Numerical Analysis

Course Contents

Solution of Non Linear Equations
Solution of Linear System of Equations
Approximation of Eigen Values
Interpolation and Polynomial Approximation
Numerical Differentiation
Numerical Integration
Numerical Solution of Ordinary Differential Equations

Introduction

We begin this chapter with some of the basic concept of representation of numbers on computers and errors introduced during computation. Problem solving using computers and the steps involved are also discussed in brief.

Number (s) System (s)

In our daily life, we use numbers based on the decimal system. In this system, we use ten symbols 0, 1,…,9 and the number 10 is called the base of the system.

Thus, when a base N is given, we need N different symbols 0, 1, 2, …,(N – 1) to represent an arbitrary number.

The number systems commonly used in computers are

Base, N	Number
2	Binary
8	Octal
10	Decimal
16	Hexadecimal

An arbitrary real number a, can be written as

$$a = a_m N^m + a_{m-1} N^{m-1} + \cdots + a_1 N^1 + a_0 + a_{-1} N^{-1} + \cdots + a_{-m} N^{-m}$$

In binary system, it has the form,

$$a = a_m 2^m + a_{m-1} 2^{m-1} + \cdots + a_1 2^1 + a_0 + a_{-1} 2^{-1} + \cdots + a_{-m} 2^{-m}$$

The decimal number 1729 is represented and calculated

$$(1729)_{10} = 1 \times 10^3 + 7 \times 10^2 + 2 \times 10^1 + 9 \times 10^0$$

While the decimal equivalent of binary number 10011001 is

$$1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 0 \times 2^{-6} + 1 \times 2^{-7}$$

$$= 1 + \frac{1}{8} + \frac{1}{16} + \frac{1}{128}$$

$$= (1.1953125)_{10}$$

Electronic computers use binary system whose base is 2. The two symbols used in this system are 0 and 1, which are called binary digits or simply bits.

The internal representation of any data within a computer is in binary from. However, we prefer data input and output of numerical results in decimal system. Within the computer, the arithmetic is carried out in binary form.

Conversion of decimal number 47 into its binary equivalent

Sol.

Errors in Computations

Numerically, computed solutions are subject to certain errors. It may be fruitful to identify the error sources and their growth while classifying the errors in numerical computation. These are

Inherent errors,

Local round-off errors

Local truncation errors

Inherent errors

It is that quantity of error which is present in the statement of the problem itself, before finding its solution. It arises due to the simplified assumptions made in the mathematical modeling of a problem. It can also arise when the data is obtained from certain physical measurements of the parameters of the problem.

Local round-off errors

Every computer has a finite word length and therefore it is possible to store only a fixed number of digits of a given input number. Since computers store information in binary form, storing an exact decimal number in its binary form into the computer memory gives an error. This error is computer dependent.

At the end of computation of a particular problem, the final results in the computer, which is obviously in binary form, should be converted into decimal form-a form understandable to the user-before their print out. Therefore, an additional error is committed at this stage too.

This error is called local round-off error.

$$(0.7625)_{10} = (0.110000110011)_2$$

If a particular computer system has a word length of 12 bits only, then the decimal number 0.7625 is stored in the computer memory in binary form as 0.110000110011. However, it is equivalent to 0.76245.

Thus, in storing the number 0.7625, we have committed an error equal to 0.00005, which is the round-off error; inherent with the computer system considered.

Thus, we define the error as

$$\text{Error} = \text{True value} - \text{Computed value}$$

Absolute error, denoted by |Error|,

While, the relative error is defined as

$$\text{Relative error} = \frac{|\text{Error}|}{|\text{True value}|}$$

Local truncation error

It is generally easier to expand a function into a power series using Taylor series expansion and evaluate it by retaining the first few terms. For example, we may approximate the function f (x) = cos x by the series

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + \cdots$$

If we use only the first three terms to compute cos x for a given x, we get an approximate answer. Here, the error is due to truncating the series. Suppose, we retain the first n terms, the truncation error (TE) is given by

Polynomial

An expression of the form $f(x) \quad a_0 x^n \quad a_1 x^{n\,1} \quad a_2 x^{n\,2} \quad ... \quad a_{n\,1} x \quad a_n$ where n is a positive integer and $a_0, a_1, a_2 \quad ....a_n$ are real constants, such type of expression is called an nth degree polynomial in x if $a_0 \quad 0z$

Algebraic equation:

An equation f(x)=0 is said to be the algebraic equation in x if it is purely a polynomial in x.

