UNIT 3:

Numerical Methods

Numerical Method

Numerical methods are techniques that are used to approximate Mathematical procedures. We need approximations because we either cannot solve the procedure analytically or because the analytical method is intractable (an example is solving a set of a thousand simultaneous linear equations for a thousand unknowns).

Different Types of Numerical Methods

The numerical analysts and Mathematicians used have a variety of tools that they use to develop numerical methods for solving Mathematical problems. The most important idea, mentioned earlier, that cuts across all sorts of Mathematical problems is that of changing a given problem with a 'near problem' that can be easily solved. There are other ideas that differ on the type of Mathematical problem solved.

An Introduction to Numerical Methods for Solving Common Division Problems Given Below:

- Euler method - the most basic way to solve ODE
- Clear and vague methods - vague methods need to solve the problem in every step
- The Euler Back Road - the obvious variation of the Euler method
- Trapezoidal law - the direct method of the second system
- Runge-Kutta Methods - one of the two main categories of problems of the first value.

Numerical Methods

- **Newton method**
  Some calculations cannot be solved using algebra or other Mathematical methods. For this we need to use numerical methods. Newton's method is one such method and allows us to calculate the solution of f (x) = 0.

- **Simpson Law**
  The other important ones cannot be assessed in terms of integration rules or basic functions. Simpson's law is a numerical method that calculates the numerical value of a direct combination.

- **Trapezoidal law**
  A trapezoidal rule is a numerical method that calculates the numerical value of a direct combination. The other important ones cannot be assessed in terms of integration rules or basic functions.

Numerical Computation

The term “numerical computations” means to use computers for solving problems involving real numbers. In this process of problem-solving, we can distinguish several more or less distinct phases. The first phase is formulation. While formulating a Mathematical model of a physical situation, scientists should take into account the fact that they expect to solve a problem on a computer. Therefore they will provide for specific objectives, proper input data, adequate checks, and for the type and amount of output.

Once a problem has been formulated, then the numerical methods, together with preliminary error analysis, must be devised for solving the problem. A numerical method that can be used to solve a problem is called an algorithm. An algorithm is a complete and unambiguous set of procedures that are used to find the solution to a Mathematical problem. The selection or construction of appropriate algorithms is done with the help of Numerical Analysis. We have to decide on a specific algorithm or set of algorithms for solving the problem, numerical analysts should also consider all the sources of error that may affect the results. They should consider how much accuracy is required. To estimate the magnitude of the round-off and discretization errors, and determine an appropriate step size or the number of iterations required.

The programmer should transform the suggested algorithm into a set of unambiguous that is followed by step-by-step instructions to the computer. The flow chart is the first step in this procedure. A flow chart is simply a set of procedures, that are usually written in logical block form, which the computer will follow. The complexity of the flow will depend upon the complexity of the problem and the amount of detail included. However, it should be possible for someone else other than the programmer to follow the flow of information from the chart. The flow chart is an effective aid to the programmer, they must translate its major functions into a program. And, at the same time, it is an effective means of communication to others who wish to understand what the program does.

Numerical Computing Characteristics

- **Accuracy:** Every numerical method introduces errors. It may be due to the use of the proper Mathematical process or due to accurate representation and change of numbers on the computer.
- **Efficiency:** Another consideration in choosing a numerical method for a Mathematical model solution efficiency Means the amount of effort required by both people and computers to use the method.
- **Numerical instability:** Another problem presented by a numerical method is numerical instability. Errors included in the calculation, from any source, increase in different ways. In some cases, these errors are usually rapid, resulting in catastrophic results.

Numerical Computing Process

- Construction of a Mathematical model.
- Construction of an appropriate numerical system.
- Implementation of a solution.
- Verification of the solution.

Significant Figures

Significant figures are used to establish the number which is presented in the form of digits. These digits carry a meaningful representation of numbers. The term significant digits are also used often instead of figures. We can identify the number of significant digits by counting all the values starting from the 1st non-zero digit located on the left. **For example,** 12.45 has four significant digits.

