23MAT204

MATHEMATICS FOR COMPUTING-3

Dr. Sarada Jayan
9448455760
6364333760

How solution to a linear system can be found as a solution of an optimization problem

Find an objective function whose solution is specified as x: Ax=b

$$Minimize\ \frac{1}{2}x^T Ax - b^T x + c,$$
$$x \in R^n, A = A^T, c \in R$$

$$Let\ \ f(x) = \frac{1}{2}x^T Ax - b^T x + c,$$
$$x \in R^n, A = A^T, c \in R$$
$$\nabla f(x) = 0 \in R^n \Rightarrow Ax - b = 0 \Rightarrow Ax = b$$
$$So, the\ solution\ is\ x\ such\ that\ Ax = b$$

Consider the system: $\begin{bmatrix} 4 & 1 & 2 \\ 1 & 12 & 3 \\ 2 & 3 & 8 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 7 \\ 13 \\ 6 \end{bmatrix}$

$$f(x,y) = \frac{1}{2}\mathrm{x}^{\mathrm{T}}\mathrm{A}\mathrm{x} - \mathrm{b}^{\mathrm{T}}\mathrm{x} + \mathrm{c} = \frac{1}{2}[\mathrm{x}\ \mathrm{y}\ \mathrm{z}]\begin{bmatrix} 4 & 1 & 2 \\ 1 & 12 & 3 \\ 2 & 3 & 8 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} - [7\ 13\ 6]\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathrm{c}$$

$$= 2x^2 + 6y^2 + 4z^2 + xy + 2xz + 3yz - 7x - 13y - 6z$$

$$\nabla f = 0 \rightarrow \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 4x + y + 2z - 7 \\ x + 12y + 3z - 13 \\ 2x + 3y + 8z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow Ax = b$$

$$Minimize \ \frac{1}{2}x^T Ax - b^T x + c,$$

$$x \in R^n, A = A^T, c \in R$$

is equivalent to

$$Ax = b$$

Find the optimization problem used to solve the system:

$$\begin{bmatrix} 8 & -1 & 5 \\ -1 & 12 & 3 \\ 5 & 3 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 12 \\ 14 \\ 11 \end{bmatrix}$$

How do we find x?

1. Use Gauss Elimination and solve Ax=b
2. Use Gauss Jacobi/Gauss Siedel Iterative methods to solve Ax=b
3. Solve original optimization problem using steepest descent method / gradient direction method
4. Use CGM to solve original optimization problem. (will learn this sem)
5. Use GMRES to solve original optimization problem. (can learn if interested)

Gauss Jacobi Iteration method to solve a system of linear equations $Ax=b$

Gauss – Jacobi Iteration method

- This is an iterative method to find the solution for Ax=b, where A is a square matrix of order n
- The algorithm starts with an initial vector $x^{(0)}$.

***Two assumptions made on Jacobi Method*:**

1. The system given by

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

Has a unique solution.

2. The coefficient matrix A has no zeros on its main diagonal, namely, $a_{11}, a_{22}, \dots, a_{nn}$ are nonzeros.

Gauss – Jacobi Iteration method

Main idea of Jacobi

To begin, solve the 1st equation for x_1, the 2nd equation for x_2 and so on to obtain the rewritten equations:

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \cdots a_{1n}x_n)$$

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

$$\vdots$$

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

$2x_1 - x_2 + x_3 = 5$
$x_1 + 8x_2 - x_3 = 6$
$x_1 + x_2 - 10x_3 = 9$

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

$x_2^{(k+1)} = \frac{6 - x_1^{(k)} + x_3^{(k)}}{8}$

$x_3^{(k+1)} = \frac{-9 + x_1^{(k)} + x_2^{(k)}}{10}$

Then make an initial guess of the solution $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \dots x_n^{(0)})$. Substitute these values into the right hand side the of the rewritten equations to obtain the *first approximation*, $\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots x_n^{(1)}\right)$.

This accomplishes one **iteration**.

In the same way, the *second approximation* $\left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \dots x_n^{(2)}\right)$ is computed by substituting the first approximation's x-vales into the right hand side of the rewritten equations.