For example

$x^5 \quad x^4 \quad 3x^2 \quad x \quad 6 \quad 0$ It is a fifth order polynomial and so this equation is an algebraic equation.

$x^3 \quad 6 \quad 0$

$x^6 \quad x \quad 0$

$y^4 \quad 4y^3 \quad 3y^2 \quad y \quad 2 \quad 0$ polynomial in y

$t^4 \quad 6t^2 \quad 21 \quad 0$ polynomail in t

These all are the examples of the polynomial or algebraic equations.

Some facts

1. Every equation of the form f(x)=0 has at least one root ,it may be real or complex.
2. Every polynomial of nth degree has n and only n roots.
3. If f(x) =0 is an equation of odd degree, then it has at least one real root whose sign is opposite to that of last term.
4. If f(x)=0 is an equation of even degree whose last term is negative then it has at least one positive and at least one negative root .

Transcendental equation

An equation is said to be transcendental equation if it has logarithmic, trigonometric and exponential function or combination of all these three.

For example

$e^x \quad 5x \quad 3 \quad 0$ it is a transcendental equation as it has an exponential function

$e^x \quad \sin x \quad 0$

$\ln x \quad \sin x \quad 0$

$2\sec^2 x \quad \tan x \quad e^x \quad 0$

These all are the examples of transcendental equation.

Root of an equation

For an equation f(x) =0 to find the solution we find such value which satisfy the equation f(x)=0,these values are known as the roots of the equation .

A value a is known as the root of an equation f(x) =0 if and only if f (a) =0.

Regula-Falsi method (Method of false position)

Here we choose two points x_n and x_{n-1} such that $f(x_n)$ and $f(x_{n-1})$ have opposite signs. Intermediate value property suggests that the graph of the y=f(x) crosses the x-axis between these two points and therefore a root lies between these two points.
Thus to find the real root of f(x)=0 using Regula-Falsi method ,we replace the part of the curve between the points $A(x_n, f(x_n))$ and $B(x_{n-1}, f(x_{n-1}))$ by a chord in the interval and we take the point of intersection of this chord with x-axis as initial approximation.
Now, the equation of the chord joining the points A and B is,

$$\frac{y - f(x_n)}{f(x_{n-1}) - f(x_n)} = \frac{x - x_n}{x_{n-1} - x_n}$$

Setting y=0 in the above equation we get

$$x = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$$

Hence the first approximation to the root is given by

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$$

We observe that $f(x_{n-1})$ and $f(x_{n+1})$ are of opposite signs thus it is possible to apply the above procedure, to determine the line through B and A_1 and so on.
Hence for successive approximation to the root above formula is used.

Example

Use the Regula-Falsi method to compute a real root of the equation $x^3 - 9x + 1 = 0$,
(i) if the root lies between 2 and 4
(ii) if the root lies between 2 and 3.
Comment on the results.

Solution

Let
$f(x) = x^3 - 9x + 1$

$f(2) = 2^3 - 9(2) + 1 = 8 - 18 + 1 = -9$ and $f(4) = 4^3 - 9(4) + 1 = 64 - 36 + 1 = 29$.

Since $f(2)$ and $f(4)$ are of opposite signs, the root of $f(x) = 0$ lies between 2 and 4.
Taking $x_1 = 2, x_2 = 4$ and using Regula-Falsi method, the first approximation is given by

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2) = 4 - \frac{4 - 2}{29 - (-9)}(29) = 4 - \frac{2(29)}{38}$$

$$= 4 - \frac{58}{38} = 4 - 1.5263 = 2.4736$$

Example

Using Regula-Falsi method, find the real root of the following equation correct, to three decimal places: $x \log_{10} x = 1.2$

Solution:

Let $f(x) = x \log_{10} x - 1.2$

$f(2) = 2\log_{10} 2 - 1.2 = -0.5979,$

$f(3) = 3\log_{10} 3 - 1.2 = 0.2314.$

Since f (2) and f (3) are of opposite signs, the real root lies between x1 = 2, x2 = 3.

The first approximation is obtained from

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2) = 3 - \frac{3-2}{0.2314 + 0.5979}(0.2314)$$

$$= 3 - \frac{0.2314}{0.8293} = 2.72097$$

$f(x_3) = $ Let $f(x) = 2.72097 \log_{10} 2.72097 - 1.2 = -0.01713.$

Since f (x2) and f (x3) are of opposite signs, the root of f (x) = 0 lies between x2 and x3. Now, the second approximation is given by

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3) = 2.72097 - \frac{2.72097 - 3}{-0.1713 - 0.2314}(-0.1713) = 2.7402$$

$f(x_4) = 2.7402 \log_{10} 2.7402 - 1.2 = -3.8905 \times 10^{-5}$

Thus, the root of the given equation correct to three decimal places is 2.740

NOTE: Here if TOL is given then we can simply find the value of TOL by subtracting both the consecutive roots and write it in the exponential notation if the required TOL is obtained then we stop.

