~~Q For~~

Q For given matrisc A in the pdf & vector b (last 2 digits of roll no. + 2) ~~b = 40~~ b(i) = 40 find the Solution of system of linear equations

$AX = b$ using Jacobi's iteration & gauss seidel method.

Sol

Jacobi iteration Method.

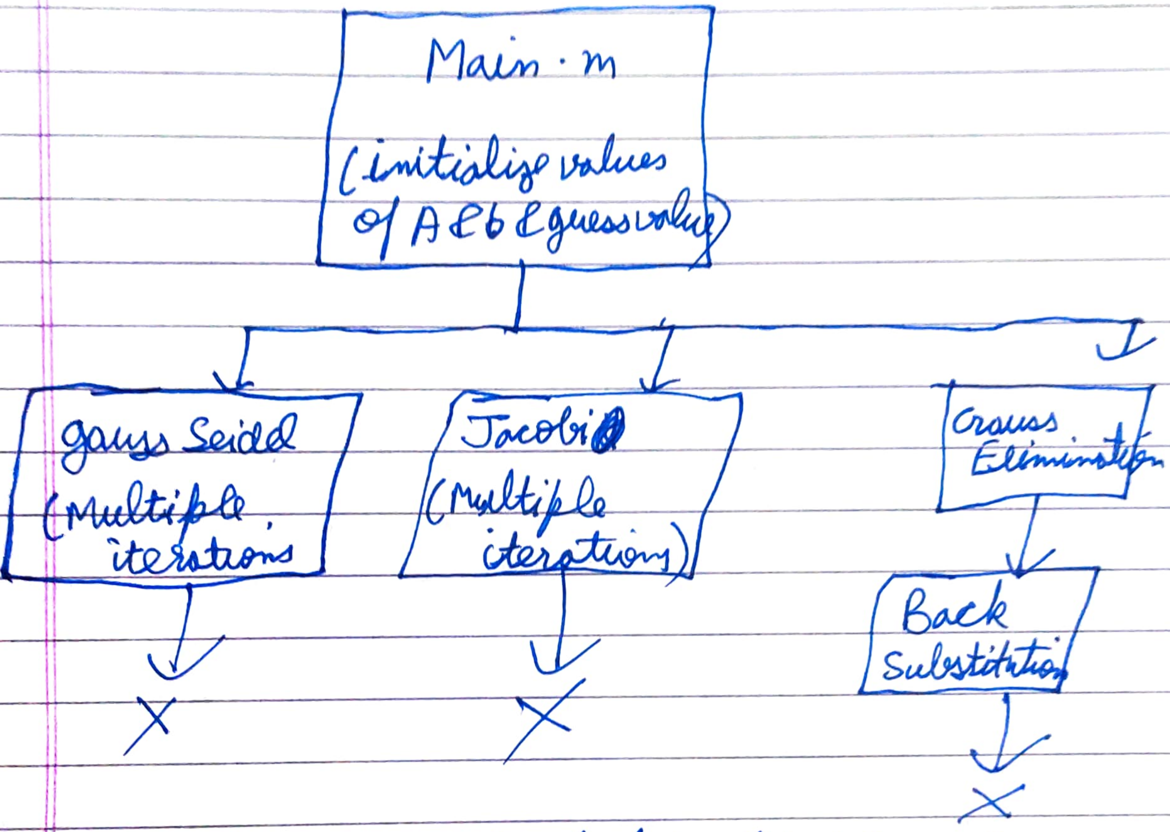
$$X^{K+1}(i) = \left(b(i) - \sum_{j=1}^{i-1} A(i,j)\, X^{K+1}(j) - \sum_{j=i+1}^{n} A(i,j)\, X^{K}(j)\right) / A(i,i$$

Grauss Seidel Method.

