Project of numerical computation(2022)

**SUBMITTED BY: AREEJ REHMAT (SP20-BSM-008)**

**SHAROZ JAHANGIR (SP20-BSM-035)**

**UMAR AKRAM(SP20-BSM-028**

**SUBMITTED TO:**

**DR. UMAIR UMER**

Bisection method:

# Bisection Method:

In Mathematics, the bisection method is a straightforward technique to find numerical solutions of an equation with one unknown. Among all the numerical methods, the bisection method is the simplest one to solve the transcendental equation.

Bisection Method Definition:

The bisection method is used to find the [roots of a polynomial](https://byjus.com/maths/roots-of-polynomials/)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 belongs 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 x2-3 = 0 for x ∈ [1, 2]

**Solution:**

Given: x2-3 = 0

Let f(x) = x2-3

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

f(x=1) = 12-3 = 1 – 3 = -2 < 0

f(x=2) = 22-3 = 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.

### Advantages of Bisection Method

* Guaranteed convergence. The bracketing approach is known as the bisection method, and it is always convergent.
* Errors can be managed. Increasing the number of iterations in the bisection method always results in a more accurate root.
* Doesn't demand complicated calculations. There are no complicated calculations required when using the bisection method. To use the bisection method, we only need to take the average of two values.
* Error bound is guaranteed. There is a guaranteed error bound in this technique, and it reduces with each repetition. Each cycle reduces the error bound by 12 per cent.
* The bisection method is simple and straightforward to programme on a computer.
* In the case of several roots, the bisection procedure is quick.

### Disadvantages of Bisection Method

* Although the Bisection method's convergence is guaranteed, it is often slow.
* Choosing a guess that is close to the root may necessitate numerous iterations to converge.
* Some equations' roots cannot be found. Because there are no bracketing values, like f(x) = x².
* Its rate of convergence is linear.
* It is incapable of determining complex roots.
* If the guess interval contains discontinuities, it cannot be used.
* It cannot be applied over an interval where the function returns values of the same sign.

### Bisection Method Problems:

The best way of understanding how the algorithm works is by looking at a bisection method example and solving it by using the bisection method formula

False Position Method:

# False Position Method

In mathematics, an ancient method of solving an equation in one variable is the **false position method**(method of false position)or**regula falsi method**. In simple words, the method is described as the trial-and-error approach of using “false” or “test” values for the variable and then altering the test value according to the result.

## False Position Method (or) Regula Falsi Method

Consider an equation f(x) = 0, which contains only one variable, i.e. x.

To find the real root of the equation f(x) = 0, we consider a sufficiently small interval (a, b) where a < b such that f(a) and f(b) will have opposite signs. According to the [intermediate value theorem](https://byjus.com/maths/intermediate-value-theorem/), this implies a root lies between a and b.

Also, the curve y = f(x) will meet the x-axis at a certain point between A[a, f(a)] and B[b, f(b)].

Now, the equation of the chord joining A[a, f(a)] and B[b, f(b)] is given by:

\(\begin{array}{l}y-f(a)=\frac{f(b)-f(a)}{(b-a)}.(x-a)\end{array} \)

Let y = 0 be the point of intersection of the chord equation (given above) with the x-axis. Then,

\(\begin{array}{l}-f(a)=\frac{f(b)-f(a)}{(b-a)}.(x-a)\end{array} \)

This can be simplified as:

\(\begin{array}{l}\frac{-f(a)(b-a)}{f(b)-f(a)}=x-a\end{array} \)

\(\begin{array}{l}\frac{af(a)-bf(a)}{f(b)-f(a)}+a=x\end{array} \)

\(\begin{array}{l}\Rightarrow x=\frac{af(a)-bf(a)+af(b)-af(a)}{f(b)-f(a)}\end{array} \)

\(\begin{array}{l}\Rightarrow x=\frac{af(b)-bf(a)}{f(b)-f(a)}\end{array} \)

Thus, the first approximation is x1 = [af(b) – bf(a)]/ [f(b) – f(a)]

Also, x1 is the root of f(x) if f(x1) = 0.

