- Matlab 210

- Jubaer Islam Nabib

$$Ax = b$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ a_{M1} & \cdots & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}$$

Using Gauss Jordan Elimination and Pivoting

$$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 = a_{14}$$

$$a_{21} x_1 + a_{22} x_2 + a_{23} x_3 = a_{24}$$

$$a_{31} x_1 + a_{32} x_2 + a_{33} x_3 = a_{34}$$

S1 : write the augmented matrix

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{array}\right]$$

S2 : Make all elements 0 below each pivots

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{array}\right]$$

$$a_{21} - \frac{a_{11}}{a_{11}} \times a_{21} = a_{21} - K_2 a_{11}$$

$$K_2 = \frac{a_{21}}{a_{11}}$$

$a_{31} - K_3 a_{11}$ where $K_3 = \dfrac{a_{31}}{a_{11}}$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} - K_2 a_{11} & a_{22} - K_2 a_{12} & a_{23} - K_2 a_{13} & a_{24} - K_2 a_{14} \\ a_{31} - K_3 a_{11} & a_{32} - K_2 a_{12} & a_{33} - K_3 a_{13} & a_{34} - K_3 a_{14} \end{bmatrix}$$

see, both are zero

reduced equation

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ 0 & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ 0 & \alpha_{32} & \alpha_{33} & \alpha_{34} \end{bmatrix}$$

$$u = \frac{a_{32}}{a_{22}} \quad \therefore$$

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ 0 & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ 0 & a_{32} - u\,a_{22} & a_{32} - u\,a_{23} & \alpha_{34} - u\,a_{24} \end{bmatrix}$$

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{22} & A_{23} & A_{24} \\ 0 & 0 & A_{33} & A_{34} \end{bmatrix}$$

```
A = [2 3 5 23;3 4 1 14;6 7 2 26] ;

for m=1:2
    for i=m+1:3
        k =  A(i,m) / A(m,m)
        for j = 1:4
            A(i,j) = A(i,j) - A(m,j)*k

        end
    end
end

disp(A)
n=3;
x = zeros(3,1)
x(n) = A(n, n+1)/A(n,n);

for i=n-1:-1:1
    sum = 0;
    for j=i+1:n
        sum = sum + A(i,j)*x(j);
    end
    x(i) = (A(i,n+1)-sum)/A(i,i);
end

% Display solution
disp('Solution vector x:');
disp(x);
```

$$x_3 = \frac{A_{34}}{A_{33}}$$

$$x_2 = \frac{\overset{i\ \ n+1}{A_{24}} - A_{23} \times x_3}{A_{22}}$$

$$x_1 = \frac{\overset{i\ \ n+1}{A_{14}} - (A_{13}x_3 + A_{12}x_2)}{A_{11}}$$

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Gauss - Seidel Iterative Method

$$X^{(0)} = [0 \quad 0 \quad 0]$$

$$x_1^{k+1} = \frac{1}{10}(19 - 3x_2^{k} - 1x_3^{k})$$

$$x_2^{k+1} = \frac{1}{10}(29 - 3x_1^{k+1} - 2x_3^{k})$$

$$x_3^{k+1} = \frac{1}{10}(35 - 1x_1^{k+1} - 2x_2^{k+1})$$

$$10x_1 + 3x_2 + 1x_3 = 19$$

$$3x_1 + 10x_2 + 2x_3 = 29$$

$$1x_1 + 2x_2 + 10x_3 = 35$$

real soln

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

1st iteration :

$$x_1^1 = \frac{1}{10}(19 - 0 - 0) = 1.9$$

$$x_2^1 = \frac{1}{10}(29 - 3 \times 1.9 + 2 \times 0) = 2.33$$

$$x_3^1 = \frac{1}{10}(35 - 1 \times 1.9 - 2 \times 2.33) = 2.844$$

2nd iteration :

$$x_1^2 = \frac{1}{10}(19 - 3x_1 - x_3^1) = 0.9167$$

$$x_2^2 = \frac{1}{10}(29 - 3x_1^2 + 2x_3^1) = 2.0562$$

$$x_3^2 = \frac{1}{10}(35 - 1x_1^2 - 2x_2^2) = 2.9971$$

3nd iteration :

$x_1^3 = 0.9834$

$x_2^3 = 2.0056$

$x_3^3 = 3.0005$

যত iteration
তত accurate

```matlab
% Gauss-Seidel Method for Solving Ax = B
clc;
clear;
close all;

% Input Coefficient Matrix A, Source Vector B, and Initial Guess
A = [10 3 1; 3 10 2; 1 2 10];
B = [19; 29; 35];
P = [0; 0; 0]; % Initial guess
n = 10; % Maximum number of iterations
e = 0.0001; % Tolerance for convergence

% Matrix size
N = length(B);
X = zeros(N,1); % Initialize the solution vector

% Gauss-Seidel Iterations
for j = 1:n
    old_P = P; % Save the previous iteration solution

    for i = 1:N
        % Compute updated value of X(i)
        P(i) = (B(i) - A(i, [1:i-1, i+1:N]) * P([1:i-1, i+1:N])) / A(i,i)

    end

    % Display the iteration number and the current solution
    fprintf('Iteration no %d\n', j);
    disp(P);

    % Check for convergence using the relative error
    if max(abs(P - old_P)) < e
        fprintf('Converged after %d iterations.\n', j);
        break;
    end
end

% Display Final Solution
fprintf('Final Solution:\n');
disp(P);
```