aligned}$$

$$\left\| b - Ax^1 \right\|_2 = 54.8546$$

Gauss – Siedel Method: Example 4

$$x_1^1 = \frac{-15 + 2.625 + 5\times 7}{2} = 11.3125$$

$$x_2^1 = \frac{21 - 4\times 7.5 + 7}{8} = -0.25$$

$$x_3^1 = 7 + 4\times 7.5 + 2.625 = 39.625$$

$$\left\| b - Ax^2 \right\|_2 = 208.3761$$

The residual term is increasing at each iteration, so the iterations are *diverging*.

Note that the matrix is <u>not diagonally dominant</u>

Pseudo-Code For GS Method

```
1) build   A, b
2) build modified A with diagonal zero → Q
3) set initial guess x=0

4) do {
     a) compute:        $\tilde{x}_i = \frac{1}{\mathbf{A}_{ii}}\left( b_i - \sum_{j=1}^{j=N} \mathbf{Q}_{ij} x_j \right)$

   b) compute error:    $err = \frac{\max_{i=1,..,N}\left(\left|x_i - \tilde{x}_i\right|\right)}{\max_{i=1,..,N}\left(\left|b_i\right|\right)}$

   c) update x:         $x_i = \tilde{x}_i$
     }while err>tol
```

Solve $6x_1 - 2x_2 + x_3 = 11$ (1)

$x_1 + 2x_2 - 5x_3 = -1$ (2)

$-2x_1 + 7x_2 + 2x_3 = 5$ (3)

$6x_1 - 2x_2 + x_3 = 11$ (1)

$-2x_1 + 7x_2 + 2x_3 = 5$ (2)

$x_1 + 2x_2 - 5x_3 = -1$ (3)

Step 1:

Re-write the equations such that each equation has the unknown with largest coefficient on the left hand side:

$$x_1 = \frac{2x_2 - x_3 + 11}{6}$$

from eq. (1)

$$x_2 = \frac{2x_1 - 2x_3 + 5}{7}$$

from eq. (3)

$$x_3 = \frac{x_1 + 2x_2 + 1}{5}$$

from eq. (2)

Step 2:

Assume the initial guesses $(x_2)^0 = (x_3)^0 = 0$, then calculate $(x_1)^1$:

$$(x_1)^1 = \frac{2(x_2)^0 - (x_3)^0 + 11}{6} = \frac{2(0) - (0) + 11}{6} = 1.833$$

Step 2a: Use the updated value $(x_1)^1 = 1.833$ and $(x_3)^0 = 0$ to calculate $(x_2)^1$

$$(x_2)^1 = \frac{2(x_1)^1 - 2(x_3)^0 + 5}{7} = \frac{2(1.833) - 2(0) + 5}{7} = 1.238$$

Step 2b: Similarly, use $(x_1)^1 = 1.833$ and $(x_2)^1 = 1.238$ to calculate $(x_3)^1$

$$(x_3)^1 = \frac{(x_1)^1 + 2(x_2)^1 + 1}{5} = \frac{(1.833) + 2(1.238) + 1}{5} = 1.062$$

$6x_1 - 2x_2 + x_3 = 11$ (1)

$-2x_1 + 7x_2 + 2x_3 = 5$ (2)

$x_1 + 2x_2 - 5x_3 = -1$ (3)

Step 3:

Repeat the same procedure for the 2nd iteration

$$(x_1)^2 = \frac{2(x_2)^1 - (x_3)^1 + 11}{6} = \frac{2(1.238) - (1.062) + 11}{6} = 2.069$$

$$(x_2)^2 = \frac{2(x_1)^2 - 2(x_3)^1 + 5}{7} = \frac{2(2.069) - 2(1.062) + 5}{7} = 1.002$$

$$(x_3)^2 = \frac{(x_1)^2 + 2(x_2)^2 + 1}{5} = \frac{(2.069) + 2(1.002) + 1}{5} = 1.015$$

Step 4:

the next iterations so that the next values are calculated using the current values:

$$(x_1)^{i+1} = \frac{2(x_2)^i - (x_3)^i + 11}{6}$$

$$(x_2)^{i+1} = \frac{2(x_1)^i - 2(x_3)^i + 5}{7}$$

$$(x_3)^{i+1} = \frac{(x_1)^i + 2(x_2)^i + 1}{5}$$

continue the above iterative procedure until $[(x_k)^{i+1} - (x_k)^{I}]/ (x_k)^{i+1} < \epsilon_s$ for k=1,2 and 3.

Unknown →	x_1	x_2	x_3
↓ Iteration			
1	1.833	1.238	1.062
2	2.069	1.002	1.015
3	1.998	0.995	0.998
4	1.999	1.000	1.000
5	2.000	1.000	1.000

JACOBI / Gauss – Siedel Method: Example 2

$$\begin{bmatrix} 4 & -1 & 1 \\ 4 & -8 & 1 \\ -2 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ -21 \\ 15 \end{bmatrix}$$

$$x^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\| b - Ax^0 \right\|_2 = 26.7395$$

Diagonally dominant matrix

$$x_1^1 = \frac{7 + x_2^0 - x_3^0}{4} = \frac{7}{4} = 1.75$$

$$x_2^1 = \frac{21 + 4x_1^0 + x_3^0}{8} = \frac{21}{8} = 2.625$$

$$x_3^1 = \frac{15 + 2x_1^0 - x_2^0}{5} = \frac{15}{5} = 3.0$$

$$\left\| b - Ax^1 \right\|_2 = 10.0452$$

$$x_1^1 = \frac{7 + x_2^0 - x_3^0}{4} = \frac{7}{4} = 1.75$$

$$x_2^1 = \frac{21 + 4x_1^1 + x_3^0}{8} = \frac{28}{8} = 3.5$$

$$x_3^1 = \frac{15 + 2x_1^1 - x_2^1}{5} = \frac{15}{5} = 3.0$$

JACOBI / Gauss – Siedel Method: Example 2

$$x_1^2 = \frac{7 + x_2^1 - x_3^1}{4}$$
$$x_2^2 = \frac{21 + 4x_1^1 + x_3^1}{8}$$
$$x_3^2 = \frac{15 + 2x_1^1 - x_2^1}{5}$$

$$= \frac{7 + 2.625 - 3}{4} = 1.65625$$
$$= \frac{21 + 4 \times 1.75 + 3}{8} = 3.875$$
$$= \frac{15 + 2 \times 1.75 - 2.625}{5} = 4.225$$

$$\left\| b - Ax^2 \right\|_2 = 6.7413$$

$$x_1^3 = \frac{7 + 3.875 - 4.225}{4} = 1.6625$$
$$x_2^3 = \frac{21 + 4 \times 1.65625 + 4.225}{8} = 3.98125$$
$$x_3^3 = \frac{15 + 2 \times 1.65625 - 3.875}{5} = 2.8875$$

$$\left\| b - Ax^2 \right\|_2 = 1.9534$$

Matrix is diagonally dominant, Jacobi iterations are *converging*

Jacobi – Iterative Method: Example 4

Gauss – Siedel

Consider 4x4 case

$$\begin{bmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 25 \\ -11 \\ 15 \end{bmatrix}$$