If f(x1) ≠ 0 and if f(x1) and f(a) have opposite signs, then we can write the second approximation as:

x2 = [af(x1) – x1f(a)]/ [f(x1) – f(a)]

Similarly, we can estimate x3, x4, x5, and so on.

Geometrical representation of the roots of the equation f(x) = 0 can be shown as:

**Question:**

Find a root for the equation 2ex sin x = 3 using the false position method and correct it to three decimal places with three iterations.

**Solution:**

Given equation: 2ex sin x = 3

This can be written as: 2ex sin x – 3 = 0

Let f(x) = 2ex sin x – 3

So, f(0) = 2e0 sin 0 – 3

= 0 – 3

= -3 < 0

And

f(1) = 2e1 sin 1 – 3

= 2e sin 1 – 3

= 1.5747 > 0

That means the root of f(x) lies between 0 and 1.

Let a = 0 and b = 1.

The first approximation = x1 = [af(b) – bf(a)]/ [f(b) – f(a)]

= [0(1.5747) – 1(-3)]/ [1.5747 – (-3)]

= 3/4.5747

= 0.6557

Thus, x1 = 0.6557

Now, substitute x = 0.6557 in f(x).

So, f(0.6557) = 2e0.6557 sin(0.6557) – 3

= 2.3493 – 3

= -0.6507 < 0

As we know, f(1) > 0

That means a root lies between 0.6557 and 1.

Let a = 0.6557

The second approximation = x2 = [af(x1) – x1f(a)]/ [f(x1) – f(a)]

= [0.6557(1.5747) – 1(-0.6507)]/ [1.5747 – (-0.6507)]

= (1.0325 + 0.6507)/(2.2254)

= 1.6832/2.2254

= 0.7563

Therefore, x2 = 0.7563

Let us substitute 0.7563 in f(x).

So, f(0.7563) = 2e0.7563 sin(0.7563) – 3

= 2.9239 – 3

= -0.0761 < 0

We know that f(1) > 0

Thus, a root lies between 0.7563 and 1.

Hence, the third approximation = x3 = [af(x2) – x2f(a)]/ [f(x2) – f(a)]

= [(0.7563)(1.5747) – 1(-0.0761)]/ [1.5747 – (-0.0761)]

= (1.1909 + 0.0761)/1.6508

= 1.2670/1.6508

= 0.7675

So, x3 = 0.7675

Therefore, the best approximation of the root up to three decimal places is 0.768 (up to three decimal places).

Newton’s Raphson Method:

# Newton Raphson Method

The **Newton Raphson Method** is referred to as one of the most commonly used techniques for finding the roots of given equations. It can be efficiently generalized to find solutions to a system of equations. Moreover, we can show that when we approach the root, the method is quadratically convergent.

## Newton Raphson Method Formula

Let x0 be the approximate root of f(x) = 0 and let x1 = x0 + h be the correct root. Then f(x1) = 0

⇒ f(x0 + h) = 0….(1)

By expanding the above equation using [Taylor’s theorem](https://byjus.com/maths/taylor-series/), we get:

f(x0) + hf1(x0) + … = 0

⇒ h = -f(x0) /f’(x0)

Therefore, x1 = x0 – f(x0)/ f’(x0)

Now, x1 is the better approximation than x0.

Similarly, the successive approximations x2, x3, …., xn+1 are given by

This is called Newton Raphson formula.

### Convergence of Newton Raphson Method

The order of convergence of Newton Raphson method is 2 or the convergence is quadratic. It converges if |f(x).f’’(x)| < |f’(x)|2. Also, this method fails if f’(x) = 0.

## Solved Examples

**Example 1:**

Find the cube root of 12 using the Newton Raphson method assuming x0 = 2.5.

**Solution:**

We know that, the iterative formula to find both root of a is given by:

From the given, a = 12, b = 3

Let x0 be the approximate cube root of 12, i.e., x0 = 2.5.

