Unit-1 Solution of Non-linear Equations (8 hr)

Numerical Computing :-

→ ~ is an approach for solving complex mathematical problems using only simple arithmetic operations. 1)

→ The approach involves formulation of mathematical models of physical situations that can be solved with arithmetic operations.

→ It requires development, analysis and use of algorithms. 2.)

→ The traditional numerical computing methods usually deals with the following topics :-

* finding the roots of equations
* Solving system of linear algebraic equations
* Interpolation and regression analysis
* Numerical Integration
* Numerical Differentiation
* Solution of differential equation
* Boundary value problems 3.)
* Solution of matrix problems

Process of Numerical Computing

→ Numerical computing involves formulation of mathematical models of physical problems that can be solved by using basic mathematical operations.

→ The problem of numeric computing can be roughly divided into the following four phases. 4.)

1.) Formulation of mathematical model
2.) Construction of approximate numerical method
3) Implementation of the method to obtain a solution
4) Validation of the solution

Scanned by CamScanner

Characteristic of Numerical Computing

Accuracy

The results we obtain must be sufficiently accurate to serve the purpose for which the mathematical model was build.

Rate of Convergence

Many numerical method are based on the idea of an iterative process. This process involves generation of a sequence of approximations with the hope that the process will converge to the required solution. The certain method converge faster than others. Some method maynot converge at all. It is therefore important to test for convergence before a method is used

Numerical Stability

In Some cases errors tend to grow expotentially with disastrous computational result. A computing process that exhibits such exponential growth is said to be numerically unstable. We must choose methods that are not only fast but also stable.

Efficiency

A method that required less computing time and less programming effort and yet achieves the desired accuracy is always preffered.

Scanned by CamScanner

Types of Error in Numerical Computing

i) In any numerical analysis error will take/arise during the calculation.

ii) Error may come in various forms and sizes; some are avoidable and some are not.

iii) To establish good numerical computing system, we need to minimize the errors as per our requirement.

* True Error

It is denoted by E_t, is the difference between the true value (also called the exact value) and the approximate value.

i.e. True error = True value − Approximate Value

Eg :-

The derivative formula for function $f(x)$ at a particular value of x can be approximately calculated by

$$f'(x) \simeq \frac{f(x+h) - f(x)}{h}$$

* Relative True Error (E_r)

It is denoted by E_r and is defined as the ratio between the true error and the true value.

i.e. Relative True error = $\dfrac{\text{True Error}}{\text{True Value}}$

Eg :- true value = 9.514

true error = −0.75061 then,

Scanned by CamScanner

Relative true error $(E_r) = \frac{\text{true error}}{\text{true value}}$

$$= -\frac{0.75061}{9.514}$$

$$= -0.078895$$

Relative true error can also be represented in percentage, for this example

$$E_r = -0.078895 \times 100\%$$

$$= -7.8895\%$$

Absolute relative true errors may also need to be calculated in such case

$$|E_r| = |-0.078895|$$

$$= 0.78895 \text{ or } 7.8895\%$$

* Approximate Error

True errors can be calculated only if true values are known but mostly we dont know true values. In such cases we want to find approximate value. When we are solving a problem numerically, we may only have access to approximate value. It is denoted by 'ε_a' and is defined as the difference between the present approximation and previous approximation.

i.e. Approximate Error (ε_a) = Present approximation − previous approximation.

Eg :- The deviation of function $f(x)$ at a particular value of x can be calculated approximately by

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Scanned by CamScanner

* Relative Approximate Error (E_{ra})

Relative approximate Error is denoted by E_{ra} and is defined as the ratio between the approximate error and the present approximation.

i.e. $\sim (E_{ra}) = \dfrac{\text{Approximate Error}}{\text{Present approximation}}$

Eg :- From above example;

Approximate error = -0.38474

Present approximation = 9.8799

Then, the relative approximate error is calculated as,

$$E_{ra} = \frac{\text{approximate error}}{\text{present approximation}}$$

$$= \frac{-0.38474}{9.8799}$$

$$= -0.03894$$

$\sim$ can also be represented in percentage as

$$E_{ra} = -0.03894 \times 100\%$$

$$= -3.894\ \%$$

Absolute relative approximate error may also need to be calculated. For this example

$$|E_{ra}| = |0.03894|.$$

$$= 0.3894 \sim 3.894\ \%$$

Scanned by CamScanner

Q1) For $f(x) = 7e^{0.5x}$ and $h = 0.3$, find the approximate value of $f'(x)$, the true value of $f'(x)$ and the true error.

Soln

Approximate value of derivative is calculated by using formula

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Now, $x = 2$ & $h = 0.3$

$$f'(x) \simeq \frac{f(2+0.3) - f(2)}{0.3}$$

$$= \frac{f(2.3) - f(2)}{0.3}$$

$$= \frac{7e^{0.5\times 2.3} - 7e^{0.5\times 2}}{0.3}$$

$$= 10.256$$

$\therefore f'(2) \simeq 10.256$

Now

Exact value of $f'(x)$ at $x = 2$ can be calculated by using our knowledge of differential calculus;

$$f(x) = 7e^{0.5x}$$

Then

$$f'(x) = 7\times 0.5 \times e^{0.5x}$$

So $f'(2) = 7\times 0.5 \times e^{0.5\times 2}$

$$= 9.514$$

$\therefore$, true error = true value − approximate value

$$= 9.514 - 10.265$$

$$E_t = -0.75061$$

Scanned by CamScanner

Q2) For $f(x) = 7e^{0.5x}$ and $x=2$. Find the following

a) $f'(2)$ using $h = 0.3$

b) $f'(2)$ using $h = 0.15$

c) Approximate error for the value of $f'(2)$

Soln

The approximate Expression for the derivate of a function is

$$f'(x) \simeq \frac{f(x+h) - f(x)}{h}$$

a) for $x=2$ and $h=0.3$

$$f'(2) \simeq \frac{f(2+0.3) - f(2)}{0.3}$$

$$= \frac{7e^{0.5\times 2.3} - 7e^{0.5\times 2}}{0.3}$$

$$= 10.285$$

b) Similarly $x=2$ and $h = 0.15$

$$f'(2) \simeq \frac{f(2+0.15) - f(2)}{0.15}$$

$$= \frac{7e^{0.5\times 2.15} - 7e^{0.5\times 2}}{0.15}$$

$$= 9.8799$$

c) The approximate error (E_a) is

E_a = Present approximation − Previous approximation

$= 9.8799 - 10.265$

$= 0.38474$

Scanned by CamScanner

Source of Errors

Errors in solving an engineering or science problem can arise due to several factors. First, the errors may be in the modeling technique i.e. A mathematical model based on using assumptions that are not acceptable. Second errors may arise from mistakes in programs themselves or in the measurement of physical quantities but in application of numerical methods, the two errors we need to focus are :-

i) Round off error

→ A computer can only represent numbers approximately

→ It occurs when a fixed number of digits are used to represent exact numbers

→ For example, a number like $1/3$ may be represented as 0.333333 on a PC. Then the round off error in this case is $\frac{1}{3} - 0.333333 = 0.0000033$

ii) Truncation Error

→ It is defined as the error caused by truncating mathematical procedure.

→ For example, the mathematical series for e^x is given as

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

→ This series has an infinite number of terms but when using this series to calculate e^x, only a finite number of terms can be used.

→ In this case, if one uses three terms to calculate e^x, then

$$e^x \simeq 1 + x + \frac{x^2}{2!}$$, then the error is

Scanned by CamScanner

$$e^x - \left(1 + x + \frac{x^2}{2!}\right) = \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

Propagation of Errors :-

→ If a calculation is made with numbers that are not exact then the calculation itself will have an error

→ Propagation of error is the effect of variable uncertainties (or error) on the uncertainty of a function based on them.

→ When the variable are the values of experimental measurement limitations which propagate to the combination of variables in the function.

Eg:- Find the bounds for the propagation error in adding two numbers, if one is calculating $x+y$ where, $x = 1.5 \pm 0.05$ & $y = 3.4 \pm 0.04$.

Soln

By looking at the number, the maximum positive value of x and y are $x = 1.55$ and $y = 3.44$.

Hence,

$$x + y = 1.55 + 3.44 = 4.99$$ is the maximum value of $x+y$.

The minimum possible value of x and y are $x = 1.45$ & $y = 3.66$

Hence,

$$x + y = 1.45 + 3.36 = 4.81$$ is the max. value of $x+y$

Hence,

$$4.81 \leq x + y \leq 4.99$$

Scanned by CamScanner

Taylor Theorem

$$f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2!} + f'''(x)\frac{h^3}{3!} + f^{iv}(x)\frac{h^4}{4!} + \dots \quad \text{— (1)}$$

provided all derivatives of $f(x)$ and are continuous between x and $x+h$. Equation (1) can also rewritten as:

$$y(h+x) = f(h) + f'(h)x + f''(h)\frac{x^2}{2!} + f'''(h)\frac{x^3}{3!} + f^{iv}(h)\frac{x^4}{4!} + \dots \quad \text{— (2)}$$

If we put $h=0$ in eqn (ii) we get,

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} + \dots \quad \text{— (3)}$$

This eqn (3) is called Maclaurian series.

Eg:- Derive the Maclaurian series of $\sin x$.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Soln

Maclaurian series is simply a Taylor series for the point $h=0$

$f(x) = \sin(x)$	$f(0) = 0$
$f'(x) = \cos x$	$f'(0) = 1$
$f''(x) = -\sin x$	$f''(0) = 0$
$f'''(x) = -\cos x$	$f'''(0) = -1$
$f^{iv}(x) = \sin x$	$f^{iv}(0) = 0$
$f^{v}(x) = \cos x$	$f^{v}(0) = 1$

Using the Maclaurian Series;

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} + f^{iv}(0)\frac{x^4}{4!} +$$

$$= 0 + 1x + 0\cdot\frac{x^2}{2!} + (-1)\frac{x^3}{3!} + 0.\frac{x^4}{4!} + 1.\frac{x^5}{} +$$

Scanned by CamScanner

$$= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Eg:- Find the value of $e^{0.25}$ using the first five terms of the Maclaurian series.