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

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

$$\text{given} \quad x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} x_1^{(1)} &= (x_2^{(0)} - 2x_3^{(0)} + 6)/10 \\ x_2^{(1)} &= (x_1^{(0)} + x_3^{(0)} - 3x_4^{(0)} + 25)/11 \\ x_3^{(1)} &= (-2x_1^{(0)} + x_2^{(0)} + x_4^{(0)} - 11)/10 \\ x_4^{(1)} &= (-3x_2^{(0)} + x_3^{(0)} + 15)/(-8) \end{aligned}$$

$$x^{(1)} = \begin{bmatrix} 0.6000 \\ 2.2727 \\ -1.1000 \\ 1.8750 \end{bmatrix}$$

Gauss – Siedel Method: Example 4

$$\begin{aligned}
x_1^{(k+1)} &= (\qquad\qquad x_2^{(k)} - 2x_3^{(k)} \qquad\qquad + 6)/10 \\
x_2^{(k+1)} &= (\quad x_1^{(k)} \qquad\qquad + x_3^{(k)} - 3x_4^{(k)} + 25)/11 \\
x_3^{(k+1)} &= (-2x_1^{(k)} + x_2^{(k)} \qquad\qquad + x_4^{(k)} - 11)/10 \\
x_4^{(k+1)} &= (\qquad - 3x_2^{(k)} + x_3^{(k)} \qquad\qquad + 15)/(-8)
\end{aligned}$$

	K=1	K=2	K=3	K=4	K=5
x1	0.6000	1.0473	0.9326	1.0152	0.9890
x2	2.2727	1.7159	2.0533	1.9537	2.0114
x3	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103
X4	1.8750	0.8852	1.1309	0.9738	1.0214
$\lVert r^{(k)} \rVert$	11.3537	4.9910	2.0299	0.8911	0.3686

	K=6	K=7	K=8	K=9	K=10
x1	1.0032	0.9981	1.0006	0.9997	1.0001
x2	1.9922	2.0023	1.9987	2.0004	1.9998
x3	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
X4	0.9944	1.0036	0.9989	1.0006	0.9998
$\lVert r^{(k)} \rVert$	0.1605	0.0671	0.0290	0.0122	0.0053

$$x^* = \begin{bmatrix} 1 \\ 2 \\ -1 \\ 1 \end{bmatrix}$$

MATLAB CODE

Ex:
Write a Matlab function for Jacobi

```
function [sol,X]=jacobi(A,b,x0)
n=length(b);
maxiter=10;
x=x0;
for k=1:maxiter
for i=1:n
    sum1=0;
    for j=1:i-1
        sum1=sum1+A(i,j)*x(j);
    end
     sum2=0;
    for j=i+1:n
        sum2=sum2+A(i,j)*x(j);
    end
    xnew(i)=(b(i)-sum1-sum2)/A(i,i)
end
X(1:n,k)=xnew;
x=xnew;
end
sol=xnew;
```

GS for general n:

$\text{for}\ \mathrm{i} = 1:\mathrm{n}$

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}$$

end

Convergence for **Gauss – Siedel Method**

E=-D^{-1}(L+U)

$$E=\begin{bmatrix} 0 & -\dfrac{a_{12}}{a_{11}} & \Lambda & \Lambda & -\dfrac{a_{1n}}{a_{11}} \\ -\dfrac{a_{21}}{a_{22}} & 0 & -\dfrac{a_{23}}{a_{22}} & \Lambda & -\dfrac{a_{2n}}{a_{22}} \\ \mathrm{M} & \mathrm{O} & \mathrm{O} & \mathrm{O} & \mathrm{M} \\ & & \mathrm{O} & \mathrm{O} & -\dfrac{a_{n-1n}}{a_{n-1n-1}} \\ -\dfrac{a_{n1}}{a_{nn}} & \Lambda & \Lambda & -\dfrac{a_{nn-1}}{a_{nn}} & 0 \end{bmatrix}$$

Convergence for **Gauss – Siedel Method**

Evaluate the infinity(maximum row sum) norm of E

$$\left\|E\right\|_\infty < 1 \Rightarrow \sum_{\substack{j=1 \\ i\neq j}}^{n} \frac{\left|a_{ij}\right|}{\left|a_{ii}\right|} < 1 \quad \text{for i} = 1,2,\ldots,\text{n}$$

$$\Rightarrow \left|a_{ii}\right| > \sum_{\substack{j=1 \\ i\neq j}}^{n} \left|a_{ij}\right|$$

Diagonally dominant matrix

If A is a diagonally dominant matrix, then Jacobi iteration converges for any initial vector

Example (Gauss-Seidel & Jacobi Iterations)

$$\begin{bmatrix} 4 & -1 & 1 \\ 4 & -8 & 1 \\ -2 & 1 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ -21 \\ 15 \end{bmatrix}$$

$$x^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left\| b - Ax^0 \right\|_2 = 26.7395$$