So, x1 = (⅓) [2x0 + 12/x02]

= (⅓) [2(2.5) + 12/(2.5)2]

= (⅓) [5 + 12/6.25]

= (⅓)(5 + 1.92)

= 6.92/3

= 2.306

Now,

x2 = (⅓)[2x1 + 12/x12]

= (1/3) [2(2.306) + 12/(2.306)2]

= (⅓) [4.612 + 12/5.3176]

= (⅓) [4.612 + 2.256]

= 6.868/3

= 2.289

Therefore, the approximate cube root of 12 is 2.289.

**Example 2:**

Find a real root of the equation -4x + cos x + 2 = 0, by Newton Raphson method up to four decimal places, assuming x0 = 0.5.

**Solution:**

Given equation: -4x + cos x + 2 = 0

x0 = 0/5

Let f(x) = -4x + cos x + 2

f’(x) = -4 – sin x

Now,

f(0) = -4(0) + cos 0 + 2 = 1 + 2 = 3 > 0

f(1) = -4(1) + cos 1 + 2 = -4 + 0.5403 + 2 = -1.4597 < 0

Thus, a root lies between 0 and 1.

Let us find the first approximation.

x1 = x0 – f(x0)/f’(x0)

= 0.5 – [-4(0.5) + cos 0.5 + 2]/ [-4 – sin 0.5]

= 0.5 – [(-2 + 2 + cos 0.5)/ (-4 – sin 0.4)]

= 0.5 – [cos 0.5/ (-4 – sin 0.5)]

= 0.5 – [0.8775/ (-4 – 0.4794)]

= 0.5 – (0.8775/-4.4794)

= 0.5 + 0.1958

= 0. 6958

Secant Method:

# Secant Method

The **secant method** is a root-finding procedure in numerical analysis that uses a series of roots of secant lines to better approximate a root of a function f.

## What is a Secant Method?

The tangent line to the curve of y = f(x) with the point of tangency (x0, f(x0) was used in Newton’s approach. The graph of the tangent line about x = α is essentially the same as the graph of y = f(x) when x0 ≈ α. The root of the tangent line was used to approximate α.

Consider employing an approximating line based on ‘[interpolation](https://byjus.com/maths/interpolation/)’. Let’s pretend we have two root estimations of root α, say, x0 and x1. Then, we have a linear function

q(x) = a0 + a1x

using q(x0) = f (x0), q(x1) = f (x1).

This line is also known as a secant line. Its formula is as follows:

## Secant Method Steps

The secant method procedures are given below using equation (1).

**Step 1: Initialization**

x0 and x1 of α are taken as initial guesses.

**Step 2: Iteration**

In the case of n = 1, 2, 3, …,

until a specific criterion for termination has been met (i.e., The desired accuracy of the answer or the maximum number of iterations has been attained).

Secant Method Convergence

If the initial values x0and x1 are close enough to the root, the secant method iterates xnand converges to a root of function f. The order of convergence is given by φ.

The convergence is particularly superliner, but not really quadratic. This solution is only valid under certain technical requirements, such as f being two times continuously differentiable and the root being simple in the question (i.e., having multiplicity 1).

There is no certainty that the secant method will converge if the beginning values are not close enough to the root. For instance, if the function f is differentiable on the interval [x0, x1], and there is a point on the interval where f’ =0, the algorithm may not converge.

Secant Method Advantages and Disadvantages

The secant method has the following advantages:

* It converges quicker than a linear rate, making it more convergent than the bisection method.
* It does not necessitate the usage of the function’s derivative, which is not available in a number of applications.
* Unlike Newton’s technique, which requires two function evaluations in every iteration, it only requires one.

