### **Gauss Seidel**

# **Algorithm**

```
Begin
Take the dimensions of the matrix n and its elements as input.
Take the initials values of x and no of iteration iter as input.
While iter>0
Make a for loop i = 0 to n-1
initialize x[i] = (a[i][n] / a[i][i]).
Make a for loop i = 0 to n-1
If (j == i)
x[i] = x[i] - ((a[i][j] / a[i][i]) * m[j]).
m[i] = x[i].

Decrease iter.

/*
Here, a[i][j] = input matrix.
m[i] = stores initial values of x.
*/
Return 0
End
```

### **Code**

```
c gauss_seidelh x

using namespace std;

class GaussSeidel

public:
    /**
    @param float **a the augmented matrix
    @param n array size

    */
    void printMatrix(float **a, int n);
    /**
    @param float **a the augmented matrix
    @param float **a the augmented matrix
    @param float **a the augmented matrix
    @param n array size
    @param float **x the array of solutions
    @param iter is the number of iterations
    */
    void solve(float **a, int n, float *x, int iter);
    /**
    Checks solution validity
    */
    bool valid_solution = false;
    protected:
    private:
    };
};
```

```
G gauss_seidel.cpp
```

```
,
```

```
#include <iostream>
#include <iomanip>
#include <algorithm>
#include "gauss_seidel.h"
using namespace std;
void GaussSeidel::printMatrix(float **a, int n)
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j \le n; j++)
            std::cout << a[i][j] << std::setw(16);</pre>
        std::cout << "\n";
    };
    std::cout << "\n";
void GaussSeidel::solve(float **a, int n, float *x, int iter)
    float m[n], tolerance;
    while (iter> 0 && iter >= 30) {
        for (int i = 0; i < n; i++){}
            x[i] = a[i][n] / a[i][i];
            for (int j = 0; j < n; j++){}
                if (j == i)
                x[i] = x[i] - ((a[i][j] / a[i][i]) * m[j]);
                m[i] = x[i];
            }
            cout<<"x"<<i + 1 << "="<<x[i]<<" ";
        cout << "\n";
        iter--;
```

```
← main.cpp

        #include <iostream>
        #include <vector>
        #include "gauss_elimination.h"
       using namespace std;
        int main()
            int iDim;
            std::vector<float> _scale;
            std::cout.precision(4);
            std::cout.setf(ios::fixed); //display using the fixed precision
            std::cout \ll "\nEnter the no. of equations\n";
            std::cin >> iDim;
            float *p[iDim];
            float arrA[iDim][iDim+1]; //Array declaration to store the augmented-matrix elements
            std::cout << "\nEnter the elements of the augmented-matrix row-wise:\n"; for (int i = 0; i < iDim; i++)
                for (int j = 0; j \le iDim; j++)
                    std::cout << "A[" << i << "][" << j << "]=";
                    std::cin >> arrA[i][j];
            for (int i = 0; i < iDim; ++i)</pre>
                p[i] = arrA[i];
            GaussElimination matrix = GaussElimination();
            std::cout << "\nThis is the matrix to be solved\n";</pre>
            matrix.printMatrix(p, iDim);
            matrix.selectMax(p, iDim, _scale);
            for (int step = 0; step < iDim-1; step++)</pre>
                matrix.solve(p, iDim,_scale, step);
            matrix.backwardSubstitution(p, iDim);
```

# **Output**

#### Gauss-Seidel Example



 Use the Gauss-Seidel method to obtain the solution to the system of equations given below