Diagonally dominant matrix

$$x_1^1 = \frac{7 + x_2^0 - x_3^0}{4} = \frac{7}{4} = 1.75$$

$$x_2^1 = \frac{21 + 4x_1^1 + x_3^0}{8} = \frac{21 + 4 \times 1.75}{8} = 3.5$$

$$x_3^1 = \frac{15 + 2x_1^1 - x_2^1}{5} = \frac{15 + 2 \times 1.75 - 3.5}{5} = 3.0$$

$$\left\| b - Ax^1 \right\|_2 = 3.0414$$

$$\left\| b - Ax^1 \right\|_2 = 10.0452$$

GS iteration

Example (Gauss-Seidel & Jacobi Iterations)

$$x_1^2 = \frac{7 + x_2^1 - x_3^1}{4} = \frac{7 + 3.5 - 3}{4} = 1.875$$

$$x_2^2 = \frac{21 + 4x_1^2 + x_3^1}{8} = \frac{21 + 4 \times 1.875 + 3}{8} = 3.9375$$

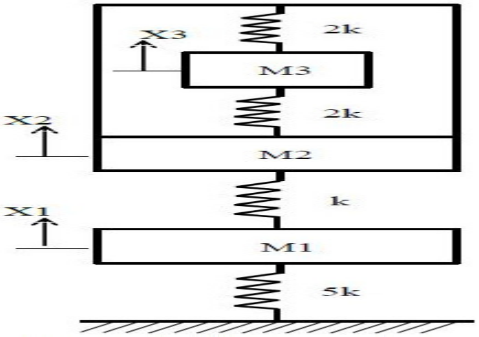
$$x_3^2 = \frac{15 + 2x_1^2 - x_2^2}{5} = \frac{15 + 2 \times 1.875 - 3.9375}{5} = 2.9625$$

$$\left\| b - Ax^2 \right\|_2 = 0.4765$$

$$\left\| b - Ax^2 \right\|_2 = 6.7413$$

GS iteration

When both Jacobi and Gauss-Seidel iterations converge, Gauss-Seidel converges faster



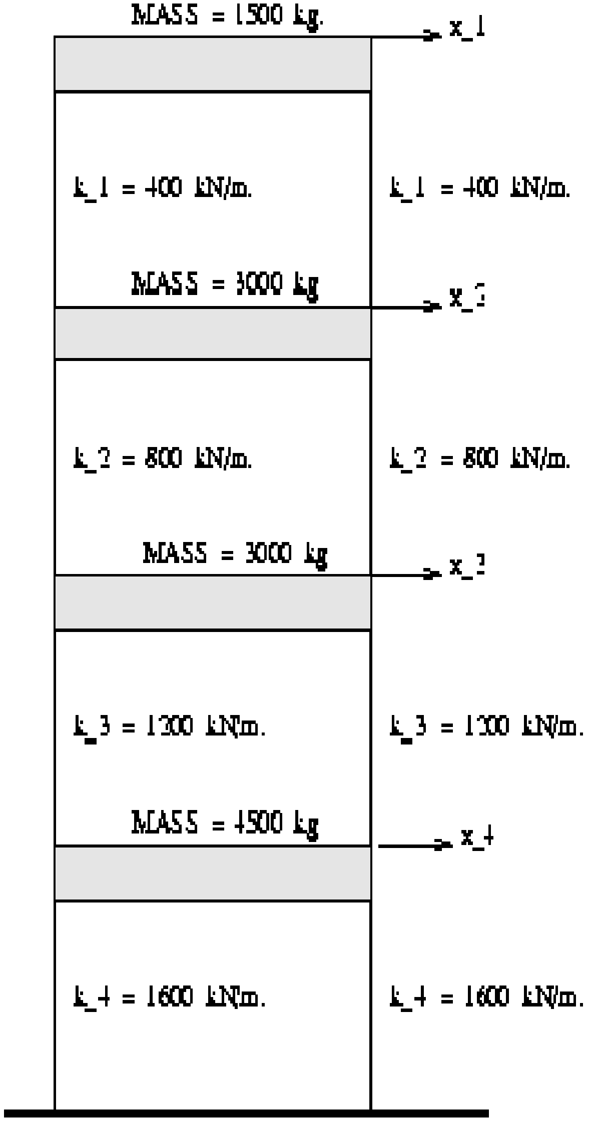
Mass and stiffness matrices

$$\mathbf{M}\frac{d^2\mathbf{x}}{dt^2}+\mathbf{K}\mathbf{x}=\mathbf{0}$$

m=W/g = (386.4k) /(396.4 in/sec^2)= 1.0 kip-sec^2/in

$$[m]=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\qquad [k]=\begin{bmatrix}500&-250&0\\-250&500&-250\\0&-250&250\end{bmatrix}$$

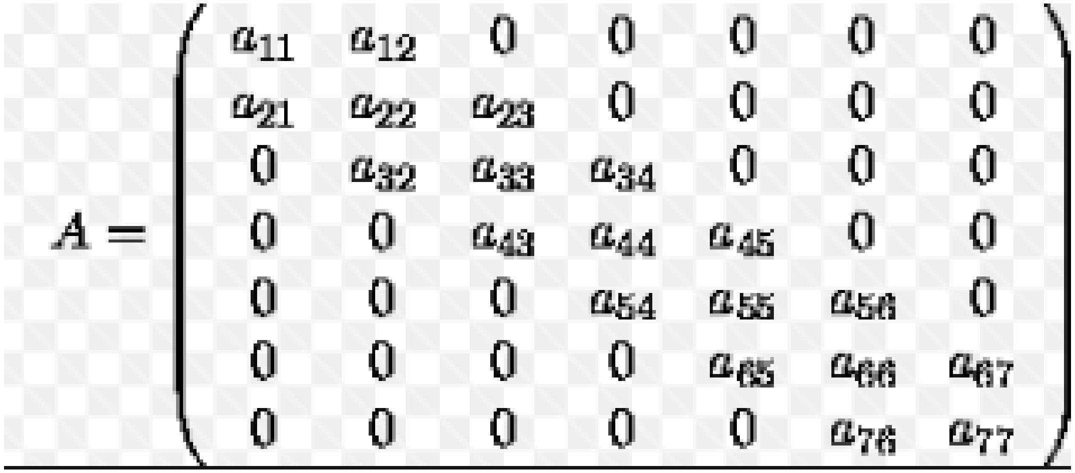
$$[k]-\omega_n^2[m]=\begin{bmatrix}500-\omega_n^2&-250&0\\-250&500-\omega_n^2&-250\\0&-250&250-\omega_n^2\end{bmatrix}$$