Method of Iteration

Method of iterations can be applied to find a real root of the equation $f(x) = 0$ by rewriting the same in the form.

$x = \phi(x)$

Let $x = x_0$ be the initial approximation to the actual root, say, of the equation .then the first approximation is $x_1 = \phi(x_0)$ and the successive approximation are $x_2 = \phi(x_1)$

$x_3 = \phi(x_2), x_4 = \phi(x_3), \ldots, x_n = \phi(x_{n-1})$ if sequence of the approximate roots, $x_1, x_2, x_3, \ldots x_n$ converges to α it is taken as the root of the equation f(x)=0.

For convergence purpose the initial approximation is to be done carefully.the choice of the x_0 is done according to the theorem.

Newton -Raphson Method

This method is one of the most powerful method and well known methods, used for finding a root of f(x)=0 the formula many be derived in many ways the simplest way to derive this formula is by using the first two terms in Taylor's series expansion of the form,

$$f(x_{n\ 1}) \quad f(x_n) \quad (x_{n\ 1} \quad x_n)\, f'(x_n)$$

setting $f(x_{n\ 1}) \quad 0$ give,s

$$f(x_n) \quad (x_{n\ 1} \quad x_n)\, f'(x_n) \quad 0$$

thus on simplificatio,n we ge,t

$$x_{n\ 1} \quad x_n \quad \frac{f(x_n)}{f'(x_n)} \text{ for } n \quad 0,1,2...$$

Geometrical interpretation

Let the curve f(x)=0 meet the x-axis at x=meet the x axis at x=.it means that is the original root of the f(x)=0.Let x_0 be the point near the root of the equation f(x)=0 then the equation of the tangent at $P_0[x_0, f(x_0)]$ is

$$y \quad f(x_0) \quad f'(x_0)(x \quad x_0)$$

This cuts the x-axis at $x_1 \quad x_0 \quad \dfrac{f(x_0)}{f'(x_0)}$

This is the first approximation to the root .if $P_1[x_1, f(x_1)]$ is the point corresponding to x_1 on the curve then the tangent at P_1 is

$$y \quad f(x_1) \quad f'(x_1)(x \quad x_1)$$

This cuts the x-axis at $x_2 \quad x_1 \quad \dfrac{f(x_1)}{f'(x_1)}$

This is the second approximation to the roo.tRepeating this process we will get the root with better approximations quite rapidly.

Note:

1. When $f'(x)$ very large .i.e. is when the sloipselarge, then h will be small (as assumed) and hence, the root can be calculated in even less time.
2. If we choose the initial approximation x_0 close to the root then we get the root of the equation very quickly.
3. The process will evidently fail if $f'(x) \quad 0$ is in the neighborhood of the root. In such cases the regula falsi method should be used.

Secant Method

The secant method is modified form of Newton-Raphson method .If in Newton-Raphson method; we replace the derivative $f'(x_n)$ by the difference ratio, i.e,

$$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

Where x_n and x_{n-1} are two approximations of the root we get

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

$$= \frac{x_n f(x_n) - x_n f(x_{n-1}) - f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

$$= \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Provided $f(x_n) \neq f(x_{n-1})$

This method requires two starting values x_0 and x_1; values at both these points are calculated which gives two points on the curve the next point on the curve are obtained by using the derived formula. We continue this procedure until we get the root of required accuracy.

Geometrical Interpretation

Geometrically, the secant method corresponds to drawing secants rather than tangents to obtain various approximations to root;

To obtain x_2 we find the intersection between the secant through the points $(x_0, f(x_0))$ And $(x_1, f(x_1))$ and the x-axis.

It is experienced that the secant method converges rather quickly .there is a possibility of the divergence if the two roots lie on the same side of the curve .the order of the convergence of the decant method equal to $\frac{(1+\sqrt{5})}{2} = 1.618$, which shows that this method has the order of convergence slightly inferior that of Newton-Raphson method, In this method f(x) is not required to change signs between the estimate.