Sol^n

$$f(x) = e^x$$

Maclaurian series is

$$e^x = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} + f^{iv}(0)\frac{x^4}{4!} + \dots$$

Here, $f(x) = e^x$ $\qquad f(0) = e^0 = 1$ —①

$f'(x) = e^x$ $\qquad f'(0) = 1$

$\vdots$

$f^{v}(x) = e^x$ $\qquad f^{v}(0) = 1$

Now, eq^n ① becomes

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

$$e^{0.25} \simeq 1 + 0.25 + \frac{0.25^2}{2!} + \frac{0.25^3}{3!} + \frac{0.25^4}{4!}$$

$$= 1.2840$$

Note:- The exact value of $e^{0.25}$ is also 1.2840

Scanned by CamScanner

Errors in Taylor series

→ The taylor series has infinite terms, so whenever we are using a Taylor series to calculate the value of a function it is being calculated approximately.

→ The taylor polynomial of order n of a function $f(x)$ with $(n+1)$ continuous derivatives is the domain $[x, x+h]$ is given by

$$f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2!} + \dots f^n(x)\frac{h^n}{n!} + R_n(x)$$

Whenever the remainder is given by

$$R_n(x) = \frac{(x-h)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

where, $x < c < x+h$

That is, c is some point in the domain $(x, x+h)$

Eg:- The Taylor series of e^x at point $x=0$ is given by

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

a) What is the truncation (true) error in the representation of e^1 if only four terms of the series are used?

b) Use the remainder theorem to find the bounds of the truncation error.

Soln/

If only four terms of series are used, then

$$e^x \simeq 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$$

$$e^1 \simeq 1 + 1 + \frac{1^2}{2!} + \frac{1^3}{3!}$$

$$= 2.66667$$

Scanned by CamScanner

The truncation (true) error would be the unused terms of the Taylor's series, which are,

$$E_t = \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots$$

$$= \frac{1^4}{4!} + \frac{1^5}{5!} + \cdots$$

$$\simeq 0.0516152$$

Now, error bounds by using remainder theorem is:

$$f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2!} + \cdots + f^{n}(x)\frac{h^n}{n!} + R_n(x)$$

where

$$R_n(x) = \frac{(x-h)^{n+1}}{(n+1)!} f^{n+1}(c), \quad x < c < (x+h)$$

c is some point in the domain $(x, x+h)$

So, in this case, we are using four terms of the Taylor series, the remainder is given by $(x = 0, h = 3)$

$$R_3(x) = \frac{(0-1)^{3+1}}{(3+1)!} f^{3+1}(c)$$

$$= \frac{1}{4!} f^4(c) = \frac{e^c}{24}$$

Then the error bound is in between

$$\frac{e^0}{24} < R_3(1) < \frac{e^1}{24}$$

$$\frac{1}{24} < R_3(1) < \frac{e}{24}$$

$$0.041667 < R_3(1) < 0.113261$$

So, the bound of error is less than 0.113261 which does occur with the calculated error of 0.0516152

Mean Value Theorem

→ Suppose f is a continuous function on $[a, b]$ and differentiable on (a, b)

→ Draw the secant line joining the point $(a, f(a))$ to the point $(b, f(b))$.

→ Clearly there's a point c where $a < c < b$ such that the tangent line to the graph of f at $x = c$ is parallel to the secant line & has the same slope.

→ The slop of the secant line is $[f(b) - f(a)] / (b-a)$ and that of the tangent line is $f'(c)$

So,

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$

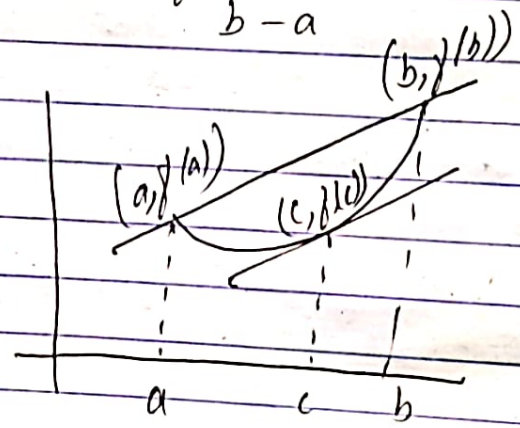


fig:- Graphical Interpretation of MVT.

Eg- Consider $f(x) = x^3 + 3x^2$ on the interval $-5 \le x \le 1$. Since f is a polynomial, f is continuous on $-5 \le x \le 1$ and differentiable on $-5 < x < 1$

$$\frac{f(1) - f(-5)}{1 - (-5)} = \frac{4 + 50}{6} = 9$$

So, we should be able to find a number c between -5 & 1 such that $f'(c) = 9$

Scanned by CamScanner

Date
Page

$f'(x) = 3x^2 + 6x$, so $f'(c) = 3c^2 + 6c$

Now,

$f'(c)$ equal to 9 and solve for c

we get

$$3c^2 + 6c = 9$$
$$\Rightarrow c^2 + 2c = 3$$
$$\Rightarrow c^2 + 2c - 3 = 0$$
$$\Rightarrow c = -3 \text{ or } c = 1$$

Since, $c = 1$ is not in the interval $-5 < x < 1$ but $c = -3$ is. Therefore, $c = -3$ is a number satisfying the conclusion of the M.V.T.

Scanned by CamScanner

Solving Non-linear Equation

Non-linear Equations

Any function of 1 variable which doesnot graph as a straight line in two dimensions or any functions of two variables which doesnot graph as a plane in three dimensions can be said to be non-linear.

Types of Non-linear equation.

i) Algebraic Equation

An equation of type $y = f(x)$ is said to be algebraic if it can be expressed in the form

$$f_n x_n + f_{n-1} x_{n-1} + f_{n-2} x_{n-2} + \dots + f_1 x_1 + f_0 = 0$$

Eg :-

→ $3x + 5y - 21 = 0$ (linear)

→ $2x + 5xy - 25 = 0$ (non-linear)

→ $x^3 + xy - 3y^3 = 0$ (non-linear)

ii) Polynomial Equations

Polynomial equations are a simple class of algebraic eq^n that are represented as follows

$$a_n x^n + a_{n-1} x_n^{n-1} + \dots + a_1 x + a_0 = 0.$$

Eg :-

$$5x^5 + x^3 + 2x^2 + 5 = 0$$

iii) Transcendental equation

A non-algebraic eq^n is called ~. These include trigonometric, exponential, & logarithmic function

Eg :- → $2\sin x - x = 0$ → $x - e^{1/x} = 0$

→ $\log x^2 - 1 = 0$

Scanned by CamScanner

Methods of Solving non-linear Equations

There are number of ways to find the roots of non-linear equations. They include

i) Direct analytical Method.
ii) Graphical Method
iii) Trial & Error method
iv) Iterative method.

Direct Analytical Method

In this method equations are solved by using derived formulas. We just need to apply the value of input parameters in the equation.

Eg:-

Solution of algebraic equation $ax^2 + bx + c = 0$ is given by the formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Solve the equation $2x^2 + 5x + 2 = 0$

Sol^n

Here, $a = 2$, $b = 5$, $c = 2$

Now, First Root, $x = \frac{-5 \pm \sqrt{5^2 - 4 \times 2 \times 2}}{2 \times 2}$

$$= \frac{-5 + \sqrt{25 - 16}}{4}$$

$$= \frac{-5 + 3}{4}$$

$$= \frac{-2}{4} = \frac{-1}{2}$$

Scanned by CamScanner

Second Root, $x = \frac{-5 - \sqrt{5^2 - 4\times2\times2}}{2\times2}$

$= \frac{-5 - \sqrt{25-16}}{4} = \frac{-5-3}{4} = \frac{-8}{4}$

$= -2$

Graphical Method

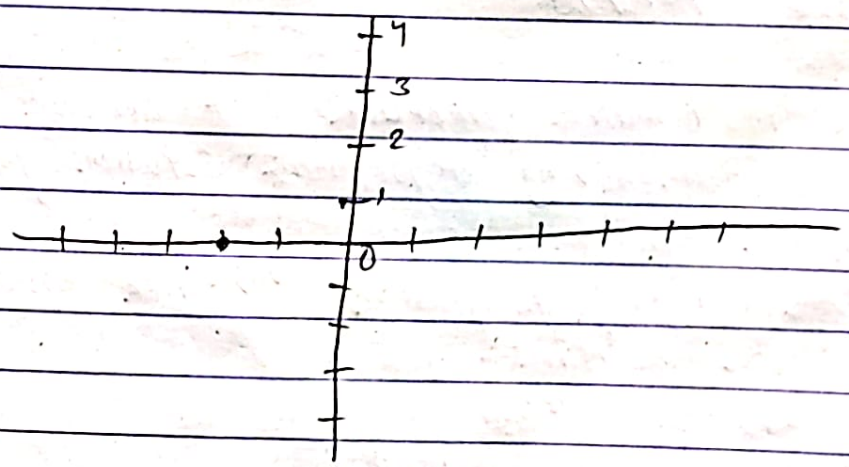
This is the simplest method to determine the root of an equation $f(x) = 0$. The procedure is quite straight-forward

→ Plot the function $f(x)$

→ Observe when it crosses the x-axis, this point represents the value for which $f(x) = 0$.

Eg:-

Solve the equation $2x^2 + 5x + 2 = 0$ by using graphical method.



x

y

$81 = 6 + 75$

Scanned by CamScanner

Trial & Error Method

→ In this method we made a series of guesses for x & then evaluate $f(x)$ at that x if it is close to zero, it is one of the approximate root of the given non-linear equations.

→ Otherwise, made another guess for x and repeat the same process again until the x for which $f(x)=0$ is found.

Eg- Solve the equation $f(x) = 2x^2+x-6 = 0$

Sol^n/

Step 1 Guess $x=0$, $f(x) = -6$

Step 2 Guess $x=1$, $f(x) = -3$

Step 3 Guess $x=-2$ $f(-2) = 0$

So, the solution of above eq^n is -2

Iterative Method

→ It is the mathematical procedure that generates a sequence of improving approximate solutions for a class of problem.

→ There are many ways of solving non-linear equations using iterative method :-

* Bisection Method
* Newton - Raphson Method
* Secant Method
* Fixed point Method
* False ~~point~~ position Method.