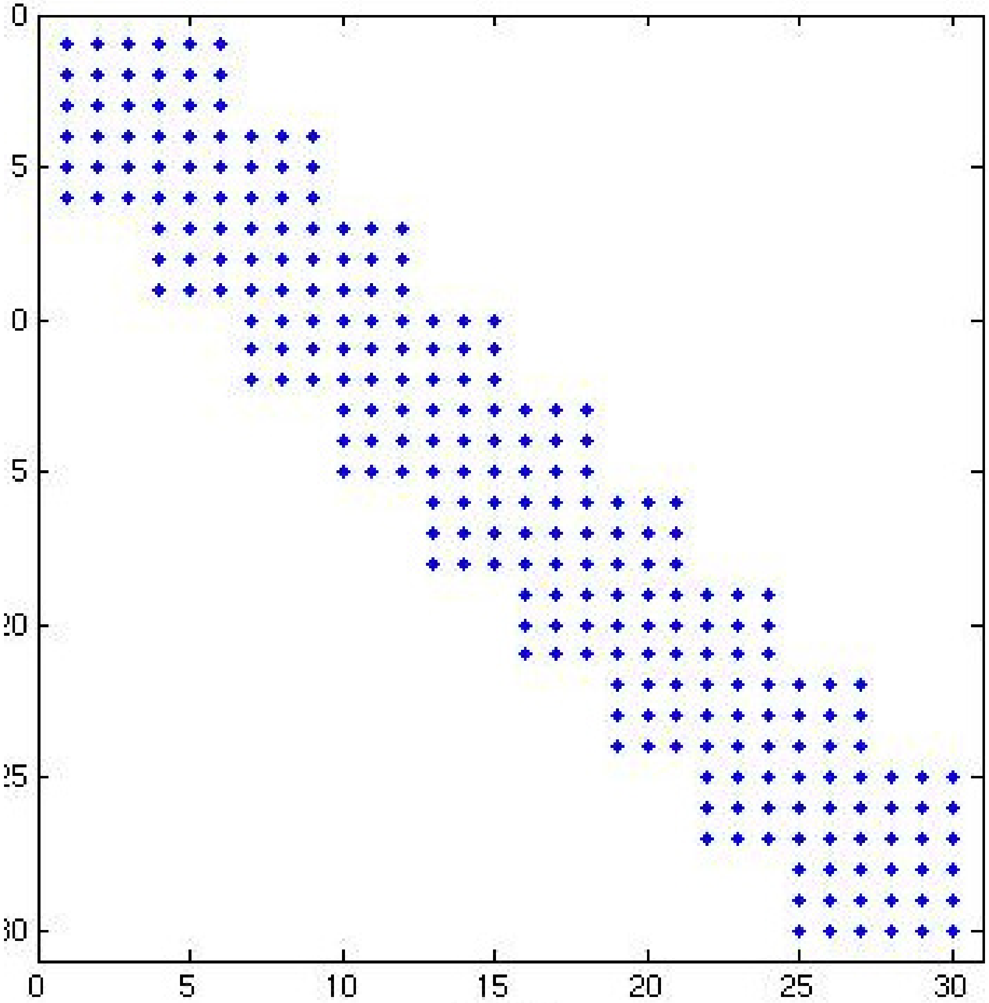
$$\text{STIFFNESS} = \begin{bmatrix}800&-800&&\\-800&2400&-1600&\\&-1600&4000&-2400\\&&-2400&5600\end{bmatrix}$$

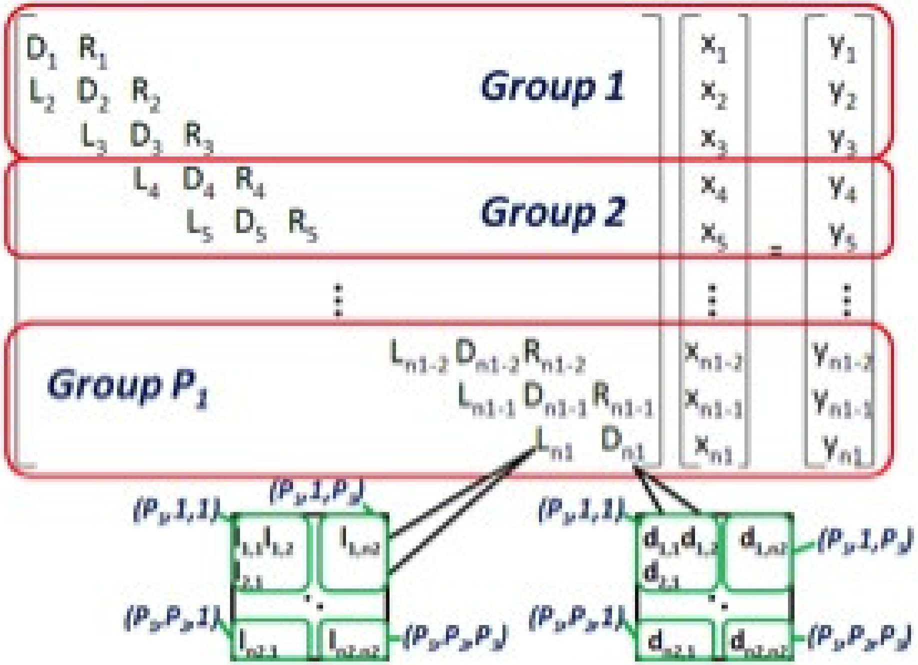
$$\text{MASS} = \begin{bmatrix}1500&&&\\&3000&&\\&&3000&\\&&&4500\end{bmatrix}$$

$$-\frac{\partial^2 u}{\partial x^2} \approx \frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} = [K]\cdot u$$ **= Ax**