Muller’s Method

Muller’s method is a generalization of the secant method, in the sense that it does not require the derivative of the function. It is an iterative method that requires three starting [redacted], [redacted], and [redacted]. A parabola is constructed that passes through the three points; then the quadratic formula is used to find a root of the quadratic for the next approximation. It has been proved that near a simple root Muller's method converges faster than the secant method and almost as fast as Newton's method. The method can be used to find real or complex zeros of a function and can be programmed to use complex arithmetic.

Example

Do two iterations of Muller’s method to solve $x^3 \; 3x \; 1 \; 0$ starting with $x_0 \; 0.5, x_2 \; 0, x_1 \; 1$

Solution

$$f(x_0) \quad f_0 \quad (0.5)^3 \quad 3(0.5) \quad 1 \quad 0.375$$

$$f(x_1) \quad f_1 \quad (1)^3 \quad 3(1) \quad 1 \quad 1$$

$$f(x_2) \quad f_2 \quad 0 \quad 3 \quad 0 \quad 1 \quad 1 \qquad u$$

$$c \quad f_0 \quad 0.375$$

$$h_1 \quad x_1 \quad x_0 \quad 0.5$$

$$h_2 \quad x_0 \quad x_2 \quad 0.5$$

$$a \quad \frac{h_2 f_1 \quad (h_1 \quad h_2) f_0 \quad h_1 f_2}{h_1 h_2 (h_1 \quad h_2)}$$

$$\frac{(0.5)(\;1) \quad (\;0.375) \quad (0.5)}{0.25} \quad 1.5$$

$$b \quad \frac{f_1 \quad f_0 \quad a h_1^2}{h_1} \quad 2$$

$$x \; ? \; x_0 \quad \frac{2c}{b \quad \sqrt{b^2 \quad 4ac}}$$

$$0.5 \quad \frac{2(\;0.375)}{2 \quad \sqrt{4 \quad 4(1.5)(0.375)}}$$

$$0.5 \quad \frac{0.75}{2 \quad \sqrt{4 \quad 2.25}} \quad 0.33333 \quad E0.5$$

Solution of Linear System of Equations and Matrix Inversion

The solution of the system of equations gives n unknown values $x_1, x_2, \ldots, x_n$, which satisfy the system simultaneously. If $m > n$, the system usually will have an infinite number of solutions.

If $|A| \neq 0$ then the system will have a unique solution.

If $|A| = 0$,

Then there exists no solution.

Numerical methods for finding the solution of the system of equations are classified as direct and iterative methods. In direct methods, we get the solution of the system after performing all the steps involved in the procedure. The direct methods consist of elimination methods and decomposition methods.

Under elimination methods, we consider, Gaussian elimination and Gauss-Jordan elimination methods

Crout's reduction also known as Cholesky's reduction is considered under decomposition methods.

Under iterative methods, the initial approximate solution is assumed to be known and is improved towards the exact solution in an iterative way. We consider Jacobi, Gauss-Seidel and relaxation methods under iterative methods.

Gaussian Elimination Method

In this method, the solution to the system of equations is obtained in two stages.

i) the given system of equations is reduced to an equivalent upper triangular form using elementary transformations

ii) the upper triangular system is solved using back substitution procedure

This method is explained by considering a system of n equations in n unknowns in the form as follows

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\;\;. \\ &\;\;. \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}\right\}$$

Stage I: Substituting the value of x_1 from first equation into the rest

$$\left.\begin{aligned} x_1 + a'_{12}x_2 + \cdots + a'_{1n}x_n &= b'_1 \\ a'_{22}x_2 + \cdots + a'_{2n}x_n &= b'_2 \\ &\;\;\vdots \\ a'_{n2}x_2 + \cdots + a'_{nn}x_n &= b'_n \end{aligned}\right\}$$

Solution of Linear System of Equations and Matrix Inversion

Gauss–Jordon Elimination Method

This method is a variation of Gaussian elimination method. In this method, the elements above and below the diagonal are simultaneously made zero. That is a given system is reduced to an equivalent diagonal form using elementary transformations. Then the solution of the resulting diagonal system is obtained. Sometimes, we normalize the pivot row with respect to the pivot element, before elimination. Partial pivoting is also used whenever the pivot element becomes zero.