Scanned by CamScanner

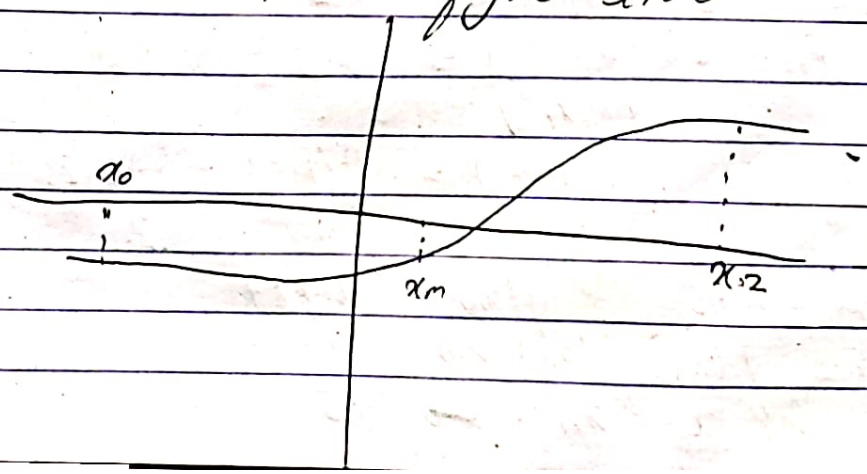
Bisection Method

- The bisection method is one of the simplest & most reliable iterative method for the solution of non-linear equations
- This menthod is also known as binary chopping or half-internal method.
- It relies on the fact that if $f(x)$ is real and continuous in the internal $a < x < b$ and $f(a)$ and $f(b)$ are of opposite signs. i.e. $f(a)\, f(b) < 0$
  then there is at least one real root in the interval between a & b.
- Let $x_1 = a$ & $x_2 = b$. Let us define another point x_m to be the mid point between a & b. That is

$$x_m = \frac{x_1 + x_2}{2}$$

- Now, there exist the following

* If $f(x_m) = 0$, we have a root at x_m
* If $f(x_1)\, f(x_m) < 0$ there is a root between x_1 & x_m
* If $f(x_2)\, f(x_m) < 0$ there is a root between x_2 & x_m
* It follows that by testing the sign of the function at midpoint, we decide which part of the interval contains the root.
* This is illustrate in the figure below



Scanned by CamScanner

Advantages of Bisection Method

→ The bisection method is always convergent, since the method brackets the roots, the method is guaranteed to converge

→ As iterations are conducted the interval gets halved. So, one can guaranteed the decrease in the error in the solution of the equation.

Drawbacks of Bisection Method

→ The convergence of bisection method is slow as it is simply based on halving the interval.

→ If one the initial guess is closes to the root, it will take larges no. of iterations to reach the root.

Eg:- Solve the equation $f(x) = 3x^2 - 6x + 2 = 0$

Sol^n/

Assume initial guess $x_1 = 1$ & $x_2 = 2$

Iteration 1: $x_1 = 1$ $\quad f(1) = 3 - 6 + 2 = -1$

$x_2 = 2$ $\quad f(2) = 3 \times 4 - 6 \times 2 + 2 = 2$

$x_m = 1.5$ $\quad f(x_m) = f(1.5) = -0.25$

Since $f(x_m) f(x_2) < 0$

So, $x_1 = x_m = 1.5$

Error $= \left|\frac{x_2 - x_1}{x_2}\right| = 0.5$

Iteration 2: $x_1 = 1.5$ $\quad f(x_1) = f(1.5) = -0.25$

$x_2 = 2$ $\quad f(x_2) = f(2) = 2$

$x_m = 1.75$ $\quad f(x_m) = 0.687$

Since $f(x_1) f(x_m) < 0$,

So, $x_2 = x_m = \cancel{0.6}\ 1.75$

Scanned by CamScanner

$$\text{Error} = \left|\frac{x_2 - x_1}{x_2}\right| = 0.25$$

Iteration 3 : $x_1 = 1.5$ $\quad f(x_1) = f(1.5) = -0.25$

$x_2 = 1.75$ $\quad f(x_2) = f(1.75) = 0.687$

$x_m = 1.625$ $\quad f(x_m) = f(1.625) = 0.171$

Since $f(x_1) f(x_m) < 0$

So, $x_2 = x_m = 1.625$

The above sol^n is continuing in the table below.

Iteration	x_1	x_2	x_m	$f(x_1)$	$f(x_2)$	$f(x_m)$	Error
4	1.5	1.625	1.563	-0.25	0.171	-0.051	0.077
~~5~~	~~1.0 0.051~~	~~0.171~~					
5	1.563	1.625	1.594	-0.051	0.171	0.058	0.038
6	1.563	1.594	1.578	-0.051	0.058	0.003	0.020
7	1.563	1.578	1.5705	-0.051	0.003	-0.024	0.010

Since, the error is below the specified limit So, Root = 1.570

Eg:2 Solve the following equation by using bisection method correct upto 2 significant digit $x^2 - 4\cos x = 0$

Sol^n

let $x_1 =$ $\quad x_2 =$

Convergence of Bisection Method

→ In bisection method, we choose a mid point x_m in the interval between x_1 & x_2.

→ Depending upon the sign of $f(x_1)$, $f(x_2)$ & $f(x_m)$, x_1 or x_2 is set equal to x_m, such that next interval contains the root.

→ In either case, the interval containing the root is reduced by the factor of 2.

→ If the procedure is repeated n times, then the interval combining the root is reduced to the size.

$$\frac{x_2 - x_1}{2^n} = \frac{\Delta x}{2^n}$$

After n iterations, the root must lie within $\pm \Delta x / 2^n$ of our estimate. This means that the error bound at n^{th} iteration is

$$E_n = \left| \frac{\Delta x}{2^n} \right|$$

Similarly,

$$E_{n+1} = \frac{E_n}{2}$$

i.e. error decreases linearly with each step by a factor of 0.5.

→ Therefore bisection method is linearly convergent.

Scanned by CamScanner

Date
Page

Algorithm

→ Decide initial values for x_1 and x_2 and stopping criterion, E
→ Compute $f_1 = f(x_1)$ & $f_2 = f(x_2)$
→ If $f_1 * f_2 > 0$, x_1 & x_2 do not bracket the root & goto 7
→ Compute $x_0 = (x_1 + x_2)/2$ and compute $f_0 = f(x_0)$
→ If $f_1 \times f_2 < 0$ then
    set $x_2 = x_0$
else set $f(x_1) = f(x_0)$
    set $x_1 = x_0$
    set $f_1 = f_0$
→ If absolute value of $(x_2 - x_1)/x_2$ is less than error E, then
    root $= (x_1 + x_2)/2$
    write the value of root
    go to step 7
else
    goto step 4
→ Stop

Program (Bisection Method)

Write 'C' program to implement bisection method to find the root of non-linear equation

Scanned by CamScanner

Date
Page

Scanned by CamScanner

Newton - Rapson Method

Newton Raphson method is based on the principle that if the initial guess of the root of $f(x) = 0$ at x_i and if one draws the tangent to the curve at $f(x_i)$, then the point x_{i+1} where tangent crosses the x-axis is an improved estimate of the ~~bet~~ root as shown in figure below.

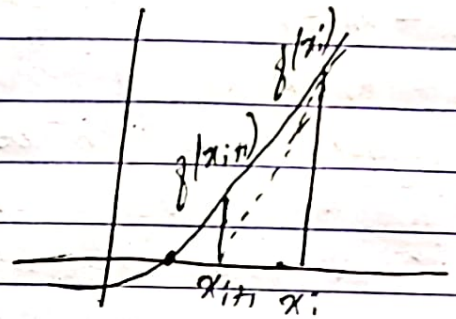


fig:- Geometrical Interpretation of Newton Raphson Method

Using the definition of the slope of a function at $x = x_i$

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

If $f(x_i)$ have root at x_{i+1} then $f(x_{i+1}) = 0$ Putting this in above eqn we get

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

This gives

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

→ This equation ~~gi~~ is called the Newton Raphson formula for solving non-linear equations of the form $f(x) = 0$

→ So starting from initial guess x_1, we can find the next guess x_{i+1} by using above equation.

Scanned by CamScanner

→ We can repeat this process until the root within a desirable error tolerence is found.

Deriving Newton-Raphson formula from Taylor Serie

→ let h is an smal increment in x_n to get next estimate of root

$$x_{n+1} - x_n = h$$

$$x_{n+1} = x_n + h$$

→ We know that Taylor serie can be written as;

$$f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2!} + \dots + f \dots$$

→ Since h is very small, we can neglect the terms containing second order & higher order terms. Then the above eqn can be written as

$$f(x+h) = f(x) + f'(x)h$$

$$f(x_n+h) = f(x_n) + f'(x_n)h$$

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

→ If x_{n+1} is the root of the given polynomial

$$\Rightarrow f(x_{n+1}) = 0$$

$$\Rightarrow f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$

→ Solving the above equation to get the value of x_{n+1}, we get,

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

→ This equation is called Newton-Raphson formula.

Scanned by CamScanner

NQ) # Solve the eqⁿ $x^2+4x-9=0$ by using Newton-Raphson method. The solution correct upto 4 significant digit.

Sol^n

$f(x) = x^2+4x-9$

$f'(x) = 2x+4$

let us assume the initial guess is 4.0

Iteration 1:

$x_0 = 4$

$f(x) = 23 \quad f'(x) = 12$

$x_1 = 4 - \frac{23}{12} = 2.08$

$\text{Error} = |(x_1 - x_0)/x_1| = |(2.08-4)/2.08| =$

$= 0.9200$

Iteration 2:

$x_0 = 2.0833$

$f(x) = 3.678$

$f'(x) = 8.166$

$x_1 = 1.633$

$\text{Error} = 0.275$

Continuing above sol^n is table as

Iteration	x	$f(x)$	$f'(x)$	Error
3	1.605	0.202	7.266	0.015
4	1.606	0.001	7.211	0.017
5	1.606	0.000	7.211	0.000

Itr	x_0	x_1	$f(x_0)$	$f'(x_0)$	Error
3	1.633	1.605	0.202	7.266	0.015

Scanned by CamScanner

Eg: 2 Find the root of the following eqn by using Newton-Raphson method.

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

Sol^n/

Here, $f(x) = x^2 - 3x + 2$

$f'(x) = 2x - 3$

Initial guess $x_0 = 3$

Iteration	x_0	x_1	$f(x_0)$	$f'(x_0)$	Error
1	3	2.33	2	3	0.287
2	2.33	2.066	0.431	1.66	0.127
3	2.066	2.003	0.070	1.132	0.031
4	2.003	2.000	0.0003	1.006	0.001

Convergence of Newton-Raphson Method

→ Let x_n be an estimate of a root of the function $f(x)$. If x_n and x_{n+1} are close to each other, then using Taylor series expansion, we can write.

→ $$f(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{f''(R)}{2!}(x_{n+1} - x_n)^2 \quad \text{—(i)}$$

where R lies somewhere in the interval x_n to x_{n+1} & third & higher order have been dropped.