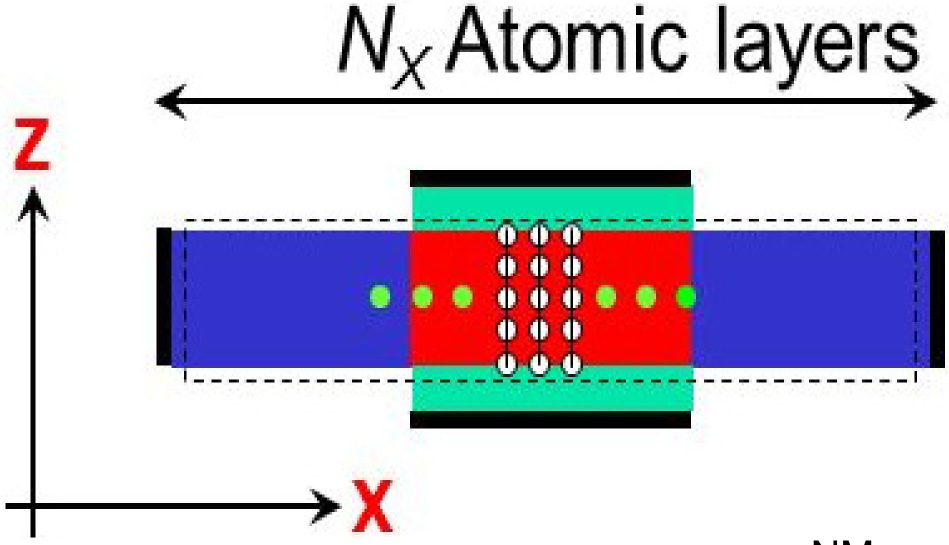


$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & 0 & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} & 0 \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} & a_{67} \\ 0 & 0 & 0 & 0 & 0 & a_{76} & a_{77} \end{pmatrix}$$




Group 1
Group 2
Group P_1
D_1 R_1
L_2 D_2 R_2
L_3 D_3 R_3
L_4 D_4 R_4
L_5 D_5 R_5
L_{n1-2} D_{n1-2} R_{n1-2}
L_{n1-1} D_{n1-1} R_{n1-1}
L_{n1} D_{n1}
x_1 x_2 x_3 x_4 x_5 x_{n1-2} x_{n1-1} x_{n1}
=
y_1 y_2 y_3 y_4 y_5 y_{n1-2} y_{n1-1} y_{n1}
$(P_y,1,1)$
$(P_y,1,P_y)$
$(P_y,P_y,1)$
(P_y,P_y,P_y)
$l_{1,1}$ $l_{1,2}$ $l_{2,1}$ $l_{1,n2}$ $l_{n2,1}$ $l_{n2,n2}$
$d_{1,1}$ $d_{1,2}$ $d_{2,1}$ $d_{1,n2}$ $d_{n2,1}$ $d_{n2,n2}$

Device Hamiltonian



$$H_{TB}\left(\vec{k}_y\right) = \begin{bmatrix} [A] & [C] & & & \\ [C]^\dagger & [B] & [D] & & \\ & [D]^\dagger & [A] & [C] & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots \end{bmatrix}$$

Tridiagonal Matrix

$$\begin{bmatrix} f_1 & g_1 & & & & & & & \\ e_2 & f_2 & g_2 & & & & & & \\ & e_3 & f_3 & g_3 & & & & & \\ & & \mathrm{O} & \mathrm{O} & \mathrm{O} & & & & \\ & & & e_i & f_i & g_i & & & \\ & & & & \mathrm{O} & \mathrm{O} & \mathrm{O} & & \\ & & & & & & e_{n-1} & f_{n-1} & g_{n-1} \\ & & & & & & & e_n & f_n \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \mathrm{M} \\ x_i \\ \mathrm{M} \\ x_{n-1} \\ x_n \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ \mathrm{M} \\ r_i \\ \mathrm{M} \\ r_{n-1} \\ r_n \end{Bmatrix}$$

- **Special case of banded matrix with bandwidth = 3**
- **Save storage, 3 × n instead of n × n**

Tridiagonal Systems

Tridiagonal Systems:

- The non-zero elements are in the main diagonal, super diagonal and subdiagonal.
- $a_{ij}=0$ *if* $|i-j| > 1$

$$\begin{bmatrix} 5 & 1 & 0 & 0 & 0 \\ 3 & 4 & 1 & 0 & 0 \\ 0 & 2 & 6 & 2 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

Tridiagonal Systems

- Occur in many applications
- Needs less storage (4n-2 compared to $n^2 + n$ for the general cases)
- Selection of pivoting rows is unnecessary (under some conditions)
- Efficiently solved by Gaussian elimination

$$\begin{bmatrix} 5 & 1 & 0 & 0 & 0 \\ 3 & 4 & 1 & 0 & 0 \\ 0 & 2 & 6 & 2 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

Algorithm to Solve Tridiagonal Systems

- Based on Naive Gaussian elimination.
- As in previous Gaussian elimination algorithms
  - Forward elimination step
  - Backward substitution step
- Elements in the super diagonal are not affected.
- Elements in the main diagonal, and B need updating

$$\begin{bmatrix} 5 & 1 & 0 & 0 & 0 \\ 3 & 4 & 1 & 0 & 0 \\ 0 & 2 & 6 & 2 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

Tridiagonal System

All the a elements will be zeros, need to update the d and b elements

The c elements are not updated

$$\begin{bmatrix} d_1 & c_1 & & & \\ a_1 & d_2 & c_2 & & \\ & a_2 & d_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_{n-1} & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \Rightarrow \begin{bmatrix} d_1 & c_1 & & & \\ & d_2' & c_2 & & \\ & & d_3' & \ddots & \\ & & & \ddots & c_{n-1} \\ & & & & d_n' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2' \\ b_3' \\ \vdots \\ b_n' \end{bmatrix}$$

Solving Tridiagonal System

Forward Eliminatio n

$$d_i \leftarrow d_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) c_{i-1}$$

$$b_i \leftarrow b_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) b_{i-1} \qquad 2 \leq i \leq n$$

Backward Substituti on

$$x_n = \frac{b_n}{d_n}$$

$$x_i = \frac{1}{d_i} (b_i - c_i x_{i+1}) \qquad \text{for} \quad i = n-1, n-2, \ldots, 1$$

Tridiagonal Matrix

$$\begin{bmatrix} f_1 & g_1 & & & & & \\ e_2 & f_2 & g_2 & & & & \\ & e_3 & f_3 & g_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & e_i & f_i & g_i & \\ & & & & \ddots & \ddots & \ddots \\ & & & & e_{n-1} & f_{n-1} & g_{n-1} \\ & & & & & e_n & f_n \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_i \\ \vdots \\ r_{n-1} \\ r_n \end{Bmatrix}$$

- Forward elimination

$$\begin{cases} f_k = f_k - \dfrac{e_k}{f_{k-1}} g_{k-1} \\ r_k = r_k - \dfrac{e_k}{f_{k-1}} r_{k-1} \end{cases} \quad k = 2,3,\ldots,n$$

