

Lab 09 - Solution of Linear Algebraic Equation Using Gauss Siedel and Gauss Jacobbi Method

Objectives

1. Find solution of linear algebraic equation using Gauss Siedel method.
 2. Find solution of linear algebraic equation using Gauss Jacobi method.
-

Theory

The solution of linear algebraic equations is fundamental in numerical methods, especially for systems that cannot be solved analytically. Two popular iterative techniques for solving such systems are the Gauss-Seidel and Gauss-Jacobi methods.

Gauss-Jacobi Method

The Gauss-Jacobi method is an iterative algorithm that solves each equation for a particular variable, assuming the other variables remain constant at their previous iteration values. The process is repeated until the solution converges within a desired tolerance. This method is simple to implement but may converge slowly, especially for large systems.

Gauss-Seidel Method

The Gauss-Seidel method improves upon the Gauss-Jacobi method by using the most recently updated values of the variables as soon as they are available within the same iteration. This often leads to faster convergence compared to the Gauss-Jacobi method. However, both methods require the coefficient matrix to be diagonally dominant or symmetric positive definite for guaranteed convergence.

Convergence Criteria

Both methods iterate until the difference between successive approximations is less than a specified tolerance. The choice of initial guesses and the properties of the coefficient matrix significantly affect the convergence rate.

These iterative methods are particularly useful for large, sparse systems where direct methods like Gaussian elimination are computationally expensive.

Algorithm

Gauss-Jacobi Method Algorithm

1. Start with an initial guess for all variables.
2. For each variable, solve its equation for the current variable using the previous iteration values for all other variables.
3. Update all variables simultaneously after each iteration.
4. Repeat steps 2–3 until the maximum difference between the current and previous values of all variables is less than the specified tolerance.

Gauss-Seidel Method Algorithm

1. Start with an initial guess for all variables.
 2. For each variable, solve its equation for the current variable using the most recently updated values for the other variables (use new values as soon as they are available).
 3. Update each variable immediately after it is computed.
 4. Repeat steps 2–3 until the maximum difference between the current and previous values of all variables is less than the specified tolerance.
-

Source Code

```
/*
Solution of Linear Algebraic Equation using Gauss Jacobi and Gauss Seidel Method

Equations:
 $4x_1 + x_2 + 2x_3 = 4$ 
 $3x_1 + 5x_2 + x_3 = 7$ 
 $x_1 + x_2 + 3x_3 = 3$ 
*/

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

// Maximum number of iterations
#define MAX_ITR 100
// Tolerance for convergence
#define tol 0.001

// Functions representing rearranged equations for iterative methods
#define f1(x1, x2, x3) (4 - x2 - 2 * x3) / 4
#define f2(x1, x2, x3) (7 - 3 * x1 - x3) / 5
#define f3(x1, x2, x3) (3 - x1 - x2) / 3

// Function to solve using Gauss-Jacobi Method
void Gauss_jacobi(float x10, float x20, float x30)
{
    float x1, x2, x3;
    int itr = 0;

    printf("\n--- Gauss-Jacobi Method ---\n");
    printf("Iteration\t x1\t\t x2\t\t x3\n");

    // Iterative process
    do
    {
        // Calculate new values using previous iteration values
        x1 = f1(x10, x20, x30);
        x2 = f2(x10, x20, x30);
        x3 = f3(x10, x20, x30);

        // Print current iteration results
        printf("%d\t\t %.4f\t %.4f\t %.4f\n", itr + 1, x1, x2, x3);

        // Check for convergence
```

```

        if (fabs(x1 - x10) < tol && fabs(x2 - x20) < tol && fabs(x3 - x30) < tol)
            break;

        // Update values for next iteration
        x10 = x1;
        x20 = x2;
        x30 = x3;
        itr++;
    } while (itr < MAX_ITR);
}

// Function to solve using Gauss-Seidel Method
void Gauss_seidel(float x10, float x20, float x30)
{
    float x1, x2, x3;
    int itr = 0;

    printf("\n--- Gauss-Seidel Method ---\n");
    printf("Iteration\t x1\t\t x2\t\t x3\n");

    // Iterative process
    do
    {
        // Calculate new values using the most recent updates
        x1 = f1(x10, x20, x30);
        x2 = f2(x1, x20, x30);    // Use updated x1
        x3 = f3(x1, x2, x30);    // Use updated x1 and x2

        // Print current iteration results
        printf("%d\t\t %.4f\t %.4f\t %.4f\n", itr + 1, x1, x2, x3);

        // Check for convergence
        if (fabs(x1 - x10) < tol && fabs(x2 - x20) < tol && fabs(x3 - x30) < tol)
            break;

        // Update values for next iteration
        x10 = x1;
        x20 = x2;
        x30 = x3;
        itr++;
    } while (itr < MAX_ITR);
}

int main()
{
    float x10, x20, x30;