The secant method has the following drawbacks:

* The secant method may not converge.
* The computed iterates have no guaranteed error bounds.
* If f0(α) = 0, it is likely to be challenging. This means that when x = α, the x-axis is tangent to the graph of y = f(x).
* Newton’s approach is more easily generalized to new ways for solving nonlinear simultaneous systems of equations.
* **Example:**
* Compute two iterations for the function f(x) = x3 – 5x + 1 = 0 using the secant method, in which the real roots of the equation f(x) lies in the interval (0, 1).
* **Solution:**
* Using the given data, we have,
* x0 = 0, x1 = 1, and
* f(x0) = 1, f(x1) = -3
* Using the secant method formula, we can write
* x2 = x1 – [(x0 – x1) / (f(x0) – f(x1))]f(x1)
* Now, substitute the known values in the formula,
* = 1 – [(0 – 1) / ((1-(-3))](-3)
* = 0.25.
* Therefore, f(x2) = – 0.234375
* Performing the second approximation, ,
* x3 = x2 – [( x1 – x2) / (f(x1) – f(x2))]f(x2)
* =(- 0.234375) – [(1 – 0.25)/(-3 – (- 0.234375))](- 0.234375)
* = 0.186441
* Hence, f(x3) = 0.074276

Fixed Point Iteration Method:

Fixed Point Iteration

The**fixed point iteration**method in numerical analysis is used to find an approximate solution to algebraic and transcendental equations. Sometimes, it becomes very tedious to find solutions to cubic, bi-quadratic and transcendental equations; then, we can apply specific numerical methods to find the solution; one among those methods is the fixed point iteration method.

The fixed point iteration method uses the concept of a fixed point in a repeated manner to compute the solution of the given equation. A fixed point is a point in the domain of a function g such that g(x) = x. In the fixed point iteration method, the given function is algebraically converted in the form of g(x) = x.

Fixed Point Iteration Method

Suppose we have an equation f(x) = 0, for which we have to find the solution. The equation can be expressed as x = g(x). Choose g(x) such that |g’(x)| < 1 at x = xo where xo,is some initial guess called fixed point iterative scheme. Then the iterative method is applied by successive approximations given by xn = g(xn – 1), that is, x1 = g(xo), x2 = g(x1) and so on.

Algorithm of Fixed Point Iteration Method

* Choose the initial value xo for the iterative method. One way to choose xo is to find the values x = a and x = b for which f(a) < 0 and f(b) > 0. By narrowing down the selection of a and b, take xo as the average of a and b.
* Express the given equation, in the form x = g(x) such that |g’(x)| < 1 at x = xo. If there more than one possibility of g(x), choose the g(x) which has the minimum value of g’(x) at x = xo.
* By applying the successive approximations xn = g(xn – 1), if f is a continuous function, we get a sequence of {xn} which converges to a point lo, which is the approximate solution of the given equation.