→ Let us assume the exact root of $f(x)$ is x_r. Then $x_{n+1} = x_r$. Therefore $f(x_{n+1}) = 0$ and substituting these values in equation (i), we get

$$0 = f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{f''(R)}{2}(x_{n+1} - x_n)^2 \quad \text{—(ii)}$$

→ Dividing eqn (ii) by $f'(x_n)$, we get

Scanned by CamScanner

$$0 = \frac{f(x_n)}{f'(x_n)} + x_{n+1} - x_n + \frac{f''(R)}{2f'(x_n)} (x_{n+1} - x_n)^2$$

$$x_r - \left(x_n - \frac{f(x_n)}{f'(x_n)}\right) + \frac{f''(R)}{2f'(x_n)} (x_r - x_n)^2 = 0 \quad \text{——— (iii)}$$

→ We know that Newton's iterative formula is

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

→ Substituting this in equation ② we get

$$x_r - x_{n+1} = -\frac{f''(R)}{2f'(x_n)} (x_r - x_n)^2 \quad \text{——— (iv)}$$

→ We know that error estimate in x_{n+1} is given by

$$e_{n+1} = x_r - x_{n+1}$$

Similarly, $e_n = x_r - x_n$

Now, eqⁿ (iv) becomes

$$e_{n+1} = -\frac{f''(R)}{2f'(x_n)} e_n^2 \quad \text{——— (v)}$$

Eqⁿ (v) shows that the error is roughly proportional to the square of the error in the previous equation. Therefore the newton Raphson Method is said to have quadratic convergence.

Scanned by CamScanner

Newton-Raphson Algorithm

→ Assign an initial guess to x, say x_0 & precision ϵ

→ Evaluate $f(x_0)$ & $f'(x_0)$.

→ Find the improved estimate of x_0

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

→ Check for the accuracy of latest estimate

- Compare the relative approximate error with predefined value ϵ
- if $|(x_1 - x_0)/x_1| < \epsilon$ stop, otherwise continue

→ Replace x_0 by x_1 & repeat step 3 & 4

Program:- Write C program to implement Newton-Raphson method to find the solution of non-linear equation.

Scanned by CamScanner

Secant Method

It uses two initial estimates but doesnot require to bracket the root.

Geometrical representation of secant method is as shown in figure below

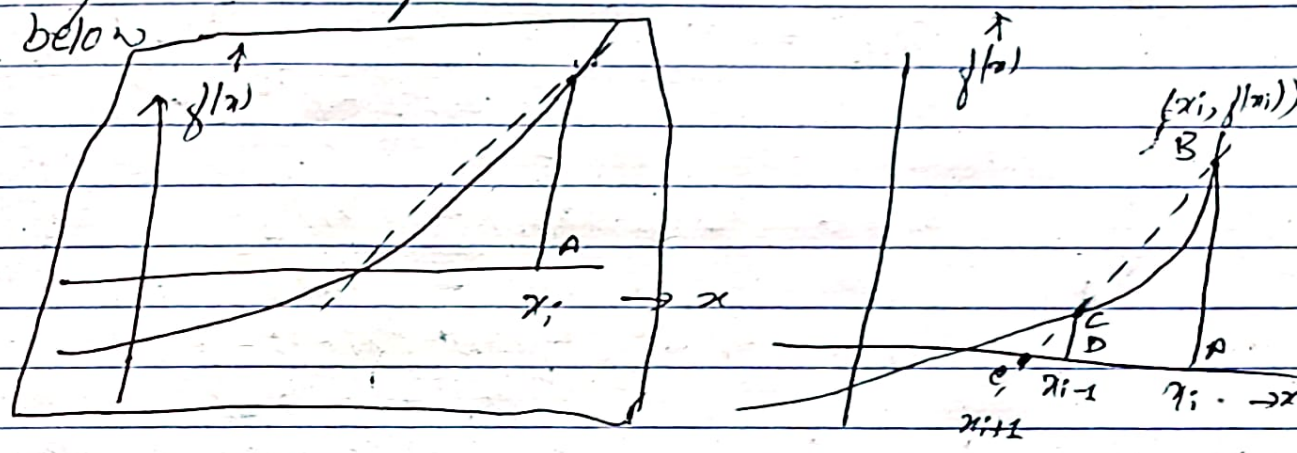


fig:- Geometrical method of secant method

Let us consider two initial guesses x_{i-1} & x_i & draw a secant line between $f(x_i)$ & $f(x_{i-1})$ passing through the x-axis at x_{i+1}.

Now, ABE & DCE are similar triangles & therefore, following relation holds

$$\frac{AB}{DC} = \frac{AE}{DE}$$

or, $$\frac{AB}{AE} = \frac{DC}{DE}$$

or, $$\frac{f(x_i)}{x_i - x_{i+1}} = \frac{f(x_{i-1})}{x_{i-1} - x_{i+1}}$$

Rearranging the terms, we get

$$x_{i+1} = \frac{f(x_{i-1})\,x_i - f(x_i)\,x_{i-1}}{f(x_{i-1}) - f(x_i)}$$

Scanned by CamScanner

Adding and Subtracting $f(x_i)x_i$ in numerator of above equation we get,

$$x_{i+1} = \frac{f(x_{i-1})x_i - f(x_i)x_{i-1} + f(x_i)x_i - f(x_i)x_i}{f(x_{i-1}) - f(x_i)}$$

$$x_{i+1} = \frac{x_i\,[f(x_{i-1}) - f(x_i)] - f(x_i)\,(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

$$x_{i+1} = \frac{x_i\,[f(x_{i-1}) - f(x_i)]}{[f(x_{i-1}) - f(x_i)]}$$

$$x_{i+1} = x_i - \frac{f(x_i)\,(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

This equation is called secant formula. ☆

N-Eg ☆ Solve the equation $2x^2 + 4x - 10 = 0$ with error precision 0.04

Soln

Let us assume that initial guesses $x_1 = 2$ & $x_2 = 6$ & specified error precision $E = 0.04$

Iteration 1:- $x_1 = 2$ $f(x_1) = 6$

$x_2 = 6$ $f(x_2) = 86$ $x_3 = x_2 - \frac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)}$

$x_3 = 1.7$ Error $= \frac{x_3 - x_2}{x_3}$

$= \frac{1.7 - 6}{1.7} = 2.529$

Iteration 2:-

$x_1 = 6$ $f(x_1) = 86$

$x_2 = 1.7$ $f(1.7) = 2.58$

$x_3 = 1.588$ Error $= 0.828$

Scanned by CamScanner

Iteration 3 :-

$x_1 = 1.7$ $\quad f(x_1) = 2.58$

$x_2 = 1.568$ $\quad f(x_2) = 1.189$

$x_3 = 1.4551$ $\quad$ Error $= 0.0775$

Continuing above solution in table as

Iteration	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	Error
4	1.568	1.4551	1.4496	1.189	0.055	0.003

Thus the root is 1.4496

Solve the equation $\cos x - 2\sin x - x^2 = 0$ with error precision 0.01

Scanned by CamScanner

Convergence of Secant Method

The secant formula for iteration is :

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i+1})}{f(x_{i-1}) - f(x_i)} \quad \text{--- (i)}$$

→ Let x_0 be the actual root of $f(x)$ & e_i the error estimate of x_i then

$$x_{i+1} = e_{i+1} + x_r$$

$$x_i = e_i + x_r$$

$$x_{i-1} = e_{i-1} + x_r$$

→ Substituting these in eq^n (i) and simplifying we get error equation as

$$e_{i+1} = \frac{e_{i-1} f(x_i) - e_i f(x_{i-1})}{f(x_i) - f(x_{i-1})} \quad \text{--- (ii)}$$

According to Mean Value Theorem, there exist at least one point, say $x = R_i$ in the interval x_i & x_r, such that

$$f'(R_i) = \frac{f(x_i) - f(x_r)}{x_i - x_r}$$

We know that $f(x_r) = 0$

$$x_i - x_r = e_i$$

and therefore, $f'(R_i) = \dfrac{f(x_i)}{e_i}$

$$\text{or, } f(x_i) = e_i f'(R_i)$$

Similarly, $f(x_{i-1}) = e_{i-1} f'(R_{i-1})$

Scanned by CamScanner

Substituting these in the numerator of eqn ② we get

$$e_{i+1} = e_i\, e_{i-1} \frac{f'(R_i) - f'(R_{i-1})}{f(x_i) - f(x_{i-1})}$$

That is, we can say

$$e_{i+1} \propto e_i\, e_{i-1} \quad \text{——} \quad ③$$

We know that, the order of convergence of an iteration process is p if

$$e_i \propto e_{i-1}^p$$

or, $e_{i+1} \propto e_i^p$

Substituting for e_{i+1} and e_i in equation ③ we get

$$e_i^p \propto e_{i-1}^p\, e_{i-1}$$

or, $e_i^p \propto e_{i-1}^{p+1}$

or, $e_i \propto e_{i-1}^{(p+1)/p}$

That is $p^2 - p - 1 = 0$

which has the solution

$$p = \frac{1 \pm \sqrt{5}}{2}$$

Since p is always positive we have

$$p = 1.618$$

It follows that the order of convergence of the secant method is 1.618 & convergence is reffered to as superlinear convergence.

Scanned by CamScanner

Algorithm :-

→ Input two initial guesses (say x_0 & x_1) & precision E

→ Evaluate $f(x_0)$ & $f(x_1)$

→ Estimate the new value of the Root as;

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

→ Find the absolute relative approximate error [E_a] as

$$E_a = \left| \frac{x_2 - x_1}{x_2} \right|$$

→ Compare the approximate relative approximate error [E_a] with prespecified relative error tolerance E.

If $E_a > E$

$x_0 = x_1$

$f(x_0) = f(x_1)$

$x_1 = x_2$

goto step 3

else

goto step 6

→ Stop

LAB Program :-

WAP (c) to implement the secant method to find the root of non-linear equation.

Scanned by CamScanner

Fixed Point Iteration Method

In fixed point iteration method we rearrange the function $f(x)=0$ such that x is on the left hand side of the equation. This can be expressed as below:

$$f(x)=0 \quad \text{—— (i)}$$

It is written as

$$x = g(x) \quad \text{—— (ii)}$$

Equation (ii) is called fixed point equation. This can be achieved by algebraic manipulation or by simply adding x to both sides of the original equation.

Eq^n (ii) provides a convenient way of providing the next value of x on the basis of current value of x.