Use factor = e_k / f_{k-1}

to eliminate subdiagonal element

Apply the same matrix operations to right hand side

- Back substitution

$$x_n = \frac{r_n}{f_n}$$

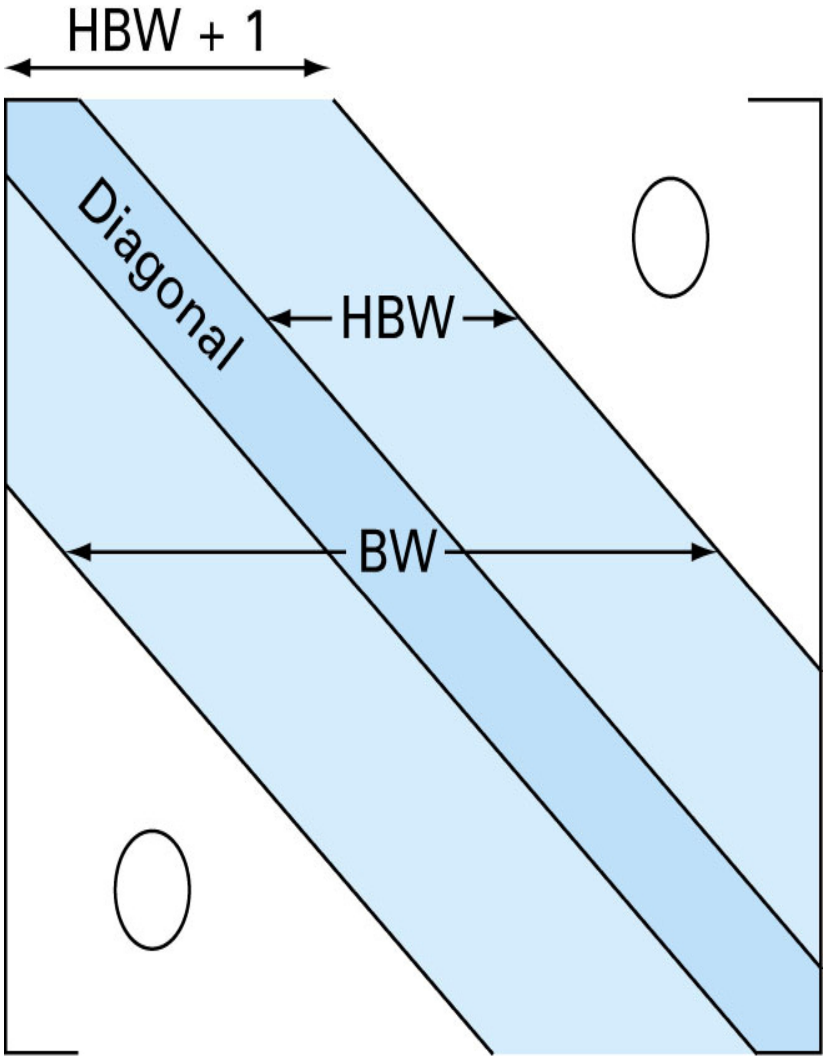
$$x_k = \frac{r_k - g_k x_{k+1}}{f_k} \qquad k = n-1, n-2, \ldots, 3,2,1$$

- Certain matrices have particular structures that can be exploited to develop efficient solution schemes (e.g. *banded*, *symmetric*)

- A *banded matrix* is a square matrix that has all elements equal to zero, with the exception of a **band** centered on the main diagonal.

- Standard Gauss Elimination is inefficient in solving banded equations because unnecessary space and time would be expended on the storage and manipulation of **zeros**.

- There is no need to store or process the zeros (off of the band)



Solving Tridiagonal Systems (Thomas Algorithm)

A tridiagonal system has a bandwidth of 3

$$\begin{bmatrix} f_1 & g_1 & & \\ e_2 & f_2 & g_2 & \\ & e_3 & f_3 & g_3 \\ & & e_4 & f_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{Bmatrix}$$

$$\boldsymbol{A} = \boldsymbol{L} * \boldsymbol{U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \boldsymbol{e'}_2 & 1 & 0 & 0 \\ 0 & \boldsymbol{e'}_3 & 1 & 0 \\ 0 & 0 & \boldsymbol{e'}_4 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 & \boldsymbol{g}_1 & & \\ & \boldsymbol{f}_2' & \boldsymbol{g}_2 & \\ & & \boldsymbol{f}_3' & \boldsymbol{g}_3 \\ & & & \boldsymbol{f}_4' \end{bmatrix}$$

DECOMPOSITION

```
DO    k = 2, n
        $e_k = e_k / f_{k-1}$
        $f_k = f_k - e_k g_{k-1}$
END DO
```

Time Complexity?

O(n)

vs. $O(n^3)$

Tridiagonal Systems

$$\begin{bmatrix}1&0&0&0\\e'_2&1&0&0\\0&e'_3&1&0\\0&0&e'_4&1\end{bmatrix}\overbrace{\begin{bmatrix}f_1&g_1&&\\&f_2'&g_2&\\&&f_3'&g_3\\&&&f_4'\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}}^{\{\,d\,\}}=\begin{bmatrix}r_1\\r_2\\r_3\\r_4\end{bmatrix}$$

$$\begin{bmatrix}1&0&0&0\\e'_2&1&0&0\\0&e'_3&1&0\\0&0&e'_4&1\end{bmatrix}\begin{bmatrix}d_1\\d_2\\d_3\\d_4\end{bmatrix}=\begin{bmatrix}r_1\\r_2\\r_3\\r_4\end{bmatrix}$$

$$\begin{bmatrix}f_1&g_1&&\\&f_2'&g_2&\\&&f_3'&g_3\\&&&f_4'\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}d_1\\d_2\\d_3\\d_4\end{bmatrix}$$

Forward Substitution

```
$d_1 = r_1$
DO    $k = 2, n$
    $d_k = r_k - e_k\, d_{k-1}$
END DO
```

Back Substitution

```
$x_n = d_n / f_n$
DO    $k = n-1, 1, -1$
    $x_k = (d_k - g_k \,.\, x_{k+1}) / f_k$
END DO
```

