**Shubhan Singh**

**2022300118**

**SE-Comps B/Batch C**

**29th February 2024**

**Scilab no.5: Gauss Jacobi Method**

**Program No.1** :- Solve using Gauss – Jacobi. Perform 7 iterations.

5x - 2y + 3z = -1

-3x + 9y + z = 2

2x – y - 7z = 3

**Code :-**

clc;

A=[5 -2 3;-3 9 1;2 -1 -7];

B=[-1; 2; 3];

n=7;

x=0;

y=0;

z=0;

for i=1:n

printf("\nIteration number %g",i);

X=(B(1)-A(1,2)\*y-A(1,3)\*z)/A(1,1);

Y=(B(2)-A(2,1)\*x-A(2,3)\*z)/A(2,2);

Z=(B(3)-A(3,1)\*x-A(3,2)\*y)/A(3,3);

printf("\nValue of x=%g",X);

printf("\nValue of y=%g",Y);

printf("\nValue of z=%g",Z);

x=X;

y=Y;

z=Z;

end

**Output :-**

**A screenshot of a computer

Description automatically generated**

**Program No. 2:-** Solve using Gauss – Jacobi. Perform 10 iterations.

10x - 2y – z - w = 3

-2x + 10y - z -w= 14

-x – y + 10z -2w = 27

-x - y -2z + 10w = -9

**Code**:-

clc;

A=[10 -2 -1 -1;-2 10 -1 -1;-1 -1 10 -2;-1 -1 -2 10];

B=[3;15;27;-9];

n=10;

x=0;

y=0;

z=0;

w=0;

for i=1:n

printf("\nIteration number %g",i);

X=(B(1)-A(1,2)\*y-A(1,3)\*z-A(1,4)\*w)/A(1,1);

Y=(B(2)-A(2,1)\*x-A(2,3)\*z-A(2,4)\*w)/A(2,2);

Z=(B(3)-A(3,1)\*x-A(3,2)\*y-A(3,4)\*w)/A(3,3);

W=(B(4)-A(4,1)\*x-A(4,2)\*y-A(4,3)\*z)/A(4,4);

printf("\nValue of x=%g",X);

printf("\nValue of y=%g",Y);

printf("\nValue of z=%g",Z);

printf("\nValue of w=%g",W);

x=X;

y=Y;

z=Z;

w=W

end

**Output** :-

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Program No. 3 :-** Solve using Gauss – Jacobi. Perform 10 iterations.

10x + y + z = 12

x + 10y + z = 12

x + y + 10z = 12

**Code** :-

clc;

A=[10 1 1;1 10 1;1 1 10];

B=[12; 12; 12];

n=10;

x=0;

y=0;

z=0;

for i=1:n

printf("\nIteration number %g",i);

X=(B(1)-A(1,2)\*y-A(1,3)\*z)/A(1,1);

Y=(B(2)-A(2,1)\*x-A(2,3)\*z)/A(2,2);

Z=(B(3)-A(3,1)\*x-A(3,2)\*y)/A(3,3);

printf("\nValue of x=%g",X);

printf("\nValue of y=%g",Y);

printf("\nValue of z=%g",Z);

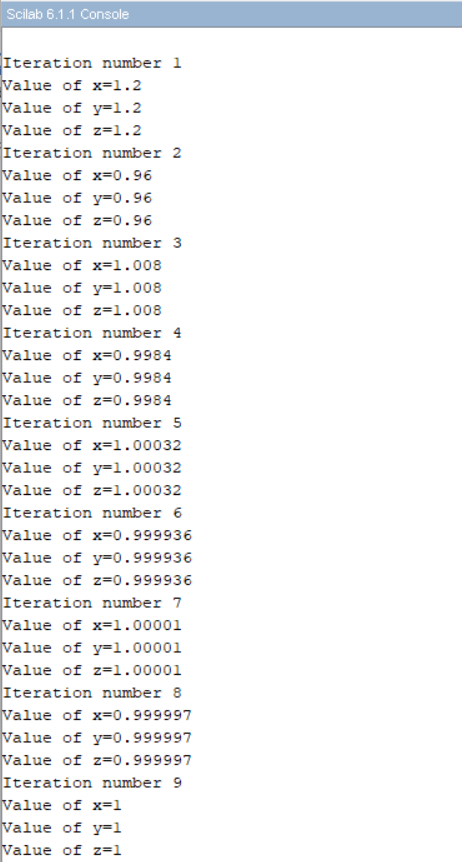
x=X;

y=Y;

z=Z;

end

**Output** :-



A screenshot of a computer

Description automatically generated

**Program no. 4** :- Solve using Gauss – Jacobi. Perform 10 iterations.

15x - 2y + 3z = 16

3x + 19y + y = 29

2x – y + 27z = 31

**Code** :-

clc;

A=[15 -2 3;3 19 1;2 -1 27];

B=[16;29;31];

n=10;

x=0;

y=0;

z=0;

for i=1:n

printf("\nIteration number %g",i);

X=(B(1)-A(1,2)\*y-A(1,3)\*z)/A(1,1);

Y=(B(2)-A(2,1)\*x-A(2,3)\*z)/A(2,2);

Z=(B(3)-A(3,1)\*x-A(3,2)\*y)/A(3,3);

printf("\nValue of x=%g",X);

printf("\nValue of y=%g",Y);

printf("\nValue of z=%g",Z);

x=X;

y=Y;

z=Z;

end

**Output** :-

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

**Program No. 5** :- Solve using Gauss – Jacobi. Perform 10 iterations.

110x + y + z = 13

4x + 140y + z = 14

6x + y + 210z = 15

**Code** :-

clc;

A=[110 1 1;4 140 1;6 1 210];

B=[13;14;15];

n=11;

x=0;

y=0;

z=0;

for i=1:n

printf("\nIteration number %g",i);

X=(B(1)-A(1,2)\*y-A(1,3)\*z)/A(1,1);

Y=(B(2)-A(2,1)\*x-A(2,3)\*z)/A(2,2);

Z=(B(3)-A(3,1)\*x-A(3,2)\*y)/A(3,3);

printf("\nValue of x=%g",X);

printf("\nValue of y=%g",Y);

printf("\nValue of z=%g",Z);

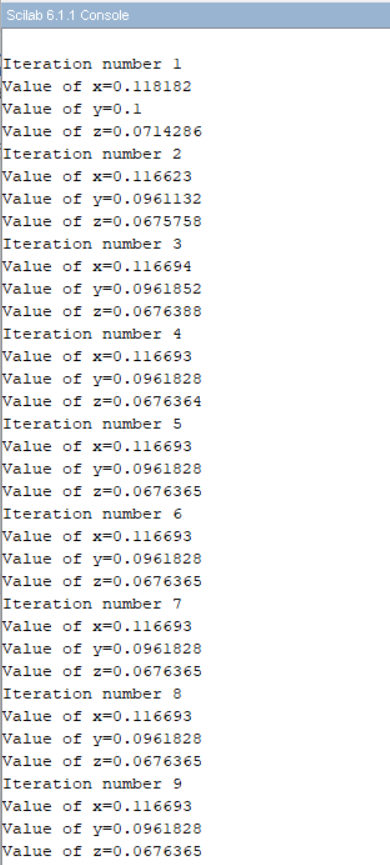
x=X;

y=Y;

z=Z;

end

**Output** :-



A screenshot of a computer

Description automatically generated