Example

Solve the system of equations using Gauss-Jordan elimination method:

$$\left.\begin{array}{l} x \quad 2y \quad z \quad 8 \\ 2x \quad 3y \quad 4z \quad 20 \\ 4x \quad 3y \quad 2z \quad 16 \end{array}\right\}$$

Solution

In matrix notation, the given system can be written as

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 4 \\ 4 & 3 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 20 \\ 16 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 5 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 16 \end{bmatrix} \quad (-2)\,R1+R2 \text{ and } (-4)\,R1+R3$$

Now, we eliminate y from the first and third rows using the second row. Thus, we get

$$\begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 12 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ 4 \\ 36 \end{bmatrix}$$

Before, eliminating z from the first and second row, normalize the third row with respect to the pivot element, we get

$$\begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ 4 \\ 3 \end{bmatrix}$$

Using the third row of above Equation, and eliminating z from the first and second rows, we obtain

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

The solution is

$x = 1, y = 2, z = 3.$

Example

Solve the system of equation by using the Gauss Jordan elimination method

$10x \quad y \quad z \quad 12$

$2x \quad 10y \quad z \quad 13$

$x \quad y \quad 5z \quad 7$

Solution

Solution of Linear System of Equations and Matrix Inversion

Jacobi's Method

This is an iterative method, where initial approximate solution to a given system of equations is assumed and is improved towards the exact solution in an iterative way.

In general, when the coefficient matrix of the system of equations is a sparse matrix (many elements are zero), iterative methods have definite advantage over direct methods in respect of economy of computer memory

Such sparse matrices arise in computing the numerical solution of partial differential equations

Let us consider

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
\vdots \qquad \vdots \qquad &\quad \vdots \qquad \vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_1
\end{aligned}$$

In this method, we assume that the coefficient matrix [A] is strictly diagonally dominant, that is, in each row of [A] the modulus of the diagonal element exceeds the sum of the off-diagonal elements.

We also assume that the diagonal element do not vanish. If any diagonal element vanishes, the equations can always be rearranged to satisfy this condition.

Now the above system of equations can be written as

$$\begin{aligned}
x_1 &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2 - \cdots - \frac{a_{1n}}{a_{11}}x_n \\
x_2 &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1 - \cdots - \frac{a_{2n}}{a_{22}}x_n \\
\vdots \quad & \quad \vdots \qquad \vdots \qquad \qquad \vdots \\
x_n &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}}x_1 - \cdots - \frac{a_{n(n-1)}}{a_{nn}}x_{n-1}
\end{aligned}$$

We shall take this solution vector $(x_1, x_2, \ldots, x_n)^T$ as a first approximation to the exact solution of system. For convenience, let us denote the first approximation vector by $(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})$ got after taking as an initial starting vector.

Substituting this first approximation in the right-hand side of system, we obtain the second approximation to the given system in the form

Solution of Linear System of Equations and Matrix Inversion Gauss–Seidel Iteration Method

It is another well-known iterative method for solving a system of linear equations of the form

$$\left.\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2\\
\vdots \qquad\quad \vdots \qquad\qquad & \quad\ \vdots\\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}\right\}$$

In Jacobi's method, the $(r + 1)$th approximation to the above system is given by Equations

$$\left.\begin{aligned}
x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2^{(r)} - \cdots - \frac{a_{1n}}{a_{11}}x_n^{(r)}\\
x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1^{(r)} - \cdots - \frac{a_{2n}}{a_{22}}x_n^{(r)}\\
\vdots \qquad & \qquad \vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots\\
x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}}x_1^{(r)} - \cdots - \frac{a_{n(n-1)}}{a_{nn}}x_{n-1}^{(r)}
\end{aligned}\right\}$$

Here we can observe that no element of $x_i^{(r+1)}$ replaces $x_i^{(r)}$ entirely for the next cycle of computation.

In Gauss-Seidel method, the corresponding elements of $x_i^{(r+1)}$ replaces those of $x_i^{(r)}$ as soon as they become available.

Hence, it is called the method of successive displacements. For illustration consider

$$\left.\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2\\
\vdots \qquad\quad \vdots \qquad\qquad & \quad\ \vdots\\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}\right\}$$

In Gauss-Seidel iteration, the $(r + 1)$th approximation or iteration is computed from:

$$\left.\begin{aligned}
x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2^{(r)} - \cdots - \frac{a_{1n}}{a_{11}}x_n^{(r)}\\
x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1^{(r+1)} - \cdots - \frac{a_{2n}}{a_{22}}x_n^{(r)}\\
\vdots \qquad & \qquad \vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots\\
x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}}x_1^{(r+1)} - \cdots - \frac{a_{n(n-1)}}{a_{nn}}x_{n-1}^{(r+1)}
\end{aligned}\right\}$$

Thus, the general procedure can be written in the following compact form