Important Facts

Some interesting facts about the fixed point iteration method are

* The form of x = g(x) can be chosen in many ways. But we choose g(x) for which |g’(x)|<1 at x = xo.
* By the fixed-point iteration method, we get a sequence of xn, which converges to the root of the given equation.
* Lower the value of g’(x), fewer the iterations are required to get the approximate solution.
* The rate of convergence is more if the value of g’(x) is smaller.
* The method is useful for finding the real roots of the equation, which is the form of an infinite series.
* The type of convergence seen is linear.
* **Example 1:**
* Find the first approximate root of the equation 2x3 – 2x – 5 = 0 up to 4 decimal places.
* **Solution:**
* Given f(x) = 2x3 – 2x – 5 = 0
* As per the algorithm, we find the value of xo, for which we have to find a and b such that f(a) < 0 and f(b) > 0
* Now, f(0) = – 5
* f(1) = – 5
* f(2) = 7
* Thus, a = 1 and b = 2
* Therefore, xo = (1 + 2)/2 = 1.5
* Now, we shall find g(x) such that |g’(x)| < 1 at x = xo
* 2x3 – 2x – 5 = 0
* ⇒ x = [(2x + 5)/2]1/3
* g(x) = [(2x + 5)/2]1/3which satisfies |g’(x)| < 1 at x = 1.5
* Now, applying the iterative method xn,= g(xn – 1) for n = 1, 2, 3, 4, 5, …
* For n = 1; x1 = g(xo) = [{2(1.5) + 5}/2]1/3= 1.5874
* For n = 2; x2 = g(x1) = [{2(1.5874) + 5}/2]1/3= 1.5989
* For n = 3; x3 = g(x2) = [{2(1.5989) + 5}/2]1/3= 1.60037
* For n = 4; x4 = g(x3) = [{2(1.60037) + 5}/2]1/3= 1.60057
* For n = 5; x5 = g(x4) = [{2(1.60057) + 5}/2]1/3= 1.60059
* For n = 6; x6 = g(x5) = [{2(1.60059) + 5}/2]1/3= 1.600597 ≈ 1.6006
* The approximate root of 2x3 – 2x – 5 = 0 by the fixed point iteration method is 1.6006.
* **Example 2:**
* Find the first approximate root of the equation cos x = 3x – 1 up to 4 decimal places.
* **Solution:**
* Let f(x) = cos x – 3x + 1 = 0
* As per the algorithm, we find the value of xo, for which we have to find a and b such that f(a) < 0 and f(b) > 0
* Now, f(0) = 2
* f(𝜋/2) = -3𝜋/2 – 1 < 0
* Hence, xois a value lying between 0 and 𝜋/2, for ease of calculation let us take xo = 0
* Now, we shall find g(x) such that |g’(x)| < 1 at x = xo
* cos x – 3x + 1 = 0
* ⇒ x = (cos x + 1)/3
* Then g(x) = (cos x + 1)/3 which satisfies |g’(x)| < 1 at x = 0
* Now, applying the iterative method xn,= g(xn – 1) for n = 1, 2, 3, 4, 5, …
* For n = 1; x1 = g(xo) = (cos 0 + 1)/3 = ⅔ = 0.66667
* For n = 2; x2 = g(x1) = {cos (0.66667) + 1}/3 = 0.595296
* For n = 3; x3 = g(x2) = {cos (0.595296) + 1}/3 = 0.609328
* For n = 4; x4 = g(x3) = {cos (0.609328) + 1}/3 = 0.6066778
* For n = 5; x5 = g(x4) = {cos (0.6066778) + 1}/3 = 0.607182
* For n = 6; x6 = g(x5) = {cos (0.607182) + 1}/3 = 0.607086
* For n = 7; x7 = g(x6) = {cos (0.607086) + 1}/3 = 0.607105
* For n = 8; x8 = g(x7) = {cos (0.607105) + 1}/3 = 0.607101 ≈ 0.6071
* The approximate root of cos x = 3x – 1 by the fixed-point iteration method is 0.6071.

# **What is Interpolation?**  Interpolation is the technique to estimate the value of a mathematical function, for any intermediate value of the independent variable.

# Newton Forward Difference Interpolation:

**Forward difference interpolation:**

The forward difference table for the above set of data points is given below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y | Δy | Δ2y | Δ3y | Δ4y |
| x0 | y0 | Δy0=y1-y0 | Δ2y0= Δy1- Δy0 | Δ3y0= Δ2y1- Δ2y0 | Δ4y0= Δ3y1- Δ3y0 |
| x1 | y1 | Δy1=y2-y1 | Δ2y1= Δy2- Δy1 | Δ3y1= Δ2y12- Δ2y1 |  |
| x2 | y2 | Δy2=y3-y2 | Δ2y2 = Δy3-Δy2 |  |  |
| x3 | y3 | Δy3=y4-y3 |  |  |  |
| x4 | y4 |  |  |  |  |

**NEWTON’S GREGORY FORWARD INTERPOLATION FORMULA**:



**Q. Consider the following set of data points and form a forward difference table.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | 2.0 | 2.25 | 2.50 | 2.75 | 3.0 |
| Y | 9.0 | 10.06 | 11.25 | 12.56 | 14.0 |

**Find y (2.35)?**

**Ans.** The forward difference table for the given set of data points is given below:

We have,

**x = 2.35,** ( given )

**h = x1 – x0** = 2.25 – 2.0 = **0.25**

**u = (x-x0) / h** = (2.35-2.0)/0.25 = 0.35/0.25 = **1.4**

Now, the forward difference table is as:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y | Δy | Δ2y | Δ3y | Δ4y |
| 2.0 | 9.0 | 10.06-9.0=1.06 | 1.19-1.06=0.13 | 0.12-0.13=-0.01 | 0.01+0.01=0.02 |
| 2.25 | 10.06 | 11.25-10.06=1.19 | 1.31-1.19=0.12 | 0.13-0.12=0.01 |  |
| 2.50 | 11.25 | 12.56-11.25=1.31 | 1.44-1.31=0.13 |  |  |
| 2.75 | 12.56 | 14.0-12.56=1.44 |  |  |  |
| 3.0 | 14.0 |  |  |  |  |

**y (2.35**)=9+(1.4)(1.06)+(1.4)(1.4-1)(0.13)/2+(1.4)(1.4-1)(1.4-2)(-0.01)/6+(1.4)(1.4-1)(1.4-2)(1.4-3)(0.02)\24

**y (2.35)** = 10.484 + 0.0364 + 0.00056 + 0.000448

**y (2.35)** = **10.52141**

# Newton’s Divided Difference Interpolation Formula

**Newton’s divided difference interpolation formula** is a interpolation technique used when the interval difference is not same for all sequence of values.

Suppose f(x0), f(x1), f(x2)………f(xn) be the (n+1) values of the function y=f(x) corresponding to the arguments x=x0, x1, x2…xn, where interval differences are not same.

**Then the first divided difference is given by**



**The second divided difference is given by**



and so on…  
Divided differences are symmetric with respect to the arguments i.e. **independent of the order of arguments.**  
so, **f[x0, x1]=f[x1, x0]  
f[x0, x1, x2]=f[x2, x1, x0]=f[x1, x2, x0]**  
By using first divided difference, second divided difference as so on .A table is formed which is called the divided difference table.

**Divided difference table:**

|  |  |  |  |
| --- | --- | --- | --- |
| xi | fi | F(xi,xj) |  |
| x1 | f1 |  |  |
|  |  | f[x1,x2]= f2-f1/x2-x1 |  |
| x2 | f2 |  | f[x1,x2,x3]=f[x3,x2]-f[x2,x1]/x3-x1 |
|  |  | f[x2,x3]= f3-f2/x3-x2 |  |
| x3 | f3 |  |  |

**Advantages of NEWTON’S DIVIDED DIFFERENCE INTERPOLATION FORMULA**

* These are useful for interpolation.
* Through difference table, we can find out the differences in higher order.
* Differences at each stage in each of the columns are easily measured by subtracting the previous value from its immediately succeeding value.
* The differences are found out successively between the two adjacent values of the y variable till the ultimate difference vanishes or become a constant.

**NEWTON’S DIVIDED DIFFERENCE INTERPOLATION FORMULA**



Examples:

Input: Value at 7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 5 | 6 | 9 | 11 |
| Y=f(x) | 12 | 13 | 14 | 16 |

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y=f(x) | f[xi,xj] | f[xi,xj,xk] | [xi,xj,xk,xp] |
| 5 | 12 |  |  |  |
|  |  | 1 |  |  |
| 6 | 13 |  | -1/6 |  |
|  |  | 1/3 |  | 1/20 |
| 9 | 14 |  | 2/15 |  |
|  |  | 1 |  |  |
| 11 | 16 |  |  |  |

value at 7 is 13.47.

# Newton Backward Interpolation:

# **Backward Differences**: The differences y1 – y0, y2 – y1, ……, yn – yn–1 when denoted by dy1, dy2, ……, dyn, respectively, are called first backward difference. Thus, the first backward differences are :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | ∇y | ∇2y | ∇3y |
| x0 | y0 |  |  |  |
|  |  | ∇y1=y1-y0 |  |  |
| x1 | y1 |  | ∇2y2=∇y2-∇y1 |  |
|  |  | ∇y2=y2-y1 |  | ∇3y0=∇2y3-∇2y2 |
| x2 | y2 |  | ∇2y3=∇y3-∇y2 |  |
|  |  | ∇y3=y3-y2 |  |  |
| x3 | y3 |  |  |  |

# ****NEWTON’S GREGORY BACKWARD INTERPOLATION FORMULA****:

# p=xnh

# y(x)=yn +p∇yn+p(p+1)2!- ∇2yn+p(p+1)(p+2)3!- ∇3yn+p(p+1)(p+2)(p+3)4!.....

# Find the solution using Newton’s backward difference formula:

|  |  |
| --- | --- |
| x | F(x) |
| 1891 | 46 |
| 1901 | 66 |
| 1911 | 81 |
| 1921 | 93 |
| 1931 | 101 |

# x= 1925 Solution:

# The value of table for x and y:

|  |  |
| --- | --- |
| x | Y |
| 1891 | 46 |
| 1901 | 66 |
| 1911 | 81 |
| 1921 | 93 |
| 1931 | 101 |

# 

# Newton’s backward difference interpolation method to find solution

Newton,s backward difference table is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | Y | ∇y | ∇2y | ∇3y | ∇4y |
| 1891 | 46 |  |  |  |  |
|  |  | 20 |  |  |  |
| 1901 | 66 |  | -5 |  |  |
|  |  | 15 |  | 2 |  |
| 1911 | 81 |  | -3 |  | -3 |
|  |  | 12 |  | -1 |  |
| 1921 | 93 |  | -4 |  |  |
|  |  | 8 |  |  |  |
| 1931 | 101 |  |  |  |  |

The value of x at you want to find the f(x):x=1925

h=x1-x0=1901-1891=10

p=(1925-1931)/10= -0.6

Newton’s backward interpolation formula is:

# p=xnh

# y(x)=yn +p∇yn+p(p+1)2!- ∇2yn+p(p+1)(p+2)3!- ∇3yn+p(p+1)(p+2)(p+3)4!.....

# y(1925)=101+(-0.6)x8+(-0.6)(-0.6+1)2x -4 +(-0.6)(-0.6+1)(-0.6+2)6x(-1)+(-0.6)(-0.6+1)(-0.6+2)(-0.6+3)24 x -3

# y(1925)=96.8368

# solution of newton’s backward interpolation method for y(1925) is 96.8368

**Lagrange’s Interpolation:**

# Interpolation is a method of finding new data points within the range of a discrete set of known data points

For example, in the given table we’re given 4 set of discrete data points, for an unknown function f(x) :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | 1 | 2 | 3 | 4 |
| xi | 0 | 1 | 2 | 5 |
| Yi=fi(x) | 2 | 3 | 12 | 147 |

**How to find?**   
Here we can apply the Lagrange’s interpolation formula to get our solution.   
The Lagrange’s Interpolation formula:

  
If, y = f(x) takes the values y0, y1, … , yn corresponding to x = x0, x1 , … , xn then,

This method is preferred over its counterparts like Newton’s method because it is applicable even for unequally spaced values of x.  
We can use interpolation techniques to find an intermediate data point say at x = 3. 

**Advantages of Lagrange Interpolation:**

* This formula is used to find the value of the function even when the arguments are not equally spaced.
* This formula is used to find the value of independent variable x corresponding to a given value of a function.

**Disadvantages of Lagrange Interpolation:**

* A change of degreein Lagrangian polynomial involves a completely new computation of all the terms.
* For a polynomial of high degree, the formula involves a large number of multiplications which make the process quite slow.
* In the Lagrange Interpolation, the degree of polynomial is chosen at the outset. So, it is difficult to find the degree of approximating polynomial which is suitable for given set of tabulated points.