If x_0 is the initial guess to the root, next approximation of the root can be calculated as

$$x_1 = g(x_0)$$

Similarly, further approximation can be computed as

$$x_2 = g(x_1)$$

In General

$$x_{n+1} = g(x_n)$$

This equation is called fixed point formula & gives sequence of x_0, x_1, x_2 ..... which is hoped to converge to a point x (fixed point) as shown in figure below

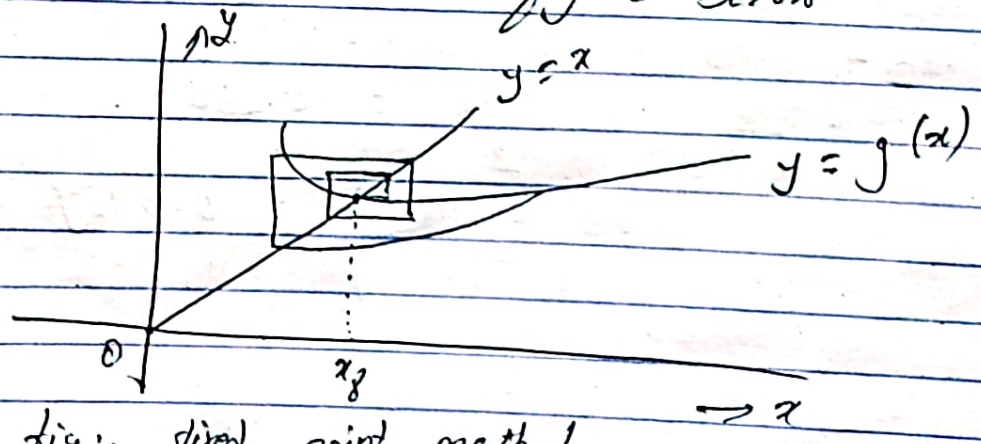


fig:- fixed point method.

Scanned by CamScanner

Eg:- Find the root of the equation $x^2-5=0$ by using fixed point method.

Eg: 2

Soln

Here, $f(x) = x^2-5$

$$x = \frac{5}{x}$$

Iteration 1

$x_0 = 1$

$x_1 = g(x_0) = \frac{5}{1} = 5$

Error $= \frac{5-1}{5} = 4/5 = 0.8$

Iteration 2

$x_1 = 5$

$x_2 = g(x_1) = \frac{5}{5} = 1$

Error $= \left|\frac{1-5}{1}\right| = 4$

Iteration 3

$x_2 = 1$

$x_3 = g(x_2) = 5/1 = 5$

Error $= \left|\frac{5-1}{5}\right| = 0.8$

Let us consider another form

$$x = \frac{5}{x}$$

$$2x = \frac{5}{x} + x$$

$$x = \frac{5}{2x} + \frac{x}{2}$$

Iteration 2

$x_0 = 1$

$x_1 = \frac{5}{2} + \frac{1}{2} = 3$

Error $= \left|\frac{3-1}{3}\right| = 0.66$

Iteration 2

$x_1 = 3$

$x_2 = \frac{5}{2\times3} + \frac{3}{2} = 2.33$

Error $= \left|\frac{2.33-3}{2.33}\right| = 0.285$

Iteration 3

$x_2 = 2.33$

$$x_3 = \frac{5}{2\times2.33} + \frac{2.33}{2} = 2.238$$

Error $= \left|\frac{2.238-2.33}{2.238}\right| = 0.0429$

Scanned by CamScanner

Find the root of the following equation by using fixed point method $x^2+x-2=0$.

Sol/

Here, $f(x) = x^2+x-2$

$x^2+x-2=0$

$x = 2-x^2$

$x = \frac{2-x}{x}$

Iteration 1

$x_0 = 4$

$x_1 = 2-4^2 = -14$

Error $= \left|\frac{-14-4}{-14}\right| = \cancel{0.875}$ $\cancel{1.285}$

Iteration 2

$x_1 = -14$

$x_2 = 2 -$

Iteration 1

$x_0 = 0$

$x_1 = 2-0^2 = 2$

Error $= \left|\frac{2-0}{2}\right| = 1$

Iteration 2

$x_1 = 2$

$x_2 = 2-2^2 = -2$

Error $= \left|\frac{-2-2}{-2}\right| = 2$

Iteration 3

$x_2 = -2$

$x_3 = 2-(-2)^2 = 2-4 = -2$

Error $= \left|\frac{-2-(-2)}{-2}\right| = 0$

$\therefore$ root $= -2$

Scanned by CamScanner

Convergence fixed point iteration method

→ Convergence of the iteration process depend on the nature of $g(x)$

→ We can theoritically prove this as follows:

→ The iteration formula is

$$x_{n+1} = g(x) \quad \text{—— (i)}$$

let x_f be the root of the equation then

$$x_f = g(x_f) \quad \text{—— (ii)}$$

Subtracting eqn (i) & eqn (ii) we get

$$x_f - x_{n+1} = g(x_f) - g(x_n) \quad \text{—— (iii)}$$

According to MVT, there is at least one point say $x = R$ in the interval between x_f & x_n such that

$$g'(R) = \frac{g(x_f) - g(x_n)}{x_f - x_n}$$

$$\therefore g(x_f) - g(x_n) = g'(R)(x_f - x_n)$$

Substituting this equation in (iii)

$$x_f - x_{n+1} = g'(R)(x_f - x_n) \quad \text{—— (iv)}$$

→ If e_i represents the error in the ith iteration, then equation (4) becomes

$$e_{n+1} = g'(R)\, e_n \quad \text{—— (iv)}$$

→ This gives the error will decrease with error iteration only if $g'(R) < 1$

→ The error is roughly proportional to the error in the previous step

→ Thus the fixed point iteration method is said to be linear convergent.

Scanned by CamScanner

Algorithm

1) Input initial guess (say x_0) & error estimate (say E)
2) Convert $f(x) = 0$ to the form $x = g(x)$
3) Estimate new value of the root x_1 as:

$$x_1 = g(x_0)$$

4) Find the absolute relative approximate error $|E_a|$ as:

$$|E_a| = \left| \frac{x_1 - x_0}{x_1} \right|$$

5) Compare absolute relative approximate error, E_a with the pre-specified relative error tolerance E.

If $|E_a| > E$

$x_0 = x_1$

goto step 3

else

goto step 6

STOP

Program

C program to implement fixed point iteration method to find the root of the non-linear equation.

Scanned by CamScanner

Synthetic Division

→ It is a method of performing polynomial long division with less writing & fewer calculations

→ Let $P(x)$ be a polynomial of degree n. If we divide $P(x)$ by $(x-a)$, we get another polynomial $q(x)$, which is quotient of degree $n-1$.

→ Assume

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + x_0$$

And

$$q(x) = b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \dots + b_0$$

So,

$$P(x) = q(x)(x-a)$$

$$a_n a^n + a_{n-1} x^{n-1} + \dots + a_0 = (b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \dots + b_0)(x-a).$$

Comparing the coefficients on both side of the equation

$$b_{n-1} = a_n$$

$$b_{n-2} = a_{n-1} + a b_{n-1}$$

$$\vdots$$

$$b_0 = a_1 + a b_1$$

Thus

$$b_n = 0 \ \&$$

$$b_{i-1} = a_i + a \times b_i \qquad i = 1 \dots n$$

Scanned by CamScanner

:- Solve the following problem by using synthetic division method $(x^4 - 3x + 5) \div (x - 4)$

Sol:

Step 1 Set the synthetic division

$$x - 4 \overline{)\, x^4 - 3x + 5}$$

Then it becomes

4	1	0	0	-3	5
		4	16	64	244
	1	4	16	61	$\boxed{249}$

Remainder = 249

Scanned by CamScanner

Remainder Theorem

→ Remainder theorem states that if the polynomial $f(x)$ is divided by $(x-c)$, then the remainder is $f(c)$.

→ This means that we can apply synthetic division & the last number is the remainder. This remainder is the functional value of $f(x)$ at $x=c$

→ Thus the remainder theorem is useful for evaluating polynomial at a ~~guess~~ given value of x.

~~Sol~~ Eg:- Given $f(x) = 3x^3 + x^2 + x - 5$ use the remainder theorem to find $f(-2)$

Sol/

-2	3	1	1	-5
	↓	-6	10	-22
	3	-5	11	-27

Remainder = -27

So, $f(-2) = -27$

Scanned by CamScanner

Finding Multiple Roots by using Newton-Raphson's Method.

All real roots of polynomials can be ~~formed~~ found by repeatedly applying Newton Raphson method & synthetic division

Actually synthetic division is used to obtain polynomial of degree n-1 from polynomial of degree n. Roots of polynomial can be found by using Newton-Raphon's formula & initial guess

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

Suppose x_r be one of the root of polynomial of degree n. Now deflate the polynomial by dividing it by $x - x_r$ to get another polynomial of degree n-1 & use x_r as an initial guess for finding next root.

Continue this process until polynomial of degree one is achieved as quotient. The polynomial will be of the form $a_1 x + a_0 = 0$

Now, the final root can be calculated as

$$x = \frac{a_0}{a_1}$$

Scanned by CamScanner

Eg:- Find the roots of eqn $x^4 - 2x^3 - 13x^2 + 38x - 24 = 0$ using Newton - Raphsons method.

Soln

Assume, Initial guess $(x_0) = 0.5$ & Error limit $(E) = 0.01$

Step 1: Use the Newton - Raphson method to get root

Thus the 4th root of the equation is 1.00

Step 2 :- Now use synthetic division to deflate the polynomial by $(x-1)$ which gives the polynomial

$$x^3 - x^2 - 14x + 24 = 0$$

Again use the N-RM with

Initial guess $(x_0) = 1$ & Error limit $(E) = 0.01$

Thus the 3rd root of the equation $= 2.00$

Step 3 :- Again use the synthetic divison to deflate the polynomial by $(x-2)$ which gives the polynomial $x^2 - x + 12 = 0$

From the Newton - Raphson method with

Initial Guess $(x_0) = 2.00$ & Error limit $(E) = 0.01$

2nd root of equation $= 3.00$

Step 4 :- Again use the synthetic division to deflate the polynomial by $(x-3)$ which gives

$$x + 4 = 0$$

Now the 1st root of the polynomial is calculated as

$$\text{Root} = -\frac{a_0}{a_1} = \frac{-4}{1} = -4$$

Thus, 1st root is -4

Scanned by CamScanner

Algorithm

Input the degree & coefficient of polynomial
Input initial guess x_0 & error limit ϵ
While $n > 1$
- Find the root using Newton - Raphson algorithm say n^{th} root is x_r
→ Divide the polynomial by $x - x_r$ to get the polynomial of degree $n-1$
- set $x_0 = x_1$
- $n = n-1$

End while
1st Root $= -\frac{a_0}{a_1}$
Stop

Scanned by CamScanner

Horners Method for Polynomial evaluation

→ It is a way to optimize the task of evaluating a polynomial evaluation

→ The method splits the polynomial into its individual terms & solve them individually.