Hand Calculations: Tridiagonal Matrix

$$\begin{cases} f_k = f_k - \dfrac{e_k}{f_{k-1}} g_{k-1} \\ r_k = r_k - \dfrac{e_k}{f_{k-1}} r_{k-1} \end{cases}$$

$$\begin{bmatrix} 1 & -2 & 0 & 0 \\ -2 & 5 & -1 & 0 \\ 0 & -1 & 2 & -0.5 \\ 0 & 0 & -0.5 & 1.25 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} -3 \\ 5 \\ 2 \\ 3.5 \end{Bmatrix}$$

$$\begin{bmatrix} f_1 & g_1 & & \\ e_2 & f_2 & g_2 & \\ & e_3 & f_3 & g_3 \\ & & e_4 & f_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}$$

$$x_n = \frac{r_n}{f_n}$$

$$x_k = \frac{r_k - g_k x_{k+1}}{f_k}$$

(a) Forward elimination

$$\begin{cases} f_2 = f_2 - \dfrac{e_2}{f_1} g_1 = 5 - \dfrac{-2}{1}(-2) = 1 \\ r_2 = r_2 - \dfrac{e_2}{f_1} r_1 = 5 - \dfrac{-2}{1}(-3) = -1 \end{cases}$$

$$\begin{cases} f_3 = f_3 - \dfrac{e_3}{f_2} g_2 = 2 - \dfrac{-1}{1}(-1) = 1 \\ r_3 = r_2 - \dfrac{e_2}{f_1} r_1 = 2 - \dfrac{-1}{1}(-1) = 1 \end{cases}$$

$$\begin{cases} f_4 = f_4 - \dfrac{e_4}{f_3} g_3 = 1.25 - \dfrac{-0.5}{1}(-0.5) = 1 \\ r_4 = r_4 - \dfrac{e_4}{f_3} r_3 = 3.5 - \dfrac{-0.5}{1}(1) = 4 \end{cases}$$

(b) Back substitution

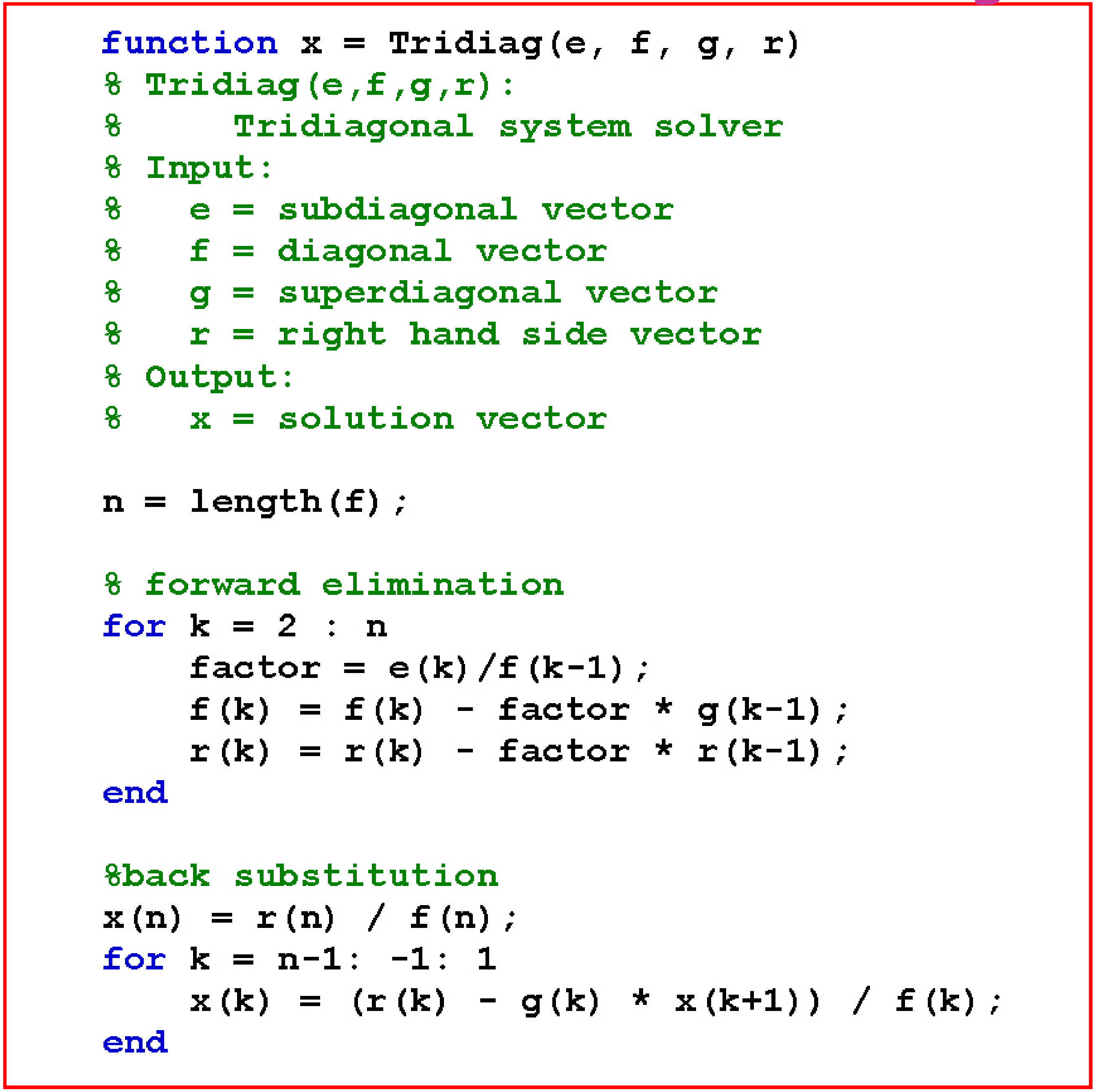
$$x_4 = \frac{r_4}{f_4} = \frac{4}{1} = 4$$

$$x_3 = \frac{r_3 - g_3 x_4}{f_3} = \frac{1 - (-0.5)(4)}{1} = 3$$

$$x_2 = \frac{r_2 - g_2 x_3}{f_2} = \frac{-1 - (-1)(3)}{1} = 2$$

$$x_1 = \frac{r_1 - g_1 x_2}{f_1} = \frac{-3 - (-2)(2)}{1} = 1$$

MATLAB M-file: `Tridiag`



```
function x = Tridiag(e, f, g, r)
% Tridiag(e,f,g,r):
%     Tridiagonal system solver
% Input:
%   e = subdiagonal vector
%   f = diagonal vector
%   g = superdiagonal vector
%   r = right hand side vector
% Output:
%   x = solution vector

n = length(f);

% forward elimination
for k = 2 : n
    factor = e(k)/f(k-1);
    f(k) = f(k) - factor * g(k-1);
    r(k) = r(k) - factor * r(k-1);
end

%back substitution
x(n) = r(n) / f(n);
for k = n-1: -1: 1
    x(k) = (r(k) - g(k) * x(k+1)) / f(k);
end
```