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

Recall that the true solution is  $x_1 = 3$ ,  $x_2 = -2.5$ , and  $x_3 = 7$ .

```
aa-ali:~/Gauss Seidel$ ./gauss_seidel
 Enter the no. of equations
Enter the elements of the augmented-matrix row-wise: A[0][0]=3 A[0][1]=-0.1 A[0][2]=-0.2
 A[0][3]=7.85
A[1][0]=0.1
A[1][1]=7
A[1][2]=-0.3
 A[1][2]=-0.3
A[1][3]=-19.3
A[2][0]=0.3
A[2][1]=-0.2
A[2][2]=10
A[2][3]=71.4
                                                                         -0.2000
-0.3000
10.0000
                                  -0.1000
7.0000
                                                                                                                  7.8500
 enter the initial values of the variables:
 Enter the no. of iterations
 x1=1035719978759872321969464890217201664.0000 x2=651023911948223660360145342795087872.0000 x3=-18051124193922993879624121600442368.0000
x1=20497390270873969562486233198755840.0000 x2=-1066439462691649092419157070184448.0000 x3=-636250502022327216043146168238080.0000 x1=-77964685125965417891958332850176.0000 x2=-26154099445377304364778063396864.0000 x3=1815858715987143901291835555840.0000
x1=-7/90406312390341/8919363328501/0.0000 x2=-2013409944357/3043047/8003390804.0000 x3=16136387/1398/143961291835353
x1=-750746107410961205894118899712.0000 x2=85547459981485644504769757184.0000 x3=24293333649773834613026717696.0000
x1=4571137713770132874725425152.0000 x2=975840878066627712689111040.0000 x3=-117617325657687639162093568.0000
x1=24686876377161889744945152.0000 x2=-5393412338950262124183552.0000 x3=-848474566916899550003200.0000
x1=-236345395636694281093120.0000 x2=-32986833895257125421056.0000 x3=6430625168677687787520.0000
 x1=-670852822793388556288.0000 x2=285181843824260939776.0000 x3=25829223583388270592.0000
x1=11228010432709001216.0000 x2=946566536695382016.0000 x3=-317909012483932160.0000
 (1=10358284296912896.0000 x2=-13772648005763072.0000 x3=-586201497075712.0000 
(1=-498168391270400.0000 x2=-18006230958080.0000 x3=14584927748096.0000
x1=-498168391270400.0000 x2=-18006230958080.0000 x3=14584927;
x1=372120879104.0000 x2=619752325120.0000 x3=1231419392.0000
x1=20740505600.0000 x2=-243517808.0000 x3=-627085504.0000
x1=-49922960.0000 x2=-26161912.0000 x3=974457.7500
x1=-807097.2500 x2=53289.6797 x3=25285.8516
x1=3464.6631 x2=1031.4270 x3=-76.1714
x1=31.9195 x2=-6.4776 x3=6.0529
x1=2.8043 x2=-2.5378 x3=7.0051
x1=2.9991 x2=-2.4998 x3=7.0000
x1=3.0000 x2=-2.5000 x3=7.0000
x1=3.0000 x3=-2.5000 x3=7.0000
 (1=3.0000 x2=-2.5000 x3=7.0000
(1=3.0000 x2=-2.5000 x3=7.0000
  1=3.0000 x2=-2.5000 x3=7.0000
1=3.0000 x2=-2.5000 x3=7.0000
     =3.0000 x2=-2.5000 x3=7.0000
  1=3.0000 x2=-2.5000
  1=3.0000 x2=-2.5000 x3=7.0000
1=3.0000 x2=-2.5000 x3=7.0000
   1=3.0000 x2=-2.5000 x3=7.0000
1=3.0000 x2=-2.5000 x3=7.0000
```

