

- [Tutorials ▾](#)
 - [Python Programming](#)
 - [C Programming](#)
 - [Numerical Methods](#)
 - [Dart Language](#)
 - [Computer Basics](#)
 - [Flutter](#)
 - [Linux](#)
 - [Deep Learning](#)
- [Examples ▾](#)
 - [C Programming Examples](#)
 - [Python Programming Examples](#)
- [Online Calculator](#)
- [Tutorials ▾](#)
 - [Python Programming](#)
 - [C Programming](#)
 - [Numerical Methods](#)
 - [Dart Language](#)
 - [Computer Basics](#)
 - [Flutter](#)
 - [Linux](#)
 - [Deep Learning](#)
- [Examples ▾](#)
 - [C Programming Examples](#)
 - [Python Programming Examples](#)
- [Online Calculator](#)
- [Algorithm for Bisection Method](#)
- [Pseudocode for Bisection Method](#)
- [C Program for Bisection Method](#)
- [C++ Program for Bisection Method](#)
- [MATLAB Program for Bisection Method](#)
- [Python Program for Bisection Method](#)
- [Bisection Method Advantages](#)
- [Bisection Method Disadvantages](#)
- [Bisection Method Features](#)
- [Convergence of Bisection Method](#)
- [Bisection Method Online Calculator](#)
- [Algorithm for Regula Falsi \(False Position Method\)](#)
- [Pseudocode for Regula Falsi \(False Position\) Method](#)
- [Features of Regula Falsi](#)
- [Falsi Position Advantages](#)
- [False Position Disadvantages](#)
- [C Program for Regula False \(False Position\) Method](#)
- [C++ Program for Regula False \(False Position\) Method](#)
- [MATLAB Program for Regula False \(False Position\) Method](#)