Example: Tridiagonal matrix

$$\begin{bmatrix} 1 & -2 & & & & \\ -2 & 6 & 4 & & & \\ & 4 & 9 & -0.5 & & \\ & & -0.5 & 3.25 & 1.5 & \\ & & & 1.5 & 1.75 & -3 \\ & & & & -3 & 13 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{Bmatrix} = \begin{Bmatrix} -3 \\ 22 \\ 35.5 \\ =7.75 \\ 4 \\ -33 \end{Bmatrix}$$

```
function [e,f,g,r] = example

e=[ 0 -2     4  -0.5  1.5  -3];
f=[ 1  6     9  3.25 1.75  13];
g=[-2  4  -0.5   1.5   -3   0];
r=[-3 22  35.5 -7.75    4 -33];
```

```
» [e,f,g,r] = example

e =
            0    -2.0000     4.0000    -0.5000     1.5000    -3.0000
f =
       1.0000     6.0000     9.0000     3.2500     1.7500    13.0000
g =
      -2.0000     4.0000    -0.5000     1.5000    -3.0000          0
r =
      -3.0000    22.0000    35.5000    -7.7500     4.0000   -33.0000

» x = Tridiag (e, f, g, r)
x =      1     2     3    -1    -2    -3
```

Note: `e(1) = 0 and g(n) = 0`

Example 2

Solve

$$\begin{bmatrix} 5 & 2 & & \\ 1 & 5 & 2 & \\ & 1 & 5 & 2 \\ & & 1 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 9 \\ 8 \\ 6 \end{bmatrix} \qquad \Rightarrow D = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix}, A = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \end{bmatrix}, C = \begin{bmatrix} 2 \\ 2 \\ 2 \\ \end{bmatrix}, \ B = \begin{bmatrix} 12 \\ 9 \\ 8 \\ 6 \end{bmatrix}$$

Forward Eliminatio n

$$d_i \leftarrow d_i - \left(\frac{a_{i-1}}{d_{i-1}}\right)c_{i-1}, \quad b_i \leftarrow b_i - \left(\frac{a_{i-1}}{d_{i-1}}\right)b_{i-1} \qquad 2 \le i \le 4$$

Backward Substituti on

$$x_n = \frac{b_n}{d_n}, \quad x_i = \frac{1}{d_i}\left(b_i - c_i x_{i+1}\right) \qquad \text{for} \quad i = 3,2,1$$

Example 2

$$D=\begin{bmatrix}5\\5\\5\\5\end{bmatrix}, A=\begin{bmatrix}1\\1\\1\\ \end{bmatrix}, C=\begin{bmatrix}2\\2\\2\\ \end{bmatrix}, \ B=\begin{bmatrix}12\\9\\8\\6\end{bmatrix}$$

Forward Elimination

$$d_2=d_2-\left(\frac{a_1}{d_1}\right)c_1=5-\frac{1\times 2}{5}=4.6, \quad b_2=b_2-\left(\frac{a_1}{d_1}\right)b_1=9-\frac{1\times 12}{5}=6.6$$

$$d_3=d_3-\left(\frac{a_2}{d_2}\right)c_2=5-\frac{1\times 2}{4.6}=4.5652, \quad b_3=b_3-\left(\frac{a_2}{d_2}\right)b_2=8-\frac{1\times 6.6}{4.6}=6.5652$$

$$d_4=d_4-\left(\frac{a_3}{d_3}\right)c_3=5-\frac{1\times 2}{4.5652}=4.5619, \quad b_4=b_4-\left(\frac{a_3}{d_3}\right)b_3=6-\frac{1\times 6.5652}{4.5652}=4.5619$$

Example 2
Backward Substitution

- After the Forward Elimination:
- Backward Substitution:

$$D^T = [5 \quad 4.6 \quad 4.5652 \quad 4.5619],\ B^T = [12 \quad 6.6 \quad 6.5652 \quad 4.5619]$$

$$x_4 = \frac{b_4}{d_4} = \frac{4.5619}{4.5619} = 1,$$

$$x_3 = \frac{b_3 - c_3 x_4}{d_3} = \frac{6.5652 - 2\times 1}{4.5652} = 1$$

$$x_2 = \frac{b_2 - c_2 x_3}{d_2} = \frac{6.6 - 2\times 1}{4.6} = 1$$

$$x_1 = \frac{b_1 - c_1 x_2}{d_1} = \frac{12 - 2\times 1}{5} = 2$$

Thank You!

Gauss-Seidel Method Algorithm

$$\begin{pmatrix} a_{11} & a_{12} & \mathrm{L} & a_{1n} \\ a_{21} & a_{22} & & \\ \mathrm{M} & & \mathrm{O} & \\ a_{n1} & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \\ b_n \end{pmatrix}$$

Split A into an upper component, a diagonal component and a lower component

Upper triangular, U

Gauss-Seidel Method Algorithm

$$\begin{pmatrix} a_{11} & a_{12} & \mathrm{L} & a_{1n} \\ a_{21} & a_{22} & & \\ \mathrm{M} & & \mathrm{O} & \\ a_{n1} & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \\ b_n \end{pmatrix}$$

Split A into an upper component, a diagonal component and a lower component

Diagonal, D

Gauss-Seidel Method Algorithm

$$\begin{pmatrix} a_{11} & a_{12} & \mathrm{L} & a_{1n} \\ a_{21} & a_{22} & & \\ \mathrm{M} & & \mathrm{O} & \\ a_{n1} & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \\ b_n \end{pmatrix}$$

Split A into an upper component, a diagonal component and a lower component

Lower triangular, L