Test your implementation by solving the following system:

```
2x1 + x2 - x3 = 0

x1 + 4x2 + 3x3 = 14

-x1 + 2x2 + 7x3 = 30
```

```
smaa@asmaa-ali:~/Gauss Seidel$ g++ -o gauss_seidel gauss_seidel.cpp main.cpp
 smaa@asmaa-ali:~/Gauss Seidel$ ./gauss_seidel
Enter the no. of equations
Enter the elements of the augmented-matrix row-wise:
A[0][0]=2
A[0][1]=1
A[0][2]=-1
A[0][3]=0
A[1][0]=1
A[1][1]=4
A[1][2]=3
A[1][3]=14
A[2][0]=-1
A[2][1]=2
A[2][2]=7
A[2][3]=30
The matrix to be solved
2.0000
                1.0000
                                -1.0000
                                                 0.0000
                                3.0000
 .0000
                4.0000
                                                14.0000
 1.0000
                 2.0000
                                 7.0000
                                                 30.0000
Enter the initial values of the variables:
X[0] = 0
X[1] = 0
X[2] = 0
Enter the no. of iterations
x1=80963497609274170424688640.0000 x2=-141686120816229798243205120.0000 x3=52047963407631397905498112.0000
x1=96867044417773607288045568.0000 x2=-63252732507245445644288000.0000 x3=31910360796166509010354176.0000
x1=47581545498784472720474112.0000 x2=-35828156395360247634460672.0000 x3=17033981732080431895937024.0000
x1=26431069063720339765198848.0000 x2=-19383252988529656559828992.0000 x3=9313939785649166672199680.0000
x1=14348596963550163919437824.0000 x2=-10572603791894039832297472.0000 x3=5070543589114142269636608.0000
x1=7821573978734467202678784.0000 x2=-5758301042404035427041280.0000 x3=2762596786384916973617152.0000
x1=4260449058509664276185088.0000 x2=-3137059638243321685475328.0000 x3=1504938415922437135794176.0000
x1=2320999027082879410634752.0000 x2=-1708953640770141742432256.0000 x3=819843778962621567336448.0000
x1=1264398637808787616956416.0000 x2=-930982529702960098705408.0000 x3=446623416198212846878720.0000
x1=688803008979383491756032.0000 x2=-507168332407904017580032.0000 x3=243305672831998363697152.0000
x1=375237002619951190638592.0000 x2=-276288523293385079914496.0000 x3=132544864172388764352512.0000
x1=204416693732886922133504.0000 x2=-150512821562513303797760.0000 x3=72206050696044577030144.0000
x1=111359436129278940413952.0000 x2=-81994399306152981561344.0000 x3=39335465966168952537088.0000
x1=60664932636160967049216.0000 x2=-44667831507767049322496.0000 x3=21428658451784849686528.0000
x1=33048244979775949504512.0000 x2=-24333554520832671219712.0000 x3=11673622968120104714240.0000
x1=18003588744476387966976.0000 x2=-13256114693684152238080.0000 x3=6359402429420683657216.0000
x1=9807759124502371368960.0000 x2=-7221491603191105585152.0000 x3=3464391761554940362752.0000
x1=5342941682373022973952.0000 x2=-3934029382496949370880.0000 x3=1887285778195274530816.0000
x1=2910657580346111950848.0000 x2=-2143128799101728063488.0000 x3=1028130840319572836352.0000
x1=1585629890079394627584.0000 x2=-1167505637943900372992.0000 x3=560091610217187377152.0000
x1=863798641672729919488.0000 x2=-636018394469352079360.0000 x3=305119367335560347648.0000
x1=470568880902456213504.0000 x2=-346481763319470358528.0000 x3=166218925415734444032.0000
x1=256350344367602401280.0000 x2=-188751771357608411136.0000 x3=90550561580469190656.0000
x1=139651170867085312000.0000 x2=-102825711703099965440.0000 x3=49328947065379618816.0000
x1=76077329384239792128.0000 x2=-56016044844117917696.0000 x3=26872776666380238848.0000
x1=41444410755249078272.0000 x2=-30515685188597448704.0000 x3=14639398090000302080.0000
x1=22577541089543061504.0000 x2=-16623934114763898880.0000 x3=7975059102445338624.0000
x1=12299496608604618752.0000 x2=-9056168204107251712.0000 x3=4344547848709210112.0000
x1=6700357888969277440.0000 x2=-4933500427493703680.0000 x3=2366765574119227392.0000
x1=3650132863367512064.0000 x2=-2687607190272868352.0000 x3=1289335438363852800.0000
   aa@asmaa-ali:~/Gauss SeidelS
```