

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

```
1  gauss_seidel.h
2  using namespace std;
3
4  class GaussSeidel
5  {
6  public:
7      /**
8       * @param float **a the augmented matrix
9       * @param n array size
10      */
11     void printMatrix(float **a, int n);
12     /**
13      * @param float **a the augmented matrix
14      * @param n array size
15      * @param float **x the array of solutions
16      * @param iter is the number of iterations
17      */
18     void solve(float **a, int n, float *x, int iter);
19     /**
20      * Checks solution validity
21      */
22     bool valid_solution = false;
23 protected:
24
25 private:
26
27 };
```

```
1 //Gaus-seidel (Written by: Asmaa Ali - Nile University)
2 #include <iostream>
3 #include <iomanip>
4 #include <algorithm>
5 #include "gauss_seidel.h"
6
7 using namespace std;
8
9 void GaussSeidel::printMatrix(float **a, int n)
10 {
11     for (int i = 0; i < n; i++) //print the new matrix
12     {
13         for (int j = 0; j <= n; j++)
14             std::cout << a[i][j] << std::setw(16);
15         std::cout << "\n";
16     };
17     std::cout << "\n";
18 }
19
20 void GaussSeidel::solve(float **a, int n, float *x, int iter)
21 {
22     float m[n], tolerance;
23     while (iter > 0 && iter >= 30) {
24         for (int i = 0; i < n; i++){
25             x[i] = a[i][n] / a[i][i];
26             for (int j = 0; j < n; j++){
27                 if (j == i)
28                     continue;
29                 x[i] = x[i] - ((a[i][j] / a[i][i]) * m[j]);
30                 m[i] = x[i];
31             }
32             cout<<"x"<<i + 1 << "="<<x[i]<<" ";
33         }
34         cout << "\n";
35         iter--;
36     }
37 }
```

```
1  #include <iostream>
2  #include <vector>
3  #include "gauss_elimination.h"
4
5  using namespace std;
6
7  int main()
8  {
9      int iDim;
10     std::vector<float> _scale;
11     std::cout.precision(4);           //set precision
12     std::cout.setf(ios::fixed);       //display using the fixed precision
13     std::cout << "\nEnter the no. of equations\n";
14     std::cin >> iDim;                 //input the no. of equations
15
16     float *p[iDim];
17     float arrA[iDim][iDim+1]; //Array declaration to store the augmented-matrix elements
18     std::cout << "\nEnter the elements of the augmented-matrix row-wise:\n";
19     for (int i = 0; i < iDim; i++)
20         for (int j = 0; j <= iDim; j++)
21         {
22             std::cout << "A[" << i << "][" << j << "]=";
23             std::cin >> arrA[i][j];
24         };
25     for (int i = 0; i < iDim; ++i)
26         p[i] = arrA[i];
27
28     GaussElimination matrix = GaussElimination();
29     std::cout << "\nThis is the matrix to be solved\n";
30     matrix.printMatrix(p, iDim);
31     matrix.selectMax(p, iDim, _scale);
32     for (int step = 0; step < iDim-1; step++)
33         matrix.solve(p, iDim, _scale, step);
34     matrix.backwardSubstitution(p, iDim);
35     return 0;
36 }
```

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 $x_3 = 7$.

[illegible]

Test your implementation by solving the following system:

$$\begin{aligned}2x_1 + x_2 - x_3 &= 0 \\ x_1 + 4x_2 + 3x_3 &= 14 \\ -x_1 + 2x_2 + 7x_3 &= 30\end{aligned}$$

```
asmaa@asmaa-ali:~/Gauss Seidel$ g++ -o gauss_seidel gauss_seidel.cpp main.cpp
asmaa@asmaa-ali:~/Gauss Seidel$ ./gauss_seidel

Enter the no. of equations
3

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
1.0000      4.0000      3.0000      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
30
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=6700357888969277440.0000 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
asmaa@asmaa-ali:~/Gauss Seidel$
```