VINIT KUMAR SINGH: 16PH20036

Assignment – 4

Gauss-Seidel method: Consider the set of algebraic linear equations,

$$a_{11}x_1+a_{12}x_2+...+a_{1n}x_n=b_1$$
$$a_{21}x_1+a_{22}x_2+...+a_{2n}x_n=b_2$$
$$a_{n1}x_1+a_{n2}x_2+...+a_{nn}x_n=b_n$$

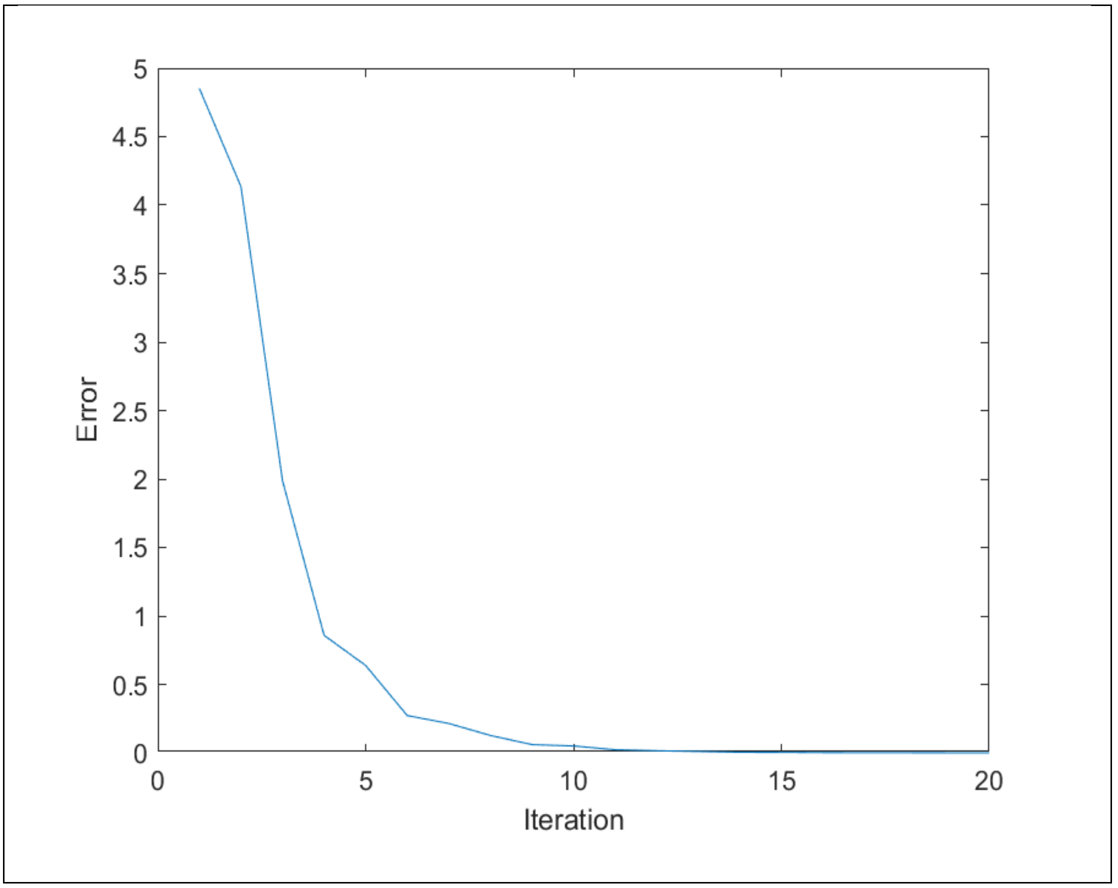
Where the coefficients and constants are given by
A= [-6 2 1 2 1; 3 8 -4 1 0; -1 1 4 10 1; 3 -4 1 9 2; 2 0 1 3 10]

And the coefficient matrix is given by *b* = [3; 4; -2; 12;1].
a) Write a code to see is the matrix *A* is diagonally dominant.
b) Write a code for solving this equation using Gauss-Seidel method in which
the convergence is achieved if error limit in successive iteration is within 0.001.

ANSWER:

```
%Function that checks whether a matrix is
Diagonally Dominant.
function [] = Diag_mat(A)
%A =[-6 2 1 2 1;3 8 -4 1 0;-1 1 4 10 1;3 -4 1 9 2;2
0 1 3 10];
%B=[18 3 6 -3;9 13 -5 2;-3 -2 4 9;6 0 11 3];
disp(A);
for i=1:size(A,1)
if abs(2*A(i,i))<sum(abs(A(i,:)))
fprintf('Not Diagonally dominant at row%d\n',i);
end
end
end
```

```matlab
%Gauss-Seidel method
clear;
clc;

A=[-6 2 1 2 1;3 8 -4 1 0;-1 1 4 10 1;3 -4 1 9 2;2 0
1 3 10];
B = [3; 4; -2; 12;1];
Diag_mat(A);
n=size(A,1);
D=zeros(n);
L=zeros(n);
U=zeros(n);
for i=1:n
    for j=1:n
        if i>j
            L(i,j)=A(i,j);
        elseif i<j
            U(i,j)=A(i,j);
        else
            D(i,j)=A(i,j);
        end
    end
end

X=zeros(5,1);
LD=L+D;

k=20;
ERR=zeros(1,k);
for i=1:k
    ERR(i)=norm(X-A\B);
    X=LD\(B-U*X);
    if ERR(i)<1e-6
       break;
    end
end
plot(ERR)
disp(i)
```