→ This method separates the lowest degree term in the polynomial from rest.

→ Thus it represent any polynomial in the form $k * x + c$ where k is the group of all terms with degree higher than one & c is the term with degree zero.

→ Consider a general polynomial equation.

$$P(x) = \sum_{i=0} a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

where $a_0, a_1, a_2 \dots, a_n$ are the real numbers & we wish to evaluate the polynomial at point say x_0

→ To accomplish this, we need to define the given polynomial in terms of nested multiplication.

$$P(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

$$P(x) = a_0 + x(a_1 + a_2 x + \dots + a_n x^{n-1})$$

$$P(x) = a_0 + x(a_1 + x(a_2 + \dots + a_n x^{n-2}))$$

$$\vdots \qquad \vdots$$

$$P(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + x(a_n))))$$

→ Now we can define new sequence of constants as

$$b_n = a_n$$

$$b_{n-1} = a_{n-1} + x b_n$$

$$\vdots$$

$$b_0 = a_0 + x b_1$$

→ Then b_0 is the value of $P(x_0)$

Scanned by CamScanner

Date
Page

Evaluate the following polynomial at $x = 2$

$$P(x) = 3x^3 - 4x^2 + 5x - 6$$

Sol/

Here, $a_3 = 3$, $a_2 = -4$, $a_1 = 5$ $a_0 = 6$ & $x_0 = 2$

Now

Now constant are calculated as follows

$$b_3 = a_3 = 3$$
$$b_2 = a_2 + x_0 * b_3 = (-4) + 2 * (3) = 2$$
$$b_1 = a_1 + x_0 * b_2 = 5 + 2 * 2 = 9$$
$$b_0 = a_0 + x_0 * b_1 = 6 + 2 * 9 = 24$$

Thus

$$P(2) = 12$$

Algorithm

Enter the coefficients of the polynomial

Enter the value at when the polynomial need to to evaluate; say x_0

Enter the degree of polynomial, say n & set $b_n = a_n$

While $n > 1$

$b_{n-1} = a_{n-1} + x_0 * b_n$

End while

Display the value of b_0, which is the value of $P(x_0)$

Stop

Program :- Write a C program to implement Horner's method to evaluate the polynomial.

Scanned by CamScanner

Lagrange Interpolation Polynomial

1)

2)

3.)

4.)

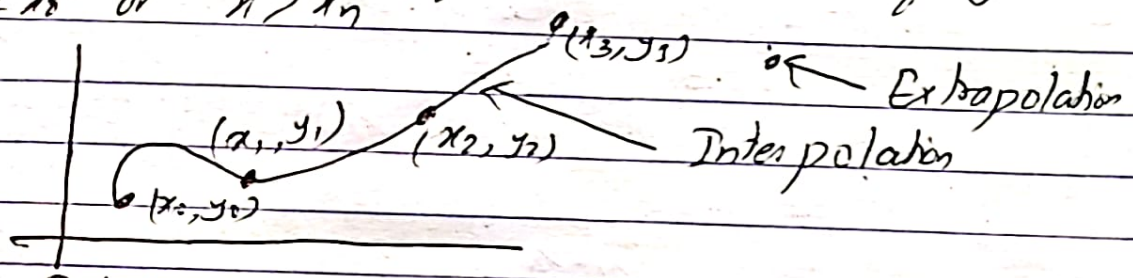
Scanned by CamScanner

Unit - 2 Interpolation & Regression

Interpolation & Extrapolation

- Given $n+1$ data points $(x_0, y_0), (x_1, y_1) \ldots (x_n, y_n)$, then interpolation is the process of finding an equation $y = f(x)$ that passes through above $(n+1)$ data points & using this equation to find the value of y at x, where $x_0 < x < x_n$

- Extrapolation is the process of finding an equation $y = f(x)$ that passes through above $(n+1)$ data points & using this equation to find the value of y at x where $x < x_0$ or $x > x_n$



Forms of Polynomial

1) Power form

$$P(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

2) Shifted Power form :-

$$P(x) = a_0 + a_1 (x-c) + a_2 (x-c)^2 + \ldots + a_n (x-c)^n$$

3) Newton form (Generalized Shifted Power form)

$$P(x) = a_0 + a_1 (x-c_1) + a_2 (x-c_1)(x-c_2) + \ldots + a_n (x-c_1)(x-c_2)\ldots(x-c_n)$$

4) 4th form

$$P_2(x) = b_0 (x-x_1)(x-x_2) + b_1 (x-x_0)(x-x_2) + b_2 (x-x_0)(x-x_1)$$

In general $$P_n(x) = \sum_{i=0}^{n} b_i \prod_{j=0, j \neq i}^{n} (x - x_j)$$

Scanned by CamScanner

Lagrange Interpolation Polynomial

→ Let us consider second order polynomial,

$$P_2(x) = b_0(x-x_1)(x-x_2) + b_1(x-x_0)(x-x_2) + b_2(x-x_0)(x-x_1)$$

→ Let (x_0, f_0), (x_1, f_1), (x_2, f_2) are interpolating points then

$$P_2(x_0) = b_0(x_0-x_1)(x_0-x_2) = f_0$$
$$P_2(x_1) = b_1(x_1-x_0)(x_1-x_2) = f_1$$
$$P_2(x_2) = b_2(x_2-x_0)(x_2-x_1) = f_2$$

→ Evaluate the value of $b_0, b_1, \& b_2$

$$b_0 = \frac{f_0}{(x_0-x_1)(x_0-x_2)}$$

$$b_1 = \frac{f_1}{(x_1-x_0)(x_1-x_2)}$$

$$b_2 = \frac{f_2}{(x_2-x_0)(x_2-x_1)}$$

Substituting the value of $b_0, b_1 \& b_2$ in equation ①

$$P_2(x) = f_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

This above equation can be written as

$$P_2(x) = f_0 L_0(x) + f_1 L_1(x) + f_2 L_2(x)$$

where,

$$L_i(x) = \prod_{\substack{j=0 \\ j\neq i}} \frac{(x-x_j)}{(x_i-x_j)}$$

In General

$$P_n(x) = \sum_{i=0}^{n} f_i L_i(x) \text{ where } L_i(x) = \prod_{j=0}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

Scanned by CamScanner

Eqn (ii) is called Lagrange polynomial and eqn (iii) is called Lagrange basis polynomial.

• Table below given the square roots of integers

x	1	2	3	4	5
	1	1.4142	1.7321	2	2.2361

Determine the quare root of 2.5

Soln

Let us take the second order Langrange polynomial

$$P_2(x) = f_0 l_0(x) + f_1 l_1(x) + f_2 l_2(x)$$

where

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{2} \frac{(x - x_j)}{(x_j - x_i)}$$

Since, we are using 2nd order polynomial, so

$x_0 = 2$, $x_1 = 3$ & $x_2 = 4$

then $f_0 = 1.4142$, $f_1 = 1.7321$, $f_2 = 2$ 2

Now, evaluating Langrange Basis Polynomial

$$l_0(2.5) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(2.5-3)(2.5-4)}{(2-3)(2-4)} = 0.3750$$

$$l_1(2.5) = \frac{(2.5-2)(2.5-4)}{(3-2)(3-4)} = 0.7500$$

$$l_2(2.5) = \frac{(2.5-2)(2.5-3)}{(4-2)(4-3)} = -0.125$$

Again

$$P_2(2.5) = (1.4142)(0.3750) + (1.7321)(0.7500) + (2)(-0.125)$$

$$= 0.5303 + 1.2991 - 0.250$$

$$= 1.5794$$

Hence the square root of 2.5 is 1.5794

For 3rd order

$$P_3(x) = f_0 l_0(x) + f_1 l_1(x) + f_2 l_2(x) + f_3 l_3(x)$$

Where $l_i(x) = \prod_{j=0, j \neq i}^{3} \frac{(x - x_j)}{(x_i - x_j)}$

Scanned by CamScanner

Find the cubic langrange Interpolation Polynomial fitting the polynomial points $y(1) = -3$, $y(3) = 0$, $y(4) = 3$, $y(6) = 132$ & find $y(5)$

Sol^n

let us take 3rd order polynomial

$$P_3(x) = f_0 l_0(x) + f_1 l_1(x) + f_2 l_2(x) + f_3 l_3(x)$$

where

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{3} \frac{(x - x_j)}{(x_i - x_j)}$$

From Given question

$x_0 = 1$, $x_1 = 3$, $x_2 = 4$, $x_3 = 6$

And $f_0 = -3$, $f_1 = 0$, $f_2 = 3$, $f_3 = 132$

Now

Evaluating langrange Basis Polynomial

$$l_0(5) = \frac{(5-3)(5-4)(5-6)}{(1-3)(1-4)(1-6)} = \frac{2 \times -1}{-2 \times -3 \times -5} = \frac{1}{15}$$

$$l_1(5) = \frac{(5-1)(5-4)(5-6)}{(3-1)(3-4)(3-6)} = \frac{4 \times 1 \times -1}{2 \times -1 \times -3} = \frac{-2}{3}$$

$$l_2(5) = \frac{(5-1)(5-3)(5-6)}{(4-1)(4-3)(4-6)} = \frac{4 \times 2 \times -1}{3 \times 1 \times -2} = \frac{4}{3}$$

$$l_3(5) = \frac{(5-1)(5-3)(5-4)}{(6-1)(6-3)(6-4)} = \frac{4 \times 2 \times 1}{5 \times 3 \times 2} = \frac{4}{15}$$

$$\therefore P_3(5) = -3 \times \frac{1}{15} + 0 \times \frac{-2}{3} + 3 \times \frac{4}{3} + 132 \times \frac{4}{15}$$

$$= \frac{-1}{5} + 4 + \frac{44 \times 4}{5}$$

$$= \frac{-1 + 20 + 44 \times 4}{5} = \frac{195}{5} = 39$$

Scanned by CamScanner

Algorithm

Read the number of points, say n
Read the point at which we need to compute the value, say x
Read interpolating points
Calculate L_i values as :

for $i=0$ to $n-1$

for $j=0$ to $n-1$

if $(i \neq j)$ $L_i = L_i \times ((x - x_j) / (x_i - x_j))$

Compute the interpolation value as

for $i=0$ to $n-1$

$v = v + f_i \times L_i$

Display the value of v
Stop

gram

Write a C program to implement the Langrange interpolation polynomial.