Ads by Google

Stop seeing this ad

Why this ad? ⓘ

- [Newton Raphson Method C++ Program](#)
- [Newton Raphson Method Python Program](#)
- [Newton-Raphson MATLAB](#)
- [Features of Newton Raphson Method](#)
- [Newton Raphson Advantages](#)
- [Newton Raphson Disadvantages](#)
- [Newton Raphson Method Online Calculator](#)
- [Secant Method Algorithm](#)
- [Secant Method Pseudocode](#)
- [Secant Method C Program](#)
- [Secant Method C++ Program with Output](#)
- [Secant Method Python Program with Output](#)
- [Secant Method Online Calculator](#)
- [Fixed Point Iteration \(Iterative\) Method Algorithm](#)
- [Fixed Point Iteration \(Iterative\) Method Pseudocode](#)
- [Fixed Point Iteration \(Iterative\) Method C Program](#)
- [Fixed Point Iteration \(Iterative\) Python Program](#)
- [Fixed Point Iteration \(Iterative\) Method C++ Program](#)
- [Fixed Point Iteration \(Iterative\) Method Online Calculator](#)
- [Gauss Elimination Method Algorithm](#)
- [Gauss Elimination Method Pseudocode](#)
- [Gauss Elimination C Program](#)
- [Gauss Elimination C++ Program with Output](#)
- [Gauss Elimination Method Python Program with Output](#)
- [Gauss Elimination Method Online Calculator](#)
- [Gauss Jordan Method Algorithm](#)
- [Gauss Jordan Method Pseudocode](#)
- [Gauss Jordan Method C Program](#)
- [Gauss Jordan Method C++ Program](#)
- [Gauss Jordan Method Python Program \(With Output\)](#)
- [Gauss Jordan Method Online Calculator](#)
- [Matrix Inverse Using Gauss Jordan Method Algorithm](#)
- [Matrix Inverse Using Gauss Jordan Method Pseudocode](#)
- [Matrix Inverse Using Gauss Jordan C Program](#)
- [Matrix Inverse Using Gauss Jordan C++ Program](#)
- [Python Program to Inverse Matrix Using Gauss Jordan](#)
- [Matrix Inverse Online Calculator](#)
- [Power Method \(Largest Eigen Value and Vector\) Algorithm](#)
- [Power Method \(Largest Eigen Value and Vector\) Pseudocode](#)
- [Power Method \(Largest Eigen Value and Vector\) C Program](#)
- [Power Method \(Largest Eigen Value and Vector\) C++ Program](#)
- [Power Method \(Largest Eigen Value & Vector\) Python Program](#)
- [Jacobi Iteration Method Algorithm](#)
- [Jacobi Iteration Method C Program](#)
- [Jacobi Iteration Method C++ Program with Output](#)
- [Python Program for Jacobi Iteration](#)
- [Gauss Seidel Iteration Method Algorithm](#)
- [Gauss Seidel Iteration Method C Program](#)
- [Gauss Seidel Iteration Method C++ Program](#)
- [Python Program for Gauss Seidel Iteration Method](#)
- [Python Program for Successive Over Relaxation](#)
- [Forward Difference Table Using C](#)
- [Forward Difference Table Using C++](#)
- [Python Program to Generate Forward Difference Table](#)
- [Python Program to Generate Backward Difference Table](#)
- [Backward Difference Table Using C](#)
- [Backward Difference Table Using C++](#)
- [Lagrange Interpolation Method Algorithm](#)
- [Lagrange Interpolation Method Pseudocode](#)
- [Lagrange Interpolation Method C Program](#)
- [Lagrange Interpolation Method C++ Program](#)
- [Lagrange Interpolation in Python](#)
- [Linear Interpolation Method Algorithm](#)
- [Linear Interpolation Method Pseudocode](#)
- [Linear Interpolation Method C Program](#)
- [Linear Interpolation Method C++ Program with Output](#)
- [Linear Interpolation Method Python Program](#)
- [Linear Regression Method Algorithm](#)
- [Linear Regression Method Pseudocode](#)
- [Linear Regression Method C Program](#)
- [Linear Regression Method C++ Program with Output](#)
- [Linear Regression Python](#)
- [Curve Fitting of Type \$y=ax^b\$ Algorithm](#)
- [Curve Fitting of Type \$y=ax^b\$ Pseudocode](#)
- [Curve Fitting \$y=ax^b\$ C Program](#)
- [Curve Fitting \$y=ax^b\$ Python Program](#)


Ads by Google

Stop seeing this ad

Why this ad? ⓘ

- [Curve Fitting \$y = ab^x\$ C Program](#)
- [Curve Fitting \$y = ab^x\$ C++ Program](#)
- [Curve Fitting \$y = ab^x\$ Python Program](#)
- [Derivative Using Forward Difference Formula Algorithm](#)
- [Derivative Using Forward Difference Formula Pseudocode](#)
- [C Program to Find Derivative Using Forward Difference Formula](#)
- [Derivative Using Backward Difference Formula Algorithm](#)
- [Derivative Using Backward Difference Formula Pseudocode](#)
- [C Program to Find Derivative Using Backward Difference Formula](#)
- [Trapezoidal Method for Numerical Integration Algorithm](#)
- [Trapezoidal Method for Numerical Integration Pseudocode](#)
- [Trapezoidal Method C Program](#)
- [Trapezoidal Method C++ Program](#)
- [Trapezoidal Method Python](#)
- [Simpson's 1/3 Rule Algorithm](#)
- [Simpson's 1/3 Rule Pseudocode](#)
- [Simpson's 1/3 Rule C Program](#)
- [Simpson's 1/3 Rule C++ Program](#)
- [Simpson's 1/3 Rule Python](#)
- [Simpson's 3/8 Rule Algorithm](#)
- [Simpson's 3/8 Rule Pseudocode](#)
- [Simpson's 3/8 Rule C Program](#)
- [Simpson's 3/8 Rule Python](#)
- [Simpson's 3/8 Rule C++ Program](#)
- [Euler's Method Algorithm](#)
- [Euler's Method Pseudocode](#)
- [Euler's Method C Program](#)
- [Euler's Method C++ Program](#)
- [Euler's Method Python](#)
- [RK4 Method Python](#)
- [Runge Kutta \(RK\) Algorithm](#)
- [Runge Kutta \(RK\) Pseudocode](#)
- [Runge Kutta \(RK\) C Program](#)
- [Runge Kutta \(RK\) C++ Program](#)

Gauss Elimination Method Using C



Data Science & AI Program

Master Python, Machine Learning, Statistics, Data Scier
Tableau with No Cost EMI.