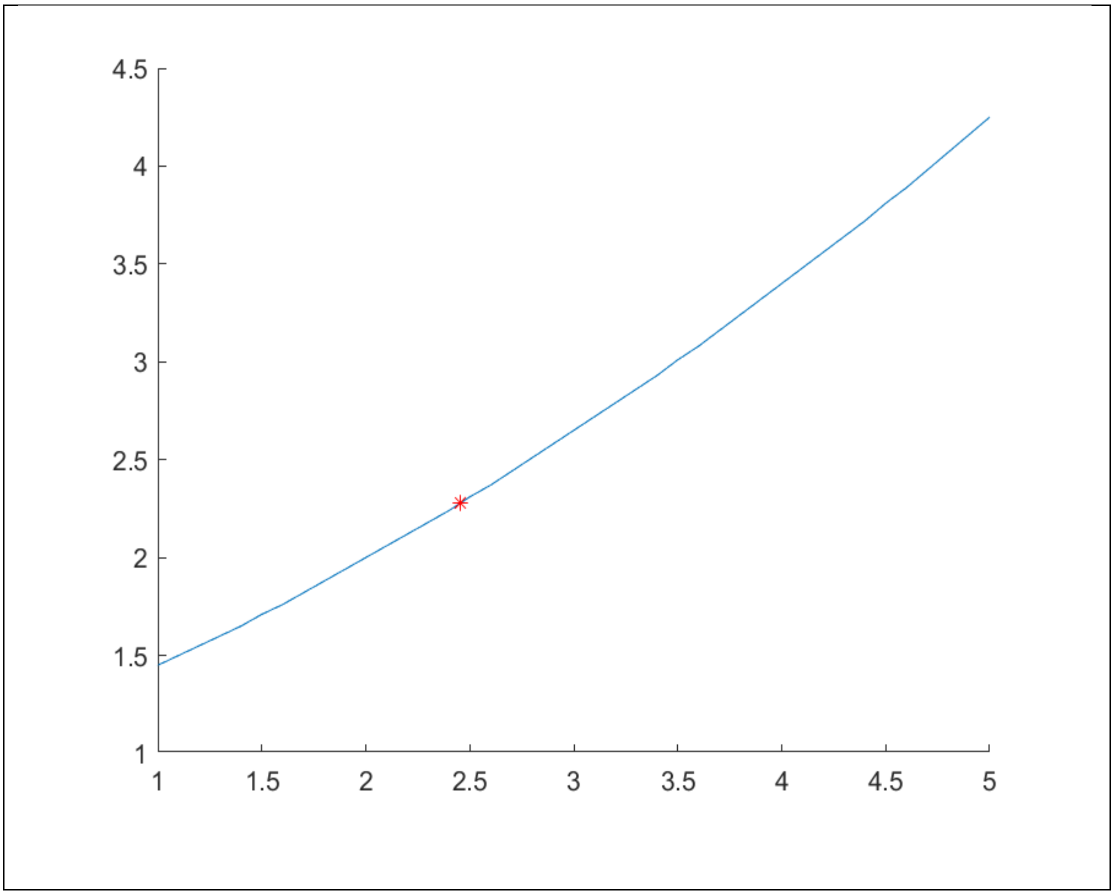
Linear interpolation: Write a code for two point segment linear interpolation
for the dataset given in file points.txt (attached)

ANSWER:

```
%Function to perform Linear Interpolation
function res = Linear(x)
A = importdata('points.txt');
if x<1 || x>5
    fprintf('Data Out of Bound\n');
else
X=A(:,1);
Y=A(:,2);
i=binary(x);
res=(Y(i+1)*(x-X(i))-Y(i)*(x-X(i+1)))/(X(i+1)-
X(i));
```

```
hold on
plot(X,Y)
scatter(x,res,'r*')
end
end


%This function searches for interval in which the
given point lies using Binary Search Algorithm
function i = binary(x)
A = importdata('points.txt');
X=A(:,1);
a=0;
b=length(X);
while b>a+1
    mid=int64((a+b)/2);
    if X(mid)<x && X(mid+1)>x
        i=mid;
        break;
    elseif X(mid)>x
        b=mid;
    else
        a=mid;
    end
    i=a;
    %fprintf('%d %d\n',a,b);
end
```