By repeated iterations, we form a sequence of approximations $\boldsymbol{x}^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots x_n^{(k)}\right)^t, \quad k = 1,2,3, \dots$

For each $k \geq 1$, generate the components $x_i^{(k)}$ of $\boldsymbol{x}^{(k)}$ from $\boldsymbol{x}^{(k-1)}$ by

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[\sum_{\substack{j=1,\\ j\neq i}}^{n} (-a_{ij}x_j^{(k-1)}) + b_i\right], \qquad \text{for } i = 1,2,,\dots n$$

Gauss – Jacobi Iteration method

Numerical Algorithm of Jacobi Method

Input: $A = [a_{ij}]$, $\boldsymbol{b}, \boldsymbol{XO} = \boldsymbol{x}^{(0)}$, tolerance TOL, maximum number of iterations N.

Step 1 Set $k = 1$

Step 2 while $(k \leq N)$ do Steps 3-6

Step 3 For for $i = 1,2,,\dots n$

$$x_i = \frac{1}{a_{ii}}\left[\sum_{\substack{j=1, \\ j\neq i}}^{n}(-a_{ij}\boldsymbol{XO}_j) + b_i\right],$$

Step 4 If $||\boldsymbol{x} - \boldsymbol{XO}|| < TOL$, then OUTPUT $(x_1\ , x_2\ , x_3\ , \dots x_n\ )$;

STOP.

Step 5 Set $k = k + 1$.

Step 6 For for $i = 1,2,,\dots n$

Set $\boldsymbol{XO}_i = x_i$.

Step 7 OUTPUT $(x_1\ , x_2\ , x_3\ , \dots x_n\ )$;

STOP.

Gauss – Jacobi Iteration method

Apply the Jacobi method to solve

$$5x_1 - 2x_2 + 3x_3 = -1$$
$$-3x_1 + 9x_2 + x_3 = 2$$
$$2x_1 - x_2 - 7x_3 = 3$$

Solution To begin, rewrite the system

$$x_1 = \frac{-1}{5} + \frac{2}{5}x_2 - \frac{3}{5}x_3$$
$$x_2 = \frac{2}{9} + \frac{3}{9}x_1 - \frac{1}{9}x_3$$
$$x_3 = -\frac{3}{7} + \frac{2}{7}x_1 - \frac{1}{7}x_2$$

Choose the initial guess $x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$

The first approximation is

Iterative formula

$$x_1^{(k)} = -\frac{1}{5} + \frac{2}{5}x_2^{(k-1)} - \frac{3}{5}x_3^{(k-1)}$$
$$x_2^{(k)} = \frac{2}{9} + \frac{3}{9}x_1^{(k-1)} - \frac{1}{9}x_3^{(k-1)}$$
$$x_3^{(k)} = -\frac{3}{7} + \frac{2}{7}x_1^{(k-1)} - \frac{1}{7}x_2^{(k-1)}$$

Ans.1

$$x_1^{(1)} = \frac{-1}{5} + \frac{2}{5}(0) - \frac{3}{5}(0) = -0.200$$
$$x_2^{(1)} = \frac{2}{9} + \frac{3}{9}(0) - \frac{1}{9}(0) = 0.222$$
$$x_3^{(1)} = -\frac{3}{7} + \frac{2}{7}(0) - \frac{1}{7}(0) = -0.429$$

Ans.2

$x_1^{(2)} = 0.146$

$x_2^{(2)} = 0.203$

$x_3^{(2)} = -0.517$

Continue iteration, we obtain

n	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$x_1^{(k)}$	0.000	-0.200	0.146	0.192			
$x_2^{(k)}$	0.000	0.222	0.203	0.328			
$x_2^{(k)}$	0.000	-0.429	-0.517	-0.416			

Is there any easy way to get these values using MATLAB ?

$Ax=b$ (Gauss-Jacobi in matrix form)

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

$$= D - \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ -a_{21} & 0 & 0 \\ -a_{31} & -a_{32} & 0 \end{bmatrix}}_{L} - \underbrace{\begin{bmatrix} 0 & -a_{12} & -a_{13} \\ 0 & 0 & -a_{23} \\ 0 & 0 & 0 \end{bmatrix}}_{U}$$