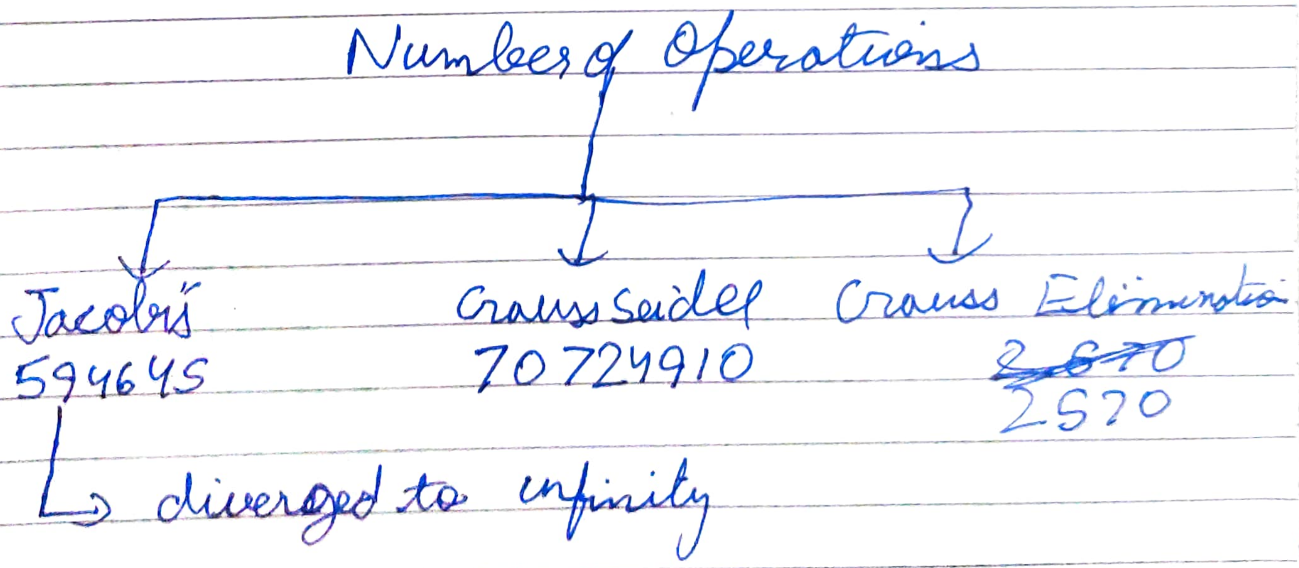
$$X^{K+1}(i) = \left(b(i) - \sum_{j=1}^{i-1} A(i,j)\, X^{K}(j) - \sum_{j=i+1}^{n} A(i,j)\, X^{K}(j)\right.$$

$$/ A(i,$$

FLOWCHART



In gauss Seidel ~~Multiple~~ valuess of X are changed ~~to~~ n times even in single iteration.



```
A = load("A.txt");
n = size(A)% getting the size of matrix A
b = zeros(15,1);% initialising values of b vector
for i = 1:15
    b(i)= 40;
end
guessX = zeros(15,1); % guess value
gaussSeidel(A,b,guessX) % result using gauss Seidel
jacobi(A,b,guessX) % result using jacobi's iteration method
GEM(A,b) % result using Gauss elimination
```

```
n =

    15    15

    "Operation count for gauss seidel method 70724910"


ans =

   1.0e+05 *

    0.0120
    0.0660
    0.1564
    0.2780
    0.4260
    0.5960
    0.7840
    0.9864
    1.2000
    1.4220
    1.6500
    1.8820
    2.1164
    2.3520
    2.5880

    "Operation count for jacobi iteration method 594645"


ans =

   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   Inf
  -Inf
   Inf
  -Inf
   Inf

No. of operations in Gauss elimination
        2345

No. of operations in back-substitution
   225


ans =

   1.0e+05 *

  Columns 1 through 7

    0.0120    0.0660    0.1564    0.2780    0.4260    0.5960    0.7840

  Columns 8 through 14

    0.9864    1.2000    1.4220    1.6500    1.8820    2.1164    2.3520

  Column 15

    2.5880
```