Input: x=2.456
Output: f(x)=2.2792



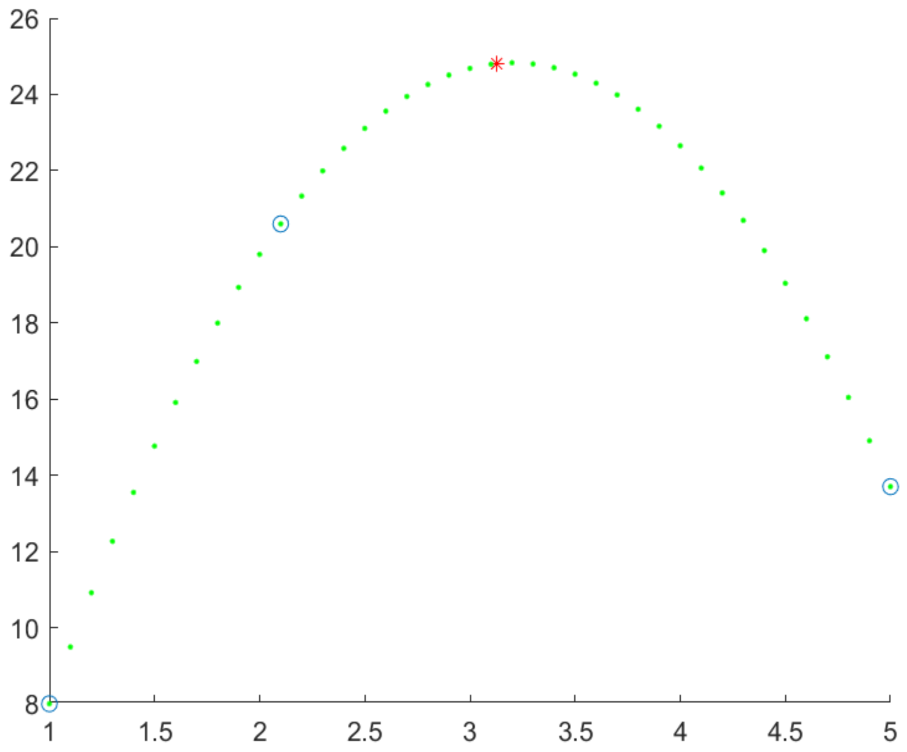
Polynomial interpolation 1: Given the three data points (x, y) = (1.0, 8.0), (2.1, 20.6) and (5.0, 13.7), write a program to return the value of y for any arbitrary x in the range [1.0, 5.0] using **second order polynomial**. Use Lagrange method of interpolation to construct the polynomial. **Plot the polynomial** along with the data points.

ANSWER:

```
function res = Poly(x)
A = [1,8;2.1,20.6;5,13.7];

X=A(:,1);
Y=A(:,2);

a=Y(1);
b=(Y(2)-Y(1))/(X(2)-X(1));
```

```
c=(Y(3)-Y(2))/(X(3)-X(2))/(X(3)-X(1)) - (Y(2)-
Y(1))/(X(3)-X(1))/(X(2)-X(1));
fprintf('%d %d %d\n',a,b,c);

res=a+b*(x-X(1))+c*(x-X(1))*(x-X(2));
hold on
scatter(X,Y)
scatter(x,res,'r*')
%a+b*(X-X(i))+c*(X-X).*(X-X(i+1));
P=1:0.1:5;
plot(P,a+b*(P-X(1))+c*(P-X(1)).*(P-X(2)),'g.')
hold off
end
```

Input: x=3.124
Output: f(x)=24.8074
Equation: 8 + 11.45*x -3.45*x^2



Instead if we use Quadratic Interpolation on the previously provided dataset.

```
function res = Quad(x)
A = importdata('points.txt');
if x<1 || x>5
    fprintf('Data Out of Bound\n');
else
X=A(:,1);
Y=A(:,2);
i=binary(x);

a=Y(i);
b=(Y(i+1)-Y(i))/(X(i+1)-X(i));
c=(Y(i+2)-Y(i+1))/(X(i+2)-X(i+1))/(X(i+2)-X(i)) -
(Y(i+1)-Y(i))/(X(i+2)-X(i))/(X(i+1)-X(i));
fprintf('%d %d %d\n',a,b,c);

res=a+b*(x-X(i))+c*(x-X(i))*(x-X(i+1));
hold on
plot(X,Y)
scatter(x,res,'r*')
%a+b*(X-X(i))+c*(X-X).*(X-X(i+1));
plot(X,a+b*(X-X(i))+c*(X-X(i)).*(X-X(i+1)),'g.')
hold off
end
end
```

Input: x=2.456
Output: f(x)=2.2804

Approximation of PI: Approximate the value of Pi using two random variables.

```
k=0;
n=1e7;
hold on
for i=1:n
    x=rand()*2-1;
    y=rand()*2-1;
    if x^2+y^2<1
        k=k+1;
    end
end
disp(4*k/n)
hold off
```

Result: 3.1417