Intellipaat

Earlier in [Gauss Elimination Method Algorithm](#) and [Gauss Elimination Method Pseudocode](#) , we discussed about an algorithm and pseudocode for solving systems of linear equation using Gauss Elimination Method. In this tutorial we are going to implement this method using C programming language.

Complete Program for Gauss Elimination method using C Programming Language

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

<

Ads by Google

Stop seeing this ad

Why this ad? ⓘ

```

int main()
{
    float a[SIZE][SIZE], x[SIZE], ratio;
    int i, j, k, n;

    clrscr();

    /* Inputs */
    /* 1. Reading number of unknowns */
    printf("Enter number of unknowns: ");
    scanf("%d", &n);
    /* 2. Reading Augmented Matrix */
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n+1; j++)
        {
            printf("a[%d][%d] = ", i, j);
            scanf("%f", &a[i][j]);
        }
    }
    /* Applying Gauss Elimination */
    for(i=1; i<=n-1; i++)
    {
        if(a[i][i] == 0.0)
        {
            printf("Mathematical Error!");
            exit(0);
        }
        for(j=i+1; j<=n; j++)
        {
            ratio = a[j][i]/a[i][i];
            for(k=1; k<=n+1; k++)
            {
                a[j][k] = a[j][k] - ratio*a[i][k];
            }
        }
    }
    /* Obtaining Solution by Back Substitution */
    x[n] = a[n][n+1]/a[n][n];
    for(i=n-1; i>=1; i--)
    {
        x[i] = a[i][n+1];
        for(j=i+1; j<=n; j++)
        {
            x[i] = x[i] - a[i][j]*x[j];
        }
        x[i] = x[i]/a[i][i];
    }
    /* Displaying Solution */
    printf("\nSolution:\n");
    for(i=1; i<=n; i++)
    {
        printf("x[%d] = %0.3f\n", i, x[i]);
    }
    getch();
}

```



<

Ads by Google

Stop seeing this ad

Why this ad? ⓘ

Output: Gauss Elimination Method for Solving Systems of Linear Equations

Enter number of unknowns: 3

```
a[1][1] = 1
a[1][2] = 1
a[1][3] = 1
a[1][4] = 9
a[2][1] = 2
a[2][2] = -3
a[2][3] = 4
a[2][4] = 13
a[3][1] = 3
a[3][2] = 4
a[3][3] = 5
a[3][4] = 40
```

Solution:

```
x[1] = 1.000
x[2] = 3.000
x[3] = 5.000
```

Recommended Readings

1. [Gauss Elimination Method Algorithm](#)
2. [Gauss Elimination Method Pseudocode](#)
3. [Gauss Elimination Method Using C](#)
4. [Gauss Elimination Method Using C++](#)



Data Science & AI Program

Master Python, Machine Learning, Statistics, Data Science
Tableau with No Cost EMI.

Intellipaat

- Bisection Method
 - [Algorithm for Bisection Method](#)
 - [Pseudocode for Bisection Method](#)
 - [C Program for Bisection Method](#)
 - [C++ Program for Bisection Method](#)
 - [MATLAB Program for Bisection Method](#)
 - [Python Program for Bisection Method](#)
 - [Bisection Method Advantages](#)
 - [Bisection Method Disadvantages](#)
 - [Bisection Method Features](#)
 - [Convergence of Bisection Method](#)
 - [Bisection Method Online Calculator](#)
- Regula Falsi (False Position) Method
 - [Algorithm for Regula Falsi \(False Position\) Method](#)
 - [Pseudocode for Regula Falsi \(False Position\) Method](#)
 - [Features of Regula Falsi](#)
 - [Falsi Position Advantages](#)
 - [False Position Disadvantages](#)
 - [C Program for Regula False \(False Position\) Method](#)
 - [C++ Program for Regula False \(False Position\) Method](#)