$\Longrightarrow A = D - L - U$

$$Ax=b \longrightarrow (D-L-U)x=b$$

$$D\bar{x} = b + (L+U)\bar{x}$$

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

$\therefore$ Soln. of $Ax=b$ is $\bar{x}^{(k)} = \underbrace{D^{-1}(L+U)}_{T}x^{(k-1)} + \underbrace{D^{-1}b}_{c}$, $\quad \bar{x}^{(k)} = T\,x^{(k-1)} + c$

```
A=[5,-2,3;-3,9,1;2,-1,-7];
b=[-1;2;3];
n=size(A,1)
D=diag(diag(A));
L=-tril(A,-1);
% to generate the L matrix with negative values
% of A in lower triangular part and also with
diagonal zero
U=-triu(A,1);
% to generate the U matrix with negative values
% of A in upper triangular part and also with
diagonal zero
T=inv(D)*(L+U);
c=inv(D)*b;
x0=[0;0;0];
x1=T*x0+c
x2=T*x1+c
x3=T*x2+c
x4=T*x3+c
x5=T*x4+c
x6=T*x5+c
```

```
x1 = 3x1

   -0.2000
    0.2222
   -0.4286

x2 = 3x1

    0.1460
    0.2032
   -0.5175

x3 = 3x1

    0.1917
    0.3284
   -0.4159

x4 = 3x1

    0.1809
    0.3323
   -0.4207

x5 = 3x1

    0.1854
    0.3293
   -0.4244

x6 = 3x1

    0.1863
    0.3312
   -0.4226
```

Gauss – Jacobi Iteration for A_{nxn} in matrix form

Consider to solve an $n \times n$ size system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$ with $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$ for $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$.

We split A into

$$A = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \dots & 0 & 0 \\ -a_{21} & \dots & 0 & 0 \\ \vdots & & \ddots & \vdots \\ -a_{n1} & \dots & -a_{n,n-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & \dots & -a_{1n} \\ 0 & 0 & & \vdots \\ \vdots & \vdots & \ddots & -a_{n-1,n} \\ 0 & 0 & \dots & 0 \end{bmatrix} = D - L - U$$

$A\boldsymbol{x} = \boldsymbol{b}$ is transformed into $(D - L - U)\boldsymbol{x} = \boldsymbol{b}$

$$D\boldsymbol{x} = (L + U)\boldsymbol{x} + \boldsymbol{b}$$

Assume D^{-1} exists and $D^{-1} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & \dots & 0 \\ 0 & \frac{1}{a_{22}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{a_{nn}} \end{bmatrix}$

Then

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

The matrix form of Jacobi iterative method is

$$\boldsymbol{x}^{(k)} = D^{-1}(L + U)\boldsymbol{x}^{(k-1)} + D^{-1}\boldsymbol{b} \qquad k = 1,2,3, \dots$$

Define $T = D^{-1}(L + U)$ and $\boldsymbol{c} = D^{-1}\boldsymbol{b}$, Jacobi iteration method can also be written as

$$\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(k-1)} + \boldsymbol{c} \qquad k = 1,2,3, \dots$$

Gauss – Jacobi Iteration for A_{nxn} in matrix form

HW

Solve :-

$$9x_1 + 3x_2 + 5x_3 - x_4 = 16$$
$$2x_1 - 5x_2 - x_3 = -4$$
$$x_1 + 6x_2 + 11x_3 - x_4 = 17$$
$$x_1 + x_2 + x_3 - 5x_4 = -2$$

$\bar{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ Use G. Jacobi for 8 iterations.

Ans:- 8^{th} iteration $\rightarrow \begin{pmatrix} 0.9953 \\ 1.0007 \\ 0.9983 \\ 1.0028 \end{pmatrix} \sim \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

Gauss – Jacobi Iteration method

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

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

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

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

$$= x_i^k + \frac{1}{a_{ii}}\left[b_i - (\textit{dotproduct} \text{ of ith row of A with old x vector})\right]$$

```
for i=1:nRow
            xnew(i)=xold(i)+ (b(i)-A(i,:)*xold)/A(i,i);
End
xold=xnew
```

Gauss – Jacobi Iteration method

```
%Gauss Jacobi Iteration Method
clc;
clear all;
A=[-4 2 1 0 0;1 -4 1 1 0;2 1 -4 1 2;0 1 1 -4 1;0 0 1 2 -4];
 b=[-4 11 -16 11 -4];
 maxIter=1000;
 errorLimit=0.00001;
 resLimit=0.00001;
 x=[1,1,1,1,1]';

[xnew,k,relError]=my_Jacobi(A,x,b,maxIter,errorLimit,resLimit);
%Check the result
display('the solution vector is')
xnew'
display('recomputed b is')
(A*xnew)'
display('original b is')
b
```

