

Name : Deepankar Sharma
Course: BCA
University Roll No: 2092014
Student Id : 20041299
Semester: 3
Date: December 12, 2021

Practical 10: Guass Seidel's Iteration Method

Objective: To find root of the system equation using Guass Seidel's Iteration method.

2. Algorithm:

1. Start
2. Input the matrix of equations in $\text{arr}[n][n+1]$ where n is number of unknown variables
3. Check the validity of the equations, if valid we can apply iterative methods !!!
4. Matrix $X[n]$, for unknown variables.
5. Take $x_0=y_0=z_0=0$
6. Convert the variables as a function of other variables.
7. Apply iteration method in loop with immediate updation:
 $x=f(y_0,z_0)$
 $y=f(z_0,x)$
 $z=f(x,y)$
 $x=x_0$
 $y=y_0$
 $z=z_0$
if $((z-z_0)>\text{allowed Error AND } (y-y_0)>\text{allowed Error AND } (x-x_0)>\text{allowed Error})$
 Terminate Loop
8. Solution:
 for($i= n-1$; $i \geq 0$; $i--$) {
 $X[i] = (\text{arr}[i][n]) / \text{arr}[i][i];$
 }
9. Print X
10. Stop

Code:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

float arr[3][4],x[3];
int n=3;
/*
```

CBNST Page No.454

$$8x-3y+2z=20$$

$$4x+11y-z=33$$

$$6x+3y+12z=35$$

*/

```
#define x(y0, z0) (arr[0][3] - (arr[0][2]*z0 + arr[0][1]*y0)) / arr[0][0]
```

```
#define y(x0, z0) (arr[1][3] - (arr[1][2]*z0 + arr[1][0]*x0)) / arr[1][1]
```

```
#define z(y0, x0) (arr[2][3] - (arr[2][0]*x0 + arr[2][1]*y0)) / arr[2][2]
```

```
void checkValidity(){
    for (int i = 0; i < n; i++)
    {
        float sum=0;
        for(int j=0; j<n; j++){
            if(i!=j){
                sum+=fabs(arr[i][j]);
            }
        }
        if((fabs(arr[i][i])-sum)<0){
            printf("Can't apply iteration method !!!!\n\n");
            exit(1);
        }
    }
    printf("Can apply iteration method !!!!\n\n");
}
```

```
int main()
{

    printf("Enter coefficients of Augmented Matrix:\n");
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n + 1; j++)
        {
            printf("a[%d][%d] = ", i, j);
            scanf("%f", &arr[i][j]);
        }
    }
    checkValidity();

    int iter = 1;
    float allErr, fabx, faby, fabz, deltaX, deltaY, deltaZ;
    float x0 = 0.0, y0 = 0.0, z0 = 0.0;
    printf("Enter the allowed error: ");
```


a[1][1] = 11

a[1][2] = -1

a[1][3] = 33

a[2][0] = 6

a[2][1] = 3

a[2][2] = 12

a[2][3] = 35

Can apply iteration method !!!!

Enter the allowed error: 0.0001

Iteration	x0	y0	z0	x1	y1	z1
1	0.000000	0.000000	0.000000	2.500000	2.090909	1.143939
2	2.500000	2.090909	1.143939	2.998106	2.013774	0.914170
3	2.998106	2.013774	0.914170	3.026623	1.982516	0.907726
4	3.026623	1.982516	0.907726	3.016512	1.985607	0.912009
5	3.016512	1.985607	0.912009	3.016600	1.985964	0.911875
6	3.016600	1.985964	0.911875	3.016768	1.985891	0.911810
7	3.016768	1.985891	0.911810	3.016757	1.985889	0.911816

The roots are:

x= 3.016757

y= 1.985889

z= 0.911816