Ads by Google

Stop seeing this ad

Why this ad? ⓘ

- [Newton Raphson \(NR\) Method Pseudocode](#)
- [Newton Raphson Method C Program](#)
- [Newton Raphson Method C++ Program](#)
- [Newton Raphson Method Python Program](#)
- [Newton-Raphson MATLAB](#)
- [Features of Newton Raphson Method](#)
- [Newton Raphson Advantages](#)
- [Newton Raphson Disadvantages](#)
- [Newton Raphson Method Online Calculator](#)
- Secant Method
 - [Secant Method Algorithm](#)
 - [Secant Method Pseudocode](#)
 - [Secant Method C Program](#)
 - [Secant Method C++ Program with Output](#)
 - [Secant Method Python Program with Output](#)
 - [Secant Method Online Calculator](#)
- Fixed Point Iteration
 - [Fixed Point Iteration \(Iterative\) Method Algorithm](#)
 - [Fixed Point Iteration \(Iterative\) Method Pseudocode](#)
 - [Fixed Point Iteration \(Iterative\) Method C Program](#)
 - [Fixed Point Iteration \(Iterative\) Python Program](#)
 - [Fixed Point Iteration \(Iterative\) Method C++ Program](#)
 - [Fixed Point Iteration \(Iterative\) Method Online Calculator](#)
- Gauss Elimination
 - [Gauss Elimination Method Algorithm](#)
 - [Gauss Elimination Method Pseudocode](#)
 - [Gauss Elimination C Program](#)
 - [Gauss Elimination C++ Program with Output](#)
 - [Gauss Elimination Method Python Program with Output](#)
 - [Gauss Elimination Method Online Calculator](#)
- Gauss Jordan Method
 - [Gauss Jordan Method Algorithm](#)
 - [Gauss Jordan Method Pseudocode](#)
 - [Gauss Jordan Method C Program](#)
 - [Gauss Jordan Method C++ Program](#)
 - [Gauss Jordan Method Python Program \(With Output\)](#)
 - [Gauss Jordan Method Online Calculator](#)
- Matrix Inverse Using Gauss Jordan
 - [Matrix Inverse Using Gauss Jordan Method Algorithm](#)
 - [Matrix Inverse Using Gauss Jordan Method Pseudocode](#)
 - [Matrix Inverse Using Gauss Jordan C Program](#)
 - [Matrix Inverse Using Gauss Jordan C++ Program](#)
 - [Python Program to Inverse Matrix Using Gauss Jordan](#)
 - [Matrix Inverse Online Calculator](#)
- Power Method
 - [Power Method \(Largest Eigen Value and Vector\) Algorithm](#)
 - [Power Method \(Largest Eigen Value and Vector\) Pseudocode](#)
 - [Power Method \(Largest Eigen Value and Vector\) C Program](#)
 - [Power Method \(Largest Eigen Value and Vector\) C++ Program](#)
 - [Power Method \(Largest Eigen Value & Vector\) Python Program](#)
- Jacobi Iteration Method
 - [Jacobi Iteration Method Algorithm](#)
 - [Jacobi Iteration Method C Program](#)
 - [Jacobi Iteration Method C++ Program with Output](#)
 - [Python Program for Jacobi Iteration](#)
- Gauss Seidel Iteration Method
 - [Gauss Seidel Iteration Method Algorithm](#)
 - [Gauss Seidel Iteration Method C Program](#)
 - [Gauss Seidel Iteration Method C++ Program](#)
 - [Python Program for Gauss Seidel Iteration Method](#)
- Successive Over-Relaxation (SOR)
 - [Python Program for Successive Over Relaxation](#)
- Interpolation
 - [Forward Difference Table Using C](#)
 - [Forward Difference Table Using C++](#)
 - [Python Program to Generate Forward Difference Table](#)
 - [Python Program to Generate Backward Difference Table](#)
 - [Backward Difference Table Using C](#)
 - [Backward Difference Table Using C++](#)
 - [Lagrange Interpolation Method Algorithm](#)
 - [Lagrange Interpolation Method Pseudocode](#)
 - [Lagrange Interpolation Method C Program](#)
 - [Lagrange Interpolation Method C++ Program](#)
 - [Lagrange Interpolation in Python](#)
 - [Linear Interpolation Method Algorithm](#)
 - [Linear Interpolation Method Pseudocode](#)