Gauss – Jacobi Iteration method contd...

```
function [xnew,k,relError]=my_Jacobi(A,x,b,maxIter,errorLimit,resLimit)
    [nRow,nCol]=size(A);
    xold=x(:);  % convert into column vector if it is not
    b=b(:);% convert into column vector if it is not
    k=0;
    relError=zeros(maxIter,1);
    Notsolved=true;
    xnew=zeros(size(xold));
    while Notsolved
          k=k+1;
          for i=1:nRow
            xnew(i)=xold(i)+ (b(i)-A(i,:)*xold)/A(i,i);
           end
        currentError=norm(xnew-xold);
        relError(k)=currentError/norm(xnew);
        if norm(b-A*xnew)<=resLimit || currentError<=errorLimit || k>maxIter
            Notsolved=false;
        else
            xold=xnew;
        end
    end
end
```

Gauss – Jacobi Output

the solution vector after all iterations is

ans =

1.0000   -2.0000   4.0001   -2.0000   1.0000

recomputed b is

ans =

-4.0000   11.0000 -16.0000   11.0000   -4.0000

original b is

b =

-4   11   -16   11   -4

Gauss – Siedel Iteration method to solve a system of linear equations Ax=b

Gauss – Siedel Iteration method

With the Jacobi method, the values of $x_i^{(k)}$ obtained in the kth iteration remain unchanged until the entire $(k+1)$th iteration has been calculated. With the Gauss-Seidel method, we use the new values $x_i^{(k+1)}$ as soon as they are known. For example, once we have computed $x_1^{(k+1)}$ from the first equation, its value is then used in the second equation to obtain the new $x_2^{(k+1)}$, and so on.

$$
\begin{aligned}
5x_1 - 2x_2 + 3x_3 &= -1 \\
-3x_1 + 9x_2 + x_3 &= 2 \\
2x_1 - x_2 - 7x_3 &= 3
\end{aligned}
$$

$$x_1^{(k+1)} = \frac{-1 + 2x_2^{(k)} - 3x_3^{(k)}}{5}$$ ✓ (same as in J.)

$$x_2^{(k+1)} = \frac{2 + 3x_1^{(k+1)} - x_3^{(k)}}{9}$$

$$x_3^{(k+1)} = \frac{-3 + 2x_1^{(k+1)} - x_2^{(k+1)}}{7}$$

Gauss – Siedel Iteration method

Numerical Algorithm of Gauss-Seidel Method

Input: $A = [a_{ij}]$, $\boldsymbol{b}, \boldsymbol{XO} = \boldsymbol{x}^{(0)}$, tolerance TOL, maximum number of iterations N.

Step 1 Set $k = 1$

Step 2 while $(k \leq N)$ do Steps 3-6

Step 3 For for $i = 1,2,,\ldots n$

$$x_i = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1}(a_{ij}x_j) - \sum_{j=i+1}^{n}(a_{ij}\boldsymbol{XO}_j) + b_i\right],$$

Step 4 If $||\boldsymbol{x} - \boldsymbol{XO}|| < TOL$, then OUTPUT $(x_1, x_2, x_3, \ldots x_n)$;

STOP.

Step 5 Set $k = k + 1$.

Step 6 For for $i = 1,2,,\ldots n$

Set $\boldsymbol{XO}_i = x_i$.

Step 7 OUTPUT $(x_1, x_2, x_3, \ldots x_n)$;

STOP.

Gauss – Siedel Iteration method in matrix form

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1}(a_{ij}x_j^{(k)}) - \sum_{j=i+1}^{n}(a_{ij}x_j^{(k-1)}) + b_i\right], \qquad \text{for } i = 1,2,,\dots n$$

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

Namely,

$$a_{11}x_1^{(k)} = -a_{12}x_2^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1$$

$$a_{21}\underline{x_1^{(k)}} + a_{22}x_2^{(k)} = -a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2$$

$$\vdots$$

$$a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \cdots a_{nn}x_n^{(k)} = b_n$$

$A = D - L - U$

$Ax = b$

$(D-L-U)x = b$

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

Matrix form of Gauss-Seidel method.