```

```
// Input initial guesses from user
printf("\nEnter the initial guesses: ");
scanf("%f%f%f", &x10, &x20, &x30);

// Call Gauss-Jacobi method
Gauss_jacobi(x10, x20, x30);

// Call Gauss-Seidel method
Gauss_seidel(x10, x20, x30);

return 0;
}
```

Output

Enter the initial guesses: 0 0 0

--- Gauss-Jacobi Method ---

Iteration	x1	x2	x3
1	1.0000	1.4000	1.0000
2	0.1500	0.6000	0.2000
3	0.7500	1.2700	0.7500
4	0.3075	0.8000	0.3267
5	0.6367	1.1502	0.6308
6	0.3970	0.8918	0.4044
7	0.5748	1.0809	0.5704
8	0.4446	0.9410	0.4481
9	0.5407	1.0436	0.5381
10	0.4700	0.9680	0.4719
11	0.5221	1.0236	0.5207
12	0.4838	0.9826	0.4848
13	0.5120	1.0128	0.5112
14	0.4912	0.9906	0.4918
15	0.5065	1.0069	0.5061
16	0.4952	0.9949	0.4955
17	0.5035	1.0038	0.5033
18	0.4974	0.9972	0.4976
19	0.5019	1.0020	0.5018
20	0.4986	0.9985	0.4987
21	0.5010	1.0011	0.5010
22	0.4992	0.9992	0.4993
23	0.5006	1.0006	0.5005
24	0.4996	0.9996	0.4996
25	0.5003	1.0003	0.5003

--- Gauss-Seidel Method ---

Iteration	x1	x2	x3
1	1.0000	0.8000	0.4000
2	0.6000	0.9600	0.4800
3	0.5200	0.9920	0.4960
4	0.5040	0.9984	0.4992
5	0.5008	0.9997	0.4998
6	0.5002	0.9999	0.5000

Discussion

The results demonstrate the effectiveness of both Gauss-Jacobi and Gauss-Seidel methods for solving systems of linear algebraic equations. As observed in the output, the Gauss-Seidel method converges faster than the Gauss-Jacobi method for the given system. This is because Gauss-Seidel utilizes the most recent values within each iteration, accelerating convergence.

However, the convergence of both methods depends on the properties of the coefficient matrix. For matrices that are not diagonally dominant or symmetric positive definite, these methods may fail to converge or may converge very slowly. The choice of initial guesses can also influence the number of iterations required.

In practical applications, iterative methods like these are preferred for large and sparse systems where direct methods become computationally expensive. Understanding the convergence behavior and limitations of each method is crucial for selecting the appropriate technique for a given problem.

Conclusion

The Gauss-Jacobi and Gauss-Seidel methods are valuable iterative techniques for solving systems of linear algebraic equations, especially when dealing with large or sparse matrices. While both methods can achieve accurate solutions, the Gauss-Seidel method typically converges more rapidly due to its use of updated variable values within each iteration. It is important to ensure that the coefficient matrix meets the necessary conditions for convergence, such as diagonal dominance or positive definiteness. By understanding the strengths and limitations of these methods, practitioners can effectively apply them to a wide range of numerical problems in engineering and science.

Name: Kushal Prasad Joshi

Roll No: 230345

Faculty: Science and Technology

Program: BE Computer

Group: B