Ads by Google

Stop seeing this ad

Why this ad? ⓘ

- [Linear Regression Method Pseudocode](#)
- [Linear Regression Method C Program](#)
- [Linear Regression Method C++ Program with Output](#)
- [Linear Regression Python](#)
- [Curve Fitting of Type \$y=ax^b\$ Algorithm](#)
- [Curve Fitting of Type \$y=ax^b\$ Pseudocode](#)
- [Curve Fitting \$y=ax^b\$ C Program](#)
- [Curve Fitting \$y=ax^b\$ Python Program](#)
- [Curve Fitting \$y=ax^b\$ C++ Program](#)
- [Curve Fitting \$y=ab^x\$ Algorithm](#)
- [Curve Fitting \$y=ab^x\$ Pseudocode](#)
- [Curve Fitting \$y=ab^x\$ C Program](#)
- [Curve Fitting \$y=ab^x\$ C++ Program](#)
- [Curve Fitting \$y=ab^x\$ Python Program](#)
- Numerical Differentiation
 - [Derivative Using Forward Difference Formula Algorithm](#)
 - [Derivative Using Forward Difference Formula Pseudocode](#)
 - [C Program to Find Derivative Using Forward Difference Formula](#)
 - [Derivative Using Backward Difference Formula Algorithm](#)
 - [Derivative Using Backward Difference Formula Pseudocode](#)
 - [C Program to Find Derivative Using Backward Difference Formula](#)
- Numerical Integration
 - [Trapezoidal Method for Numerical Integration Algorithm](#)
 - [Trapezoidal Method for Numerical Integration Pseudocode](#)
 - [Trapezoidal Method C Program](#)
 - [Trapezoidal Method C++ Program](#)
 - [Trapezoidal Method Python](#)
 - [Simpson's 1/3 Rule Algorithm](#)
 - [Simpson's 1/3 Rule Pseudocode](#)
 - [Simpson's 1/3 Rule C Program](#)
 - [Simpson's 1/3 Rule C++ Program](#)
 - [Simpson's 1/3 Rule Python](#)
 - [Simpson's 3/8 Rule Algorithm](#)
 - [Simpson's 3/8 Rule Pseudocode](#)
 - [Simpson's 3/8 Rule C Program](#)
 - [Simpson's 3/8 Rule Python](#)
 - [Simpson's 3/8 Rule C++ Program](#)
- Ordinary Differential Equation
 - [Euler's Method Algorithm](#)
 - [Euler's Method Pseudocode](#)
 - [Euler's Method C Program](#)
 - [Euler's Method C++ Program](#)
 - [Euler's Method Python](#)
 - [RK4 Method Python](#)
 - [Runge Kutta \(RK\) Algorithm](#)
 - [Runge Kutta \(RK\) Pseudocode](#)
 - [Runge Kutta \(RK\) C Program](#)
 - [Runge Kutta \(RK\) C++ Program](#)

Data Science & AI Program

Master Python, Machine Learning, Statistics, Data Science, AI, Table
No Cost EMI.

Intellipaat



About Us

Codesansar is online platform that provides tutorials and examples on popular programming languages.

Links

- [About Us](#)
- [Contact Us](#)
- [Privacy Policy](#)



Ads by Google

Stop seeing this ad

Why this ad? ⓘ

- [Twitter](#)
- [Youtube](#)
- [Github](#)
- [Instagram](#)

© 2022 Codesansar. All Rights Reserved.



Ads by Google

Stop seeing this ad

Why this ad? ⓘ