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

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

Define $T_g = (D-L)^{-1}U$ and $\boldsymbol{c}_g = (D-L)^{-1}\boldsymbol{b}$, Gauss-Seidel method can be written as

$$\boldsymbol{x}^{(k)} = T_g\boldsymbol{x}^{(k-1)} + \boldsymbol{c}_g \qquad k = 1,2,3,\dots$$

$\underbrace{(D-L)}_{A} x = Ux + b$

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

$Ax = b$

$x = A^{-1}b$

Gauss – Siedel Iteration method in matrix form

$$5x_1 - 2x_2 + 3x_n = -1$$
$$-3x_1 + 9x_2 + x_n = 2$$
$$2x_1 - x_2 - 7x_n = 3$$

$5x_1 - 2x_2 + 3x_3 = -1$
$-3x_1 + 9x_2 + x_3 = 2$ using G. Siedel method.
$2x_1 - x_2 - 7x_3 = 3$
with $x^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

```
A=[5,-2,3;-3,9,1;2,-1,-7];
b=[-1;2;3];
n=size(A,1)
D=diag(diag(A));
L=tril(A,-1);
U=triu(A,1);
Tg=inv(D-L)*U
cg=inv(D-L)*b;
x0=[0;0;0];
x1=Tg*x0+cg
x2=Tg*x1+cg
x3=Tg*x2+cg
x4=Tg*x3+cg
x5=Tg*x4+cg
x6=Tg*x5+cg
```

$A = \begin{bmatrix} 5 & -2 & 3 \\ -3 & 9 & 1 \\ 2 & -1 & -1 \end{bmatrix};$

$D = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & -1 \end{bmatrix}; \quad L = \begin{bmatrix} 0 & 0 & 0 \\ 3 & 0 & 0 \\ -2 & 1 & 0 \end{bmatrix}; \quad U = \begin{bmatrix} 0 & 2 & -3 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$

$(D-L-U)x = b$
$(D-L)x = Ux + b$
$\Rightarrow x = (D-L)^{-1}Ux + (D-L)^{-1}b$

$$T_g = (D-L)^{-1}U \quad \text{and} \quad \boldsymbol{c}_g = (D-L)^{-1}\boldsymbol{b}$$
$$\boldsymbol{x}^{(k)} = T_g \boldsymbol{x}^{(k-1)} + \boldsymbol{c}_g$$

```
x1 = 3x1
   -0.2000
    0.1556
   -0.5079

x2 = 3x1
    0.1670
    0.3343
   -0.4286

x3 = 3x1
    0.1909
    0.3335
   -0.4217

x4 = 3x1
    0.1864
    0.3312
   -0.4226

x5 = 3x1
    0.1861
    0.3312
   -0.4227

x6 = 3x1
    0.1861
    0.3312
   -0.4227
```

equal upto 4 decimals

Gauss-Siedel gives the solution in lesser number of iterations than the Jacobi method. For this problem the accuracy obtained in the fourth iteration of Gauss Siedel was obtained only in the sixth iteration of Gauss-Jacobi

Gauss – Siedel Iteration method

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

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

A slight change in the previous code

```
xnew=xold;  %  the crucial step
for i=1:nRow
            xnew(i)=xnew(i)+ (b(i)-A(i,:)*xnew)/A(i,i);
End
xold=xnew
```

Gauss – Siedel Iteration method

Write the algorithm for the Gauss-Siedel iteration method in MATLAB as Homework

Convergence of Gauss–Jacobi and Gauss-Siedel Methods

- **These methods converge if the matrix A is strictly diagonally dominant.**

Diagonally dominant :-

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

A is strictly diagonally dominant if

$$|a_{11}| > |a_{12}| + |a_{13}|$$
$$|a_{22}| > |a_{21}| + |a_{23}|$$
$$|a_{33}| > |a_{31}| + |a_{32}|$$

Eg :-

$$\begin{bmatrix} 9 & 1 & 3 \\ 0 & 5 & 4 \\ 6 & 1 & 8 \end{bmatrix}, \quad \begin{bmatrix} -9 & 2 & -4 \\ 1 & 5 & -3 \\ -6 & 1 & 8 \end{bmatrix}$$

G.J & G.S

→ Use for square matrices

→ $a_{ii} \neq 0$

→ $|a_{ii}| > \sum_{j=1}^{n} |a_{ij}|$ (diagonal dominance)