Published with MATLAB® R2021a

```
function result = jacobi(A,b,previousX)
counter = 0;
[n n] = size(A);
result = previousX;
for i = 1:n
    %Code for pivoting
    var = A(i,i);
    for j = i+1:n
        if abs(A(j,i))>abs(var)
            temp = A(i,:);
            A(i,:)= A(j,:);
            A(j,:) = temp;
            temp1 = b(i);
            b(i) = b(j);
            b(j)=temp1;
            b(j)=temp1;
            var = A(i,i);
        end
    end
    val = b(i);
    for j = 1:n
        if j ~= i
            % using initial values from guess vector X
            val = val - A(i,j)*previousX(j);
            counter = counter + 2;
        end
    end
    val = val/A(i,i);
    counter = counter + 1;
    result(i) = val;
end

while max(abs((result-previousX)./result)) >0.00001 % Checking for max
 error
% Re-iterating for closer values of X
previousX = result;
for i = 1:n
    %Code for pivoting
    var = A(i,i);
    for j = i+1:n
        if abs(A(j,i))>abs(var)
            temp = A(i,:);
            A(i,:)= A(j,:);
            A(j,:) = temp;
            temp1 = b(i);
            b(i) = b(j);
            b(j)=temp1;
            b(j)=temp1;
            var = A(i,i);
        end
    end
    val = b(i);
    for j = 1:n
        if j ~= i
            val = val - A(i,j)*previousX(j);
            counter = counter +2;
        end
    end
    val = val/A(i,i);
    counter = counter +1;
    result(i) = val;
end
end
% In Jacobi's iteration method the values of X are diverging even
 after
% pivoting
display("Operation count for jacobi iteration method "+counter);
return
end
```

```
Not enough input arguments.

Error in jacobi (line 3)
[n n] = size(A);
```

Published with MATLAB® R2021a

```
function result = gaussSeidel(A,b,previousX)
counter = 0;
[n n] = size(A);
result = previousX;
for i = 1:n
    %Code for pivoting
    var = A(i,i);
    for j = i+1:n
        if abs(A(j,i))>abs(var)
            temp = A(i,:);
            A(i,:)= A(j,:);
            A(j,:) = temp;
            temp1 = b(i);
            b(i) = b(j);
            b(j)=temp1;
            b(j)=temp1;
            var = A(i,i);
        end
    end
    % first iteration
    val = b(i);
    for j = 1:n
        if j ~= i
            val = val - A(i,j)*result(j);
            counter = counter+2;
        end
    end
    val = val/A(i,i);
    counter = counter +1;
    result(i) = val; % replacing values in result simultaniously
end
while max(abs((result-previousX)./result)) > 0.00000000001 %checking
 for max error
%Re-iterating to get closer values of result
previousX = result;
for i = 1:n
    %Code for pivoting
    var = A(i,i);
    for j = i+1:n
        if abs(A(j,i))>abs(var)
            temp = A(i,:);
            A(i,:)= A(j,:);
            A(j,:) = temp;
            temp1 = b(i);
            b(i) = b(j);
            b(j)=temp1;
            b(j)=temp1;
            var = A(i,i);
        end
    end
    val = b(i);
    for j = 1:n
```

```
        if j ~= i
            val = val - A(i,j)*result(j);
            counter = counter +2;
        end
    end
    val = val/A(i,i);
    counter = counter +1;
    result(i) = val;
end
end
display("Operation count for gauss seidel method "+counter);
return
end
```

```
Not enough input arguments.

Error in gaussSeidel (line 3)
[n n] = size(A);
```

Published with MATLAB® R2021a

```
function x = GEM(A,b)
[m n] = size(A);
counter =0;
for i = 1:m-1;
    var = A(i,i);
    for j = i+1:m;
        if abs(A(j,i))>abs(var);
            temp = A(i,:);
            A(i,:)= A(j,:);
            A(j,:) = temp;
            temp1 = b(i);
            b(i) = b(j);
            b(j)=temp1;
            b(j)=temp1;
            var = A(i,i);
        end
    end
    for j = i+1:m;
        factor = A(j,i)/A(i,i);
        counter = counter +1;
        A(j,:) = A(j,:) -factor*A(i,:);
        counter = counter + n+1-j +n -1;
        b(j) = b(j) - factor*b(i);
        counter = counter +2;
    end
end

disp("No. of operations in Gauss elimination");
disp(counter);

x = BackSubstitution(A,b);

return
end
```

```
Not enough input arguments.

Error in GEM (line 2)
[m n] = size(A);
```

Published with MATLAB® R2021a

```
function result = BackSubstitution(A,b);
[m n] =size(A);
x = 1:n;
counter =0;
for i = 1:n;
    val = b(n+1-i);
    j = i;
    while j>1;
        val = val - A(n+1-i,n+2-j)*x(n+2-j);
        counter = counter + 2;
        j = j-1;
    end
    val = val/ A(n+1-i,n+1-i);
    counter = counter + 1;
    x(n+1-i)=val;
end
disp("No. of operations in back-substitution");
disp(counter);
result = x;
return
end
```

```
Not enough input arguments.

Error in BackSubstitution (line 2)
[m n] =size(A);
```

Published with MATLAB® R2021a