So I wrote this piece of code for solving a system of linear equations using Gauss-Seidel's Iterative method in the fifth semester of my undergraduate course for my Numerical Analysis Class. Hope you guys find it useful.

```
//Gaus-seidel (Written by: Manas Sharma - University of Delhi)
#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    cout.precision(4);
    cout.setf(ios::fixed);
    int n,i,j,k,flag=0,count=0;
    cout<<"\nEnter the no. of equations\n";
    cin>>n;                     //Input no. of equations
    double a[n][n+1];            //declare a 2d array for storing the elements of the
augmented matrix
    double x[n];                //declare an array to store the values of variables
    double eps,y;
    cout<<"\nEnter the elements of the augmented matrix row-wise:\n";
    for (i=0;i<n;i++)
        for (j=0;j<=n;j++)
            cin>>a[i][j];
    cout<<"\nEnter the initial values of the variables:\n";
    for (i=0;i<n;i++)
        cin>>x[i];
    cout<<"\nEnter the accuracy upto which you want the solution:\n";
    cin>>eps;
    for (i=0;i<n;i++)                    //Pivotisation(partial) to make the equations
diagonally dominant
        for (k=i+1;k<n;k++)
            if (abs(a[i][i])<abs(a[k][i]))
                for (j=0;j<=n;j++)
                {
                    double temp=a[i][j];
                    a[i][j]=a[k][j];
                    a[k][j]=temp;
                }
    cout<<"Iter"<<setw(10);
    for(i=0;i<n;i++)
        cout<<"x"<<i<<setw(18);
    cout<<"\n-----------------------------------------------------------------------";
    do                           //Perform iterations to calculate x1,x2,...xn
    {
        cout<<"\n"<<count+1<<"."<<setw(16);
        for (i=0;i<n;i++)                //Loop that calculates x1,x2,...xn
        {
            y=x[i];
            x[i]=a[i][n];
            for (j=0;j<n;j++)
            {
```

```
                if (j!=i)
                x[i]=x[i]-a[i][j]*x[j];
            }
            x[i]=x[i]/a[i][i];
            if (abs(x[i]-y)<=eps)            //Compare the ne value with the last value
                flag++;
            cout<<x[i]<<setw(18);
        }
        cout<<"\n";
        count++;
    }while(flag<n);                        //If the values of all the variables don't
differ from their previious values with error more than eps then flag must be n and
hence stop the loop
    cout<<"\n The solution is as follows:\n";
    for (i=0;i<n;i++)
        cout<<"x"<<i<<" = "<<x[i]<<endl;        //Print the contents of x[]
    return 0;
}
```

```
Enter the no. of equations
3

Enter the elements of the augmented matrix row-wise:
27      6       -1      85
6       15      2       72
1       1       54      110

Enter the initial values of the variables:
0       0       0

Enter the accuracy upto which you want the solution:
0.0001
Iter         x0                 x1                 x2
-------------------------------------------------------------
1.          3.1481            3.5407            1.9132

2.          2.4322            3.5720            1.9258

3.          2.4257            3.5729            1.9260

4.          2.4255            3.5730            1.9260

5.          2.4255            3.5730            1.9260

 The solution is as follows:
x0 = 2.4255
x1 = 3.5730
x2 = 1.9260
```

Sample Output

```
Enter the no. of equations
4

Enter the elements of the augmented matrix row-wise:
10      -2      -1      -1      3
-2      10      -1      -1      15
-1      -1      10      -2      27
-1      -1      -2      10      -9

Enter the initial values of the variables:
0       0       0       0

Enter the accuracy upto which you want the solution:
.001
Iter         x0                x1                x2                x3
-----------------------------------------------------------------
1.          0.3000            1.5600            2.8860           -0.1368

2.          0.8869            1.9523            2.9566           -0.0248

3.          0.9836            1.9899            2.9924           -0.0042

4.          0.9968            1.9982            2.9987           -0.0008

5.          0.9994            1.9997            2.9998           -0.0001

6.          0.9999            1.9999            3.0000           -0.0000

 The solution is as follows:
0.9999
1.9999
3.0000
-0.0000
```

Sample Output

More Resources:
More programs on Numerical Analysis:
http://www.bragitoff.com/2015/11/numerical-analysis-c-programs-for-various-techniques/
Lab Write-Up(with Algorithm and Flow-Chart):
http://www.bragitoff.com/2015/10/gauss-seidel-lab-write-up-with-algorithm-and-flowchart/
Video Explaining the above code:

http://equation-solver.org/
https://en.wikipedia.org/wiki/Equation

Manas Sharma

I’m a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time to time. Can code in most of the popular languages. I like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

manas.bragitoff.com/

Share this:

Click to share on Facebook (Opens in new window)

Click to share on Twitter (Opens in new window)

Click to share on WhatsApp (Opens in new window)

Click to share on Pinterest (Opens in new window)

Click to share on Reddit (Opens in new window)

Click to share on LinkedIn (Opens in new window)

Click to email a link to a friend (Opens in new window)

Click to print (Opens in new window)

Click to share on Tumblr (Opens in new window)

Click to share on Pocket (Opens in new window)

Click to share on Telegram (Opens in new window)

[wpedon id="7041" align="center"]