The significant figures of a given number are those significant or important digits, which convey the meaning according to its accuracy. For example, 6.658 has four significant digits. These substantial figures provide precision to the numbers. They are also termed as significant digits.

Rules for Significant Figures

- All non-zero digits are significant. 198745 contains six significant digits.
- All zeros that occur between any two non zero digits are significant. For example, 108.0097 contains seven significant digits.
- All zeros that are on the right of a decimal point and also to the left of a non-zero digit is never significant. For example, 0.00798 contained three significant digits.
- All zeros that are on the right of a decimal point are significant, only if, a non-zero digit does not follow them. For example, 20.00 contains four significant digits.
- All the zeros that are on the right of the last non-zero digit, after the decimal point, are significant. For example, 0.0079800 contains five significant digits.
- All the zeros that are on the right of the last non-zero digit are significant if they come from a measurement. For example, 1090 m contains four significant digits.

Rounding Significant Figures

A number is rounded off to the required number of significant digits by leaving one or more digits from the right. When the first digit in left is less than 5, the last digit held should remain constant. When the first digit is greater than 5, the last digit is rounded up. When the digit left is exactly 5, the number held is rounded up or down to receive an even number. When more than one digit is left, rounding off should be done as a whole instead of one digit at a time.

There are two rules to round off the significant numbers:

1. First, we have to check, up to which digit the rounding off should be performed. If the number after the rounding off digit is less than 5, then we have to exclude all the numbers present on the right side.

2. But if the digit next to the rounding off digit is greater than 5, then we have to add 1 to the rounding off digit and exclude the other numbers on the right side.

Errors in numerical methods

Numerical methods are mathematical techniques used to approximate solutions to problems that cannot be solved exactly. While these methods are valuable tools, they are not without limitations and can introduce errors in various ways. Here are some common errors that can arise in numerical methods:

1. Round-off error: Round-off error occurs due to the limited precision of computers. Numbers that cannot be represented exactly in binary format are rounded to the nearest representable value. Cumulative round-off errors can propagate and affect the accuracy of calculations.
2. Truncation error: Truncation error arises from approximating an infinite process or function by a finite one. It occurs when a mathematical expression or algorithm is simplified or truncated, leading to deviations from the exact solution. Truncation errors can be minimized by using more accurate approximation methods or by increasing the number of steps or iterations.
3. Discretization error: Discretization error occurs when continuous problems or functions are approximated by discrete methods. For example, when solving differential equations, the continuous domain is divided into discrete points, and the equations are approximated at these points. The error arises due to the mismatch between the continuous and discrete representations.
4. Interpolation error: Interpolation is a common technique used to estimate values between known data points. Interpolation errors occur when the chosen interpolation method fails to capture the true behavior of the underlying data accurately. Higher-order interpolation methods can reduce this error.
5. Convergence error: Numerical methods often rely on iterative processes that converge towards the desired solution. Convergence error refers to the difference between the obtained approximation and the true solution. If the convergence is slow or the iteration is terminated prematurely, the error can be significant.
6. Stability error: Some numerical methods may become unstable for certain input conditions. Stability errors lead to unbounded growth of errors over time or iterations, rendering the solution useless. Analyzing the stability of a method is crucial to ensure accurate and reliable results.
7. Discretization step error: In numerical methods that involve dividing the domain into smaller steps, such as in numerical integration or differentiation, the size of the step can

affect the accuracy of the approximation. Smaller steps generally yield more accurate results, but there is a trade-off with computational efficiency.

Error propagation

Error propagation refers to the phenomenon where errors introduced at one stage of a numerical method can propagate and affect the accuracy of subsequent calculations. It is essential to understand how errors propagate to assess the overall reliability of a numerical solution. Here are some common ways in which error propagation can occur in numerical methods:

1. Round-off error propagation: Round-off errors arise due to the limited precision of computers. When performing arithmetic operations, round-off errors can accumulate and propagate throughout the computations, leading to a loss of accuracy. The magnitude of round-off error propagation depends on the number of operations performed and the precision of the arithmetic used.
2. Truncation error propagation: Truncation errors occur when an infinite process or function is approximated by a finite one. If the truncation error is not negligible compared to the desired accuracy, it can propagate and affect subsequent calculations. Truncation errors typically increase as the number of steps or iterations increases, leading to larger errors in the final result.
3. Sensitivity to initial conditions: Some numerical methods, such as iterative methods or chaotic systems, are sensitive to the initial conditions. Small errors in the initial values or starting points can lead to significant deviations in the final solution. The error propagation can be particularly pronounced in systems with unstable or chaotic behavior.
4. Discretization error propagation: Discretization errors occur when continuous problems or functions are approximated by discrete methods. These errors can propagate if subsequent calculations rely on the inaccurate discrete representation. As the discretization becomes finer or coarser, the propagation of errors can vary, and it is crucial to strike a balance between accuracy and computational efficiency.
5. Interpolation error propagation: Interpolation is often used to estimate values between known data points. If the interpolation method used is not accurate, errors in the estimated values can propagate throughout the calculations that depend on those estimates. Higher-order interpolation methods generally lead to smaller error propagation.
6. Stability error propagation: Numerical methods that are unstable can exhibit rapid error growth. If the error at each iteration or time step increases without bound, it will propagate and render the solution useless. Stable methods that control error growth are preferred to ensure reliable and accurate results.

Roots of non linear equations

In numerical methods, finding the roots of nonlinear equations is often accomplished using iterative methods. These methods involve approximating the root of an equation through a sequence of iterations until a desired level of accuracy is achieved. Here are two commonly used iterative methods for finding roots of nonlinear equations:

1. Newton's Method (or Newton-Raphson Method): Newton's method is an iterative root-finding algorithm that utilizes the concept of tangent lines to approximate the root of a function. Given an initial guess x_0, the method iteratively refines the approximation using the formula:

   $x_{i+1} = x_i - f(x_i)/f'(x_i)$

   Here, f(x) is the nonlinear function, and f'(x) is its derivative. The process continues until the desired level of accuracy is achieved or convergence is obtained.

2. Secant Method: The secant method is a numerical technique for finding roots of a function. It is similar to Newton's method but does not require the derivative of the function. Instead, it estimates the derivative using the slope of a secant line through two points on the function. The iteration formula is as follows:

   $x_{i+1} = x_i - f(x_i) * (x_i - x_{i-1}) / (f(x_i) - f(x_{i-1}))$

   In this case, x_{i-1} and x_i are two successive approximations. The process continues until convergence is achieved.

These methods are just a few examples of the numerous iterative techniques available for finding roots of nonlinear equations. Each method has its own advantages and limitations, and the choice of method depends on factors such as the properties of the function, initial guess, and desired level of accuracy.

Bisection Method

The bisection method is used to find the roots of a polynomial equation. It separates the interval and subdivides the interval in which the root of the equation lies. The principle behind this method is the intermediate theorem for continuous functions. It works by narrowing the gap between the positive and negative intervals until it closes in on the correct answer. This method narrows the gap by taking the average of the positive and negative intervals. It is a simple method and it is relatively slow. The bisection method is also known as interval halving method, root-finding method, binary search method or dichotomy method.

Let us consider a continuous function “f” which is defined on the closed interval [a, b], is given with f(a) and f(b) of different signs. Then by intermediate theorem, there exists a point x belong to (a, b) for which f(x) = 0.

Bisection Method Algorithm

Follow the below procedure to get the solution for the continuous function:

For any continuous function f(x),

- Find two points, say a and b such that a < b and f(a)* f(b) < 0
- Find the midpoint of a and b, say “t”
- t is the root of the given function if f(t) = 0; else follow the next step
- Divide the interval [a, b] – If f(t)*f(a) <0, there exist a root between t and a
  – else if f(t) *f (b) < 0, there exist a root between t and b
- Repeat above three steps until f(t) = 0.

The bisection method is an approximation method to find the roots of the given equation by repeatedly dividing the interval. This method will divide the interval until the resulting interval is found, which is extremely small.

Bisection Method Example

Question: Determine the root of the given equation $x^2-3 = 0$ for $x \in [1, 2]$

Solution:

Given: $x^2-3 = 0$

Let $f(x) = x^2-3$

Now, find the value of f(x) at a= 1 and b=2.

$f(x=1) = 1^2-3 = 1 – 3 = -2 < 0$

$f(x=2) = 2^2-3 = 4 – 3 = 1 > 0$

The given function is continuous, and the root lies in the interval [1, 2].

Let “t” be the midpoint of the interval.

I.e., t = (1+2)/2

$t = 3 / 2$

$t = 1.5$

Therefore, the value of the function at “t” is

$f(t) = f(1.5) = (1.5)^2 - 3 = 2.25 – 3 = -0.75 < 0$

If $f(t)<0$, assume $a = t$.

and

If $f(t)>0$, assume $b = t$.

$f(t)$ is negative, so a is replaced with $t = 1.5$ for the next iterations.

The iterations for the given functions are:

Iterations	**a**	**b**	**t**	**f(a)**	**f(b)**	**f(t)**
1	1	2	1.5	-2	1	-0.75
2	1.5	2	1.75	-0.75	1	0.062
3	1.5	1.75	1.625	-0.75	0.0625	-0.359
4	1.625	1.75	1.6875	-0.3594	0.0625	-0.1523
5	1.6875	1.75	1.7188	-01523	0.0625	-0.0457
6	1.7188	1.75	1.7344	-0.0457	0.0625	0.0081
7	1.7188	1.7344	1.7266	-0.0457	0.0081	-0.0189

So, at the seventh iteration, we get the final interval [1.7266, 1.7344]

Hence, 1.7344 is the approximated solution.

Regula Falsi Method:

Regula Falsi is one of the oldest methods to find the real root of an equation f(x) = 0 and closely resembles with Bisection method. It requires less computational effort as we need to evaluate only one function per iteration.

Formula

`X3 = X1(fX2) - X2(fX1)/ f(X2) -f(X1)`

Example

Problem: Find a root of an equation f(x)=x3-x-1

Solution:

Given equation, x3-x-1=0

let x = 0, 1, 2

In 1st iteration :

f(1)=-1<0 and f(2)=5>0

Root lies between these two points x0=1 and x1=2

x2=x0-f(x0)

= x1-x0

f(x1)-f(x0)

x2=1-(-1)·

= 2-1

= 5-(-1)

x2=1.16667

f(x2)=f(1.16667)=-0.5787<0

In 2nd iteration :

f(1.16667)=-0.5787<0 and f(2)=5>0

Root lies between these two points x0=1.16667 and x1=2

x3=x0-f(x0)

x1-x0

f(x1)-f(x0)

x3=1.16667-(-0.5787)

2-1.16667

5-(-0.5787)

x3=1.25311

f(x3)=f(1.25311)=-0.28536<0

In 3rd iteration :
f(1.25311)=-0.28536<0 and f(2)=5>0

Root lies between these two points x0=1.25311 and x1=2

x4=x0-f(x0)·

x1-x0

f(x1)-f(x0)

x4=1.25311-(-0.28536)·

2-1.25311

5-(-0.28536)

x4=1.29344

f(x4)=f(1.29344)=-0.12954<0

In 4th iteration :
f(1.29344)=-0.12954<0 and f(2)=5>0

Root lies between these two points x0=1.29344 and x1=2

x5=x0-f(x0)·

x1-x0

f(x1)-f(x0)

x5=1.29344-(-0.12954)·

2-1.29344

5-(-0.12954)

x5=1.31128

f(x5)=f(1.31128)=-0.05659<0

In 5th iteration :
f(1.31128)=-0.05659<0 and f(2)=5>0

Root lies between these two points x0=1.31128 and x1=2 *x6=x0-f(x0)·*

x1-x0

f(x1)-f(x0)

x6=1.31128-(-0.05659)·

2-1.31128

5-(-0.05659)

x6=1.31899

f(x6)=f(1.31899)=-0.0243<0

In 6th iteration :
f(1.31899)=-0.0243<0 and f(2)=5>0

Root lies between these two points x0=1.31899 and x1=2

x7=x0-f(x0)·

x1-x0

f(x1)-f(x0)

x7=1.31899-(-0.0243)·

2-1.31899

5-(-0.0243)

x7=1.32228

f(x7)=f(1.32228)=-0.01036<0

In 7th iteration :
f(1.32228)=-0.01036<0 and f(2)=5>0

Root lies between these two points x0=1.32228 and x1=2

x8=x0-f(x0)·

x1-x0

f(x1)-f(x0)

x8=1.32228-(-0.01036)·

2-1.32228

5-(-0.01036)

x8=1.32368

The approximate root of the equation x3-x-1=0 using the Regula Falsi method is 1.32368

Newton Raphson method

The Newton-Raphson method which is also known as Newton's method, is an iterative numerical method used to find the roots of a real-valued function. This formula is named after Sir Isaac Newton and Joseph Raphson, as they independently contributed to its development. Newton Raphson Method or Newton's Method is an algorithm to approximate the roots of zeros of the real-valued functions, using guess for the first iteration (x_0) and then approximating the next iteration(x_1) which is close to roots, using the following formula.

$$x_1 = x_0 – f(x_0)/f'(x_0)$$

where,

- ***x_0*** *is the initial value of x,*
- ***$f(x_0)$*** *is the value of the equation at initial value, and*
- ***$f'(x_0)$*** *is the value of the first order derivative of the equation or function at the initial value x_0.*

Note: $f'(x_0)$ should not be zero else the fraction part of the formula will change to infinity which means f(x) should not be a constant function.

Newton Raphson Method Formula

In the general form, the Newton-Raphson method formula is written as follows:

$$x_n = x_{n-1} – f(x_{n-1})/f'(x_{n-1})$$

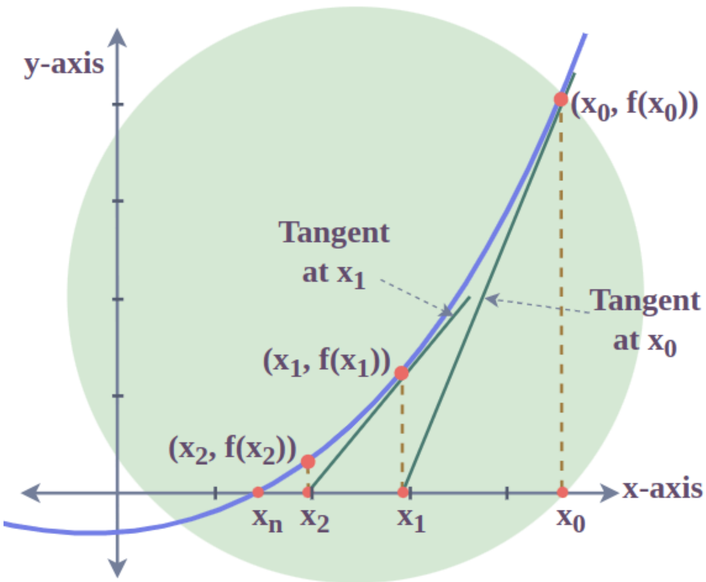
Where,

- ***x_{n-1}*** *is the estimated $(n-1)_{th}$ root of the function,*
- ***$f(x_{n-1})$*** *is the value of the equation at $(n-1)_{th}$ estimated root, and*
- ***$f'(x_{n-1})$*** *is the value of the first order derivative of the equation or function at x_{n-1}.*

Newton Raphson Method Calculation

Assume the equation or functions whose roots are to be calculated as f(x) = 0.
In order to prove the validity of Newton Raphson method following steps are followed:

Step 1: *Draw a graph of f(x) for different values of x as shown below:*



Step 2: *A tangent is drawn to f(x) at x_0. This is the initial value.*
Step 3:*This tangent will intersect the X- axis at some fixed point (x_1,0) if the first derivative of f(x) is not zero i.e.* ***$f'(x_0) \neq 0$.***
Step 4: *As this method assumes iteration of roots, this x_1 is considered to be the next approximation of the root.*
Step 5: *Now steps 2 to 4 are repeated until we reach the actual root x^*.*
Now we know that the slope-intercept equation of any line is represented as y = mx + c,
Where **m** is the slope of the line and **c** is the x-intercept of the line.
Using the same formula we, get

$$y = f(x_0) + f'(x_0)(x - x_0)$$

Here $f(x_0)$ represents the c and $f'(x_0)$ represents the slope of the tangent m. As this equation holds true for every value of x, it must hold true for x_1. Thus, substituting x with x_1, and equating the equation to zero as we need to calculate the roots, we get:

$$0 = f(x_0) + f'(x_0)(x_1 - x_0)$$

$$x_1 = x_0 - f(x_0)/f'(x_0)$$

Which is the Newton Raphson method formula.
Thus, Newton Raphson's method was mathematically proved and accepted to be valid.

Convergence of Newton Raphson Method

The Newton-Raphson method tends to converge if the following condition holds true:

$$|f(x).f''(x)| < |f'(x)|^2$$

It means that the method converges when the modulus of the product of the value of the function at x and the second derivative of a function at x is lesser than the square of the modulo of the first derivative of the function at x. The

Newton-Raphson Method has a convergence of order 2 which means it has a quadratic convergence.

Note:

Newton Raphson's method is not valid if the first derivative of the function is 0 which means f'(x) = 0. It is only possible when the given function is a constant function.

Secant Method

Secant method is also a recursive method for finding the root for the polynomials by successive approximation. It's similar to the **Regular-falsi** method but here we don't need to check $f(x_1)f(x_2)<0$ again and again after every approximation. In this method, the neighbourhoods roots are approximated by secant line or chord to the function **f(x)**. It's also advantageous of this method that we don't need to differentiate the given function **f(x)**, as we do in **Newton-raphson** method.

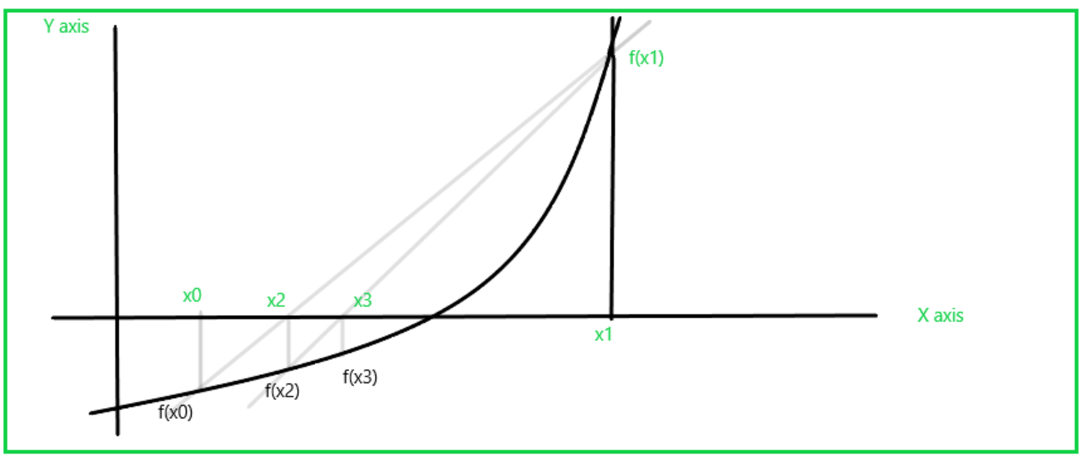


Figure – Secant Method

Now, we'll derive the formula for secant method. The equation of Secant line passing through two points is :

Here, m=slope

So, apply for $(x_1, f(x_1))$ and $(x_0, f(x_0))$

$Y - f(x_1) = [f(x_0)-f(x_1)/(x_0-x_1)] (x-x_1)$ **Equation (1)**

As we're finding root of function f(x) so, Y=f(x)=0 in Equation (1) and the point where the secant line cut the x-axis is,

$x= x_1 - [(x_0 - x_1)/ (f(x_0) - f(x_1)]f(x_1)$.

We use the above result for successive approximation for the root of function f(x). Let’s say the first approximation is **$x=x_2$:**
$x_2 = x_1 - [(x_0 - x_1)/ (f(x_0)-f(x_1))]f(x_1)$
Similarly, the second approximation would be **$x = x_3$**:
$x_3 = x_2 - [(x_1-x_2)/ (f(x_1)-f(x_2))]f(x_2)$
And so on, **till k^{th} iteration,**,
$x_{k+1} = x_k - [(x_{k-1} - x_k) / (f(x_{k-1}) - f(x_k))]f(x_k)$
Note: To start the solution of the function f(x) two initial guesses are required such that **$f(x_0)<0$** and **$f(x_1)>0$.** Usually it hasn’t been asked to find, that root of the polynomial f(x) at which **f(x) =0.** Mostly You would only be asked by the problem to find the root of the **f(x)** till two decimal places or three decimal places or four etc.

Advantages of Secant Method:

- The speed of convergence of secant method is faster than that of Bisection and Regula falsi method.
- It uses the two most recent approximations of root to find new approximations, instead of using only those approximations which bound the interval to enclose root

Disadvantages of Secant Method:

- The Convergence in secant method is not always assured.
- If at any stage of iteration this method fails.
- Since convergence is not guaranteed, therefore we should put limit on maximum number of iterations while implementing this method on computer.

Example-1 :

Compute the root of the equation $x^2e^{-x/2} = 1$ in the interval [0, 2] using the secant method. The root should be correct to three decimal places.

Solution –

$x_0 = 1.42$, $x_1 = 1.43$, $f(x_0) = -0.0086$, $f(x_1) = 0.00034$.
Apply, **secant method,** The first approximation is,
$x_2 = x_1 - [(x_0 - x_1) / (f(x_0) - f(x_1)]f(x_1)$
$= 1.43 - [(1.42 - 1.43) / (0.00034 - (-0.0086))](0.00034)$
$= 1.4296$
$f(x_2) = -0.000011$ (–ve)
The second approximation is,
$x_3 = x_2 - [(x_1 - x_2) / (f(x_1) - f(x_2))]f(x_2)$
$= 1.4296 - [(1.42 - 1.4296) / (0.00034 - (-0.000011](-0.000011)$
$= 1.4292$
Since, **x_2** and **x_3** matching up to **three decimal places,** the required root is **1.429**.

Example-2 :

A real root of the equation $f(x) = x^3 - 5x + 1 = 0$ lies in the interval (0, 1). Perform four iterations of the secant method.

Solution –
We have, $x_0 = 0$, $x_1 = 1$, $f(x_0) = 1$, $f(x_1) = -3$
$x_2 = x_1 - [(x_0 - x_1) / (f(x_0) - f(x_1))]f(x_1)$
$= 1 - [(0 - 1) / ((1-(-3))](-3)$
= 0.25.

$f(x_2) = -0.234375$
The second approximation is,
$x_3 = x_2 - [(x_1 - x_2) / (f(x_1) - f(x_2))]f(x_2)$
$=(-0.234375) - [(1 - 0.25)/(-3 - (-0.234375))](-0.234375)$
= 0.186441

$f(x_3)$
The third approximation is,
$x4 = x3 - [(x2 - x3) / (f(x2) - f(x3))]f(x3)$
$= 0.186441 - [(0.25 - 0.186441) / (-0.234375)$ **$- (0.074276)](-0.234375)$**
= 0.201736.

$f(x4) = -0.000470$
The fourth approximation is,
$x5 = x4 - [(x3 - x4) / (f(x3) - f(x4))]f(x4)$
$= 0.201736 - [(0.186441 - 0.201736) / (0.074276 - (-0.000470)](-0.000470)$
= 0.201640