Scanned by CamScanner

Newtons Divided Difference Interpolation

Let us consider a polynomial of degree n,

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2) + \dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}) \quad \text{--(i)}$$

→ To construct interpolation polynomial, we need to find the value of coefficients $a_0, a_1, \dots a_n$.

→ Let us consider $(x_0, f_0), (x_1, f_1), \dots (x_n, f_n)$ are interpolating points then,

$$P_n(x_0) = a_0 = f_0 \quad \therefore a_0 = f_0$$

$$P_n(x_1) = a_0 + a_1(x_1 - x_0) = f_1$$

or, $f_1 = a_0 + a_1(x_1 - x_0)$

or, $a_1 = \dfrac{f_1 - a_0}{x_1 - x_0}$

$$P_n(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$$

$$\therefore a_2 = \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}$$

Note that $a_0, a_1, \& a_2$ are first, second & third divided difference respectively, let denote these values as

$$a_0 = f[x_0] = f_0$$

$$a_1 = f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0}$$

$$a_2 = f[x_0, x_1, x_2] = \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}$$

x_1

x_0

x_1

x_2

x_3

x_4

Scanned by CamScanner

This leads us to write the general form of the polynomial as :

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, x_2 \dots x_n](x - x_0)(x - x_1) \dots (x - x_{n-1})$$

$$P_n(x) = f[x_0] + \sum_{i=1}^{n} f[x_0, x_1, \dots x_i] \prod_{j=0}^{i-1} (x - x_j)$$

Divided difference Table:-

The coefficient of Newtons interpolation polynomial are calculated from divided of interpolating points recursively. So we can create table to make computation efficient. The divided difference table for polynomial of degree 4 is

$f(x_i)$	1st diff	2nd diff	3rd diff	4th diff
$f(x_0)$				
	$f[x_0, x_1]$			
$f(x_1)$		$f[x_0, x_1, x_2]$		
	$f[x_1, x_2]$		$f[x_0, x_1, x_2, x_3]$	
$f(x_2)$		$f[x_1, x_2, x_3]$		$f[x_0, x_1, x_2, x_3, x_4]$
	$f[x_2, x_3]$		$f[x_1, x_2, x_3, x_4]$	
$f(x_3)$		$f[x_2, x_3, x_4]$		
	$f[x_3, x_4]$			
$f(x_4)$				

Scanned by CamScanner

Eg:- Given the following data points, obtain the dividend difference table. Use the table to estimate the value of $f(1.5)$

i	0	1	2	3	4
x_i	1	2	3	4	5
$f(x_i)$	0	7	26	63	124

Sol^n/

The divided difference table for given data points is

x_i	$f(x_i)$	1st diff	2nd diff	3rd diff	4th diff
1	0				
		7			
2	7		6		
		19		1	
3	26		9		0
		37		1	
4	63		12		
		61			
5	124				

$x = 1.5$

$P_0(x) = P_0(1.5) = a_0 = 0$

$P_1(x) = P_1(1.5) = a_0 + a_1(1.5-1) = 0 + 7 \times 0.5 = 3.5$

$P_2(x) = P_2(1.5) = a_0 + a_1(1.5-1) + a_2(1.5-1)(1.5-2)$

$= 3.5 + 6(0.5)(-0.5) = 2$

$P_3(x) = P_3(1.5) = P_2(1.5) + a_3(1.5-1)(1.5-2)(1.5-3)$

$= 2 + 1(0.5)(-0.5)(-2.5)$

$= 2.0375$

Eg:-

x_i
2
7
9
10

Scanned by CamScanner

Given the following ..... find $f(8)$

x	2	7	9	10
$f(x)$	168	120	72	63

Sol^n

The dividend diff table for given data points;

$f(x_i)$	1st diff	2nd diff	3rd diff
168			
	-9.6		
120		-2.05	
	-24		-1.118
72		-11	
	-9		
63			

$P_0(8) = a_0 = 168$

$P_1(8) = a_0 + a_1(8-2) = 168 + -9.6(6) = 110.4$

$P_2(8) = P_1(8) + a_2(8-2)(8-7) = 110.4 + -2.05(6)(1)$
$= 98.1$

$P_3(8) = P_2(8) + a_3(8-2)(8-7)(8-9)$
$= 98.1 + -1.118 \times 6 \times -1$
$= 104.808$

Scanned by CamScanner

50

1	0			
		0.3010		
2	0.3010		−0.002405	
		0.1761		0.0123
3	0.4771		−0.02555	
		0.125		
4	0.6021			

Scanned by CamScanner

Algorithm of Newton's dividend difference Interpolation:

Read the number of interpolating points, say n
Read a value where we need to calculate interpolation, say x
Read the given points
Assign the value of $fx[x_i]$ to $dd[x_i]$
for $i=0$ to $n-1$
$dd[i] = fx[i]$
Compute the dividend difference
for $i=0$ to $n-1$
for $j=n-1$ to $i+1$

$$dd[j] = \frac{dd[j] - dd[j-1]}{x[j] - x[j-i-1]}$$

set $v=0$ & $p=1$
Calculate the interpolation value
for $i=0$ to $n-1$
{
for $j=0$ to $i-1$
{ $p = p * (x - x[j])$
}
$v = v + dd[i] * p$
Reset $P=1$
}
Display the value of v
Stop

$x[\,] = [2 \mid 4 \mid 6 \mid 8]$

$fx[\,] = [4 \mid 16 \mid 36 \mid 24]$

$dd[\,] = [4 \mid 16 \mid 36 \mid 24]$

$i=0$, $j = 3$ to 1

$dd[\,] = [4 \mid 6 \mid 10 \mid 14]$

$i=1$ $j=3$ to 2

$dd[\,] = [4 \mid 6 \mid 1 \mid 1]$

$i=2$ $j=3$ to 3

$dd[\,] = [4 \mid 6 \mid 1 \mid 0]$

Scanned by CamScanner

Newton's Forward and Backward Interpolation

→ Newton's divided difference formula can be expressed in simplified form when $x_1, x_2, \ldots, x_n$ are arranged consecutively in equal spacing.

→ Newton's forward and backward formula can be used in equal intervals.

→ Newton's forward formula is used when we need to find the value at the beginning and backward formula is used ~~at the~~ when we need to find the value at the end.

Newton's forward Interpolation formula :-

We know that the Newton's divided difference interpolation formula :-

$$P_n(x) = f[x_0] + f[x_0\, x_1](x-x_0) + f[x_0, x_1, x_2](x-x_0)(x-x_1) + \ldots + f[x_0, x_1 \ldots x_n](x-x_0)\ldots(x-x_n)$$

Let us define $h = x_i - x_{i-1}$ ~~so the Newtons~~, for all $i = 0, 1 \ldots n$

Suppose,

$$x = x_0 + sh$$

Since $x_k = x_0 + kh$

Then, $x - x_k = (s-k)h$

This gives,

$$x - x_0 = sh$$

$$x - x_1 = (s-1)h$$

$$\vdots$$

$$x - x_{n-1} = (s-n+1)h$$

Scanned by CamScanner

Now, the Newtons divided difference interpolation formula becomes,

$$P_n(x) = f[x_0] + f[x_0, x_1]sh + f[x_0, x_1, x_2]\, s(s-1)h^2 + \dots + f[x_0, x_1, \dots x_n]\, s(s-1)(s-2)\dots(s-n+1)h^n$$

$$= \sum_{i=0}^{n} f[x_0, x_1, x_2, \dots x_i]\, s(s-1)(s-2)\dots(s-i+1)h^i$$

Using the binomial notation,

$$\binom{s}{k} = \frac{s(s-1)(s-2)(s-3)\dots(s-k+1)}{k!}$$

Then, the above formula becomes

$$P_n(x) = \sum_{i=0}^{n} \binom{s}{i} i!\, h^i f[x_0, x_1, \dots x_i]$$

Again, define some notations

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{1}{h}\Delta f(x_0)$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_1, x_0]}{x_2 - x_0}$$

$$= \frac{1}{2h}\left[\frac{1}{h}\Delta f(x_1) - \frac{1}{h}\Delta f(x_0)\right]$$

$$= \frac{1}{2h^2}\Delta^2 f(x_0)$$

$$f[x_0, x_1, \dots x_n] = \frac{f[x_1, x_2 \dots x_n] - f[x_0, x_1, \dots x_{n-1}]}{x_n - x_0} = \frac{1}{nh^n}\Delta^n f(x_0)$$

Now, the above formula becomes

$$P_n(x) = \sum_{i=0}^{n} \binom{s}{i} i!\, h^i \frac{1}{ih^i}\Delta^i f(x_0)$$

$$P_n(x) = P_n(x_0 + sh) = \sum_{i=0}^{n} \binom{s}{i} \frac{i!}{i}\Delta^i f(x_0)$$

Scanned by CamScanner

Forward difference table :-

We are create the forward difference table to calculate the difference value as below :-

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$	$\Delta^n f_i$
x_0	f_0					
		Δf_0				
x_1	f_1		$\Delta^2 f_0$			
		Δf_1		$\Delta^3 f_0$		
x_2	f_2		$\Delta^2 f_1$		$\Delta^4 f_0$	
		Δf_2		$\Delta^3 f_1$		$\Delta^5 f_0$
x_3	f_3		$\Delta^2 f_2$		$\Delta^4 f_1$	
		Δf_3		$\Delta^3 f_2$		
x_4	f_4		$\Delta^2 f_3$			
		Δf_4				
x_5	f_5					

N.E # Estimate the value of θ at $\theta = 25°$ using Newton Gregory forward difference formula with the help of following table

θ	10	20	30	40	50
$\sin\theta$	0.1736	0.3420	0.5000	0.6428	0.7660

Sol/ Here, $h = x_{i+1} - x_i = 10$

$$x = x_0 + sh \Rightarrow s = \frac{x - x_0}{h} = \frac{25-10}{10} = 1.5$$

Scanned by CamScanner

f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0.1736				
	0.1684			
0.3420		-0.0104		
	0.1580		0.0048	
0.5000		-0.0152		-0.0004
	0.1428		0.0044	
0.6428		-0.0196		
	0.1232			
0.7660				

We know that;

$$P_n(x) = \sum_{i=0}^{n} \binom{s}{i} \Delta^i f(x_0)$$

where $\binom{s}{i} = \dfrac{s(s-1)(s-2)\cdots(s-i+1)}{i!}$

$$P_4(x) = \binom{s}{0}\Delta^0 f(x_0) + \binom{s}{1}\Delta f(x_0) + \binom{s}{2}\Delta^2 f(x_0) + \binom{s}{3}\Delta^3 f(x_0) + \binom{s}{4}\Delta^4 f(x_0)$$

$$= f(x_0) + s\,\Delta f(x_0) + \frac{s(s-1)}{2}\Delta^2 f(x_0) + \frac{s(s-1)(s-2)}{3!}\Delta^3 f(x_0) + \frac{s(s-1)(s-2)(s-3)}{4!}\Delta^4 f(x_0)$$

$$P_4(25) = 0.1736 + (1.5)(0.1684) + \frac{(1.5)(1.5-1)}{2}(-0.0104) + \frac{(1.5)(1.5-1)(1.5-2)}{-6} \times (0.0048) + \frac{(1.5)(1.5-1)(1.5-2)(1.5-3)}{24}(-0.0004)$$

$$= 0.4220$$

Thus, $\sin\theta = \sin 25 = 0.4220$

Scanned by CamScanner

x	1	2	3	4
$f(x)$	0.5	0.3333	0.25	0.20

$(f = 2.5)$

Sol/

Here, $h = x_{i+1} - x_i = 1$

$x = x_0 + sh \Rightarrow s = \frac{x - x_0}{h} = \frac{2.5-1}{1} = 1.5$

x_i	f_i	Δf_i	$\Delta^2 f_i$
1	0.5		
2	0.3333		
3	0.25		
4	0.20		

1.)
2.)
3.)
4)
5.)
6.)
7.)
8)

Scanned by CamScanner

Algorithm

Read the number of point, say n
Read the interpolating point at which we need to calculate the value, say x
Read the given points
Calculate the first forward difference
for i = 0 to n-1
fd[i] = fx[i]
Computing other differences as
for i = 0 to n-1
for j = n-1 to i+1
fd[j] = fx[j] - fx[j-1]
Set xv = fx[0], p = 1
Calculate the interpolating value as
for i = 1 to n-1
p = 1
for k = 1 to i-1
{ p = p * (s-k+1)
}
$xv = xv + \frac{p * fd[i]}{i!}$

Display the interpolating value xv.

Scanned by CamScanner

Newton's Backward Interpolation Polynomial

By rearranging interpolating values as $x_n, x_{n-1}, x_{n-2} \dots x_0$, the Newton's divided difference formula becomes

$$P_n(x) = f[x_n] + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x_{n-2}] (x - x_n)(x - x_{n-1}) + \dots + f[x_n, x_{n-1}, \dots x_0](x - x_n) \dots (x - x_1)$$

Let us define, $h = x_{i+1} - x_i$ for all $i = 0, \dots n$

Suppose,

$$x = x_n + sh$$

Since $x_k = x_n + (k-n)h$

$$\Rightarrow x - x_k = (s - k - n)h$$

This gives

$$x - x_1 = (s - 1 + n)h$$

$$x - x_2 = (s - 2 + n)h$$

$$\vdots$$

$$x - x_{n-1} = (s+1)h$$

$$x - x_n = sh$$

Then the equation (1) can be written as

$$P_n(x) = f[x_n] + f[x_n, x_{n-1}]sh + f[x_n, x_{n-1}, x_{n-2}]s(s+1)h^2 + \dots + f[x_n, x_{n-1}, x_{n-2}, \dots x_0]s(s+1)(s+2)\dots(s+n-1)h^n \quad \text{—— (2)}$$

Now, converting equation (2) by using the backward difference notations

$$f[x_n, x_{n-1}] = \frac{f[x_n] - f[x_{n-1}]}{x_n - x_{n-1}} = \frac{1}{h}[f[x_n] - f[x_{n-1}]]$$

$$= \frac{1}{h} \nabla f(x_n)$$

$$f[x_n, x_{n-1}, x_{n-2}] = \frac{f[x_n, x_{n-1}] - f[x_{n-1}, x_{n-2}]}{x_n - x_{n-2}}$$

$$= \frac{1}{2h}\left[\frac{1}{h}\nabla f(x_n) + \frac{1}{h}\nabla f(x_{n-1})\right] = \frac{1}{2!}\nabla^2 f(x_n)$$

Scanned by CamScanner

In General,

$$f[x_n, x_{n-1}, x_{n-2}, \dots x_k] = \frac{1}{k!\,h^k} \nabla^k f(x_n)$$

Now, the equation ② becomes

$$P_n(x) = f[x_n] + \nabla f(x_n) s + \frac{1}{2!} \nabla^2 f(x_n) s(s+1) + \dots + \frac{\nabla^n f(x_n)}{n!} s(s+1) \dots (s+n-1)$$

This is the required Newton - Gregory backward difference interpolation formula.

Newtons Backward difference Table :-

We can compute the Newton's backward interpolation by using backward difference table

x_i	f_i	$\nabla f(x_i)$	$\nabla^2 f(x_i)$	$\nabla^3 f(x_i)$	$\nabla^4 f(x_i)$
x_0	f_0				
		$\nabla f(x_1)$			
x_1	f_1		$\Delta^2 f(x_2)$		
		$\nabla f(x_2)$		$\nabla^3 f(x_3)$	
x_2	f_2		$\nabla^2 f(x_3)$		$\nabla^4 (x_4)$
		$\Delta f(x_3)$		$\nabla^3 (x_4)$	
x_3	f_3		$\nabla^2 f(x_4)$		
		$\nabla f(x_4)$			
x_4	f_4				

Scanned by CamScanner

N.E # Estimate the value of θ at $\theta = 25$ using Newton-Gregory backward difference formula with the help of following table

θ	10	20	30	40	50
sine	0.1736	0.3420	0.5000	0.6428	0.7660

Here, $h = 20 - 10 = 10$, $s = \frac{(x - x_n)}{h} = \frac{25 - 50}{10} = -2.5$

The backward difference table for given data is

θ	sinθ	$\nabla f(x_i)$	$\nabla^2 f(x_i)$	$\nabla^3 f(x_i)$	$\nabla^4 f(x_i)$
10	0.1796				
		0.1684			
20	0.3420		-0.0104		
		0.1580		0.0048	
30	0.5000		-0.0152		-0.0004
		0.1428		0.0044	
40	0.6428		-0.0196		
		0.1238			
50	0.7660				

We know that

$$P_n(x) = P_n(x_n + sh) = f[x_n] + \nabla f(x_n) s + \frac{\nabla^2 f(x_n)}{2!} s(s+1) + \dots + \frac{\nabla^n f(x_n)}{n!} s(s+1)(s+2)\dots(s+n-1)$$

$$P_4(25) = f[x_4] + \nabla f(x_4) s + \frac{\nabla^2 f(x_4)}{2} s(s+1) + \frac{\nabla^3 f(x_4)}{6} s(s+1)(s+2) + \frac{\nabla^4 f(x_4)}{24} s(s+1)(s+2)(s+3)$$

Scanned by CamScanner

$= 0.7660 + (0.1232)(-2.5) + \frac{(-0.0196)}{2}(-2.5)(-2.5+1)$

$--- + \frac{0.00044}{6}(-2.5)(-2.5+1)(-2.5+2) +$

$\frac{(-0.0004)}{24}(-2.5)(-2.5+1)(-2.5+2)(-2.5+3)$

$= 0.4220$

Thus $\sin 25 = 0.4220$

#

x	1.2	1.3	1.4	1.5
$f(x)$	1.063	1.091	1.119	1.145

$h = 1.3 - 1.2 = 0.1 \quad s = \frac{(x - x_n)}{h} = \frac{1.35 - 1.5}{0.1} = -1.5$

The backward difference table for given data is

x	$f(x)$			
1.2	1.063			
		0.028		
1.3	1.091		0	
		0.028		-0.002
1.4	1.119		-0.002	
		0.026		
1.5	1.145			

$P_n(x) = 1.145 + 0.026 \times (-1.5) + (-0.002)(-1.5)(-1.5+1)$
$+ (-0.002)(-1.5)(-1.5+1)(-1.5+2)$
$= 1.145 - 0.036 - 0.0015 - 0.00075$
$= 1.10675$

Scanned by CamScanner

Newtons Backward difference Interpolation :-

Algorithm :-

1.) Read the number of points, say n
2.) Read the value at which we need to calculate interpolated value say x_p
3.) Read the n data points, say $x[i]$ & $fx[i]$
4.) Compute $h = x[i] - x[0]$, & $s = (x_p - x[n-1])/h$
5.) Compute first backward difference as
    for $i = 0$ to $n-1$
        $bd[i] = fx[i]$
6.) Compute 2nd to n^{th} backward difference as
    for $i = n-1$ to 1
        for $j = 0$ to $i-1$
            $bd[j] = bd[j+1] - bd[j]$
7.) set $v = bd[n-1]$, set $p = 1$
8) Compute interpolation value,
    for $i = 1$ to $n-1$
        for $j = 1$ to i
            $p = p * (s+j-1)$
        End for
        $v = v + (bd[n-i-1] * p)/i!$
        Reset $p = 1$
    End for
9). Display v

Scanned by CamScanner

Date
Page

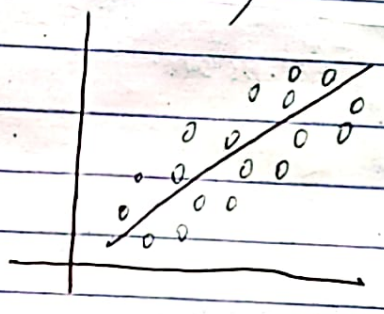
Scanned by CamScanner

Date
Page

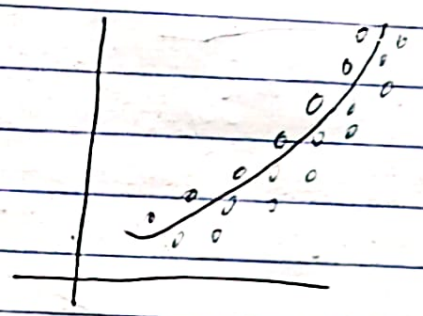
Scanned by CamScanner

Regression Analysis

It is a form of predictive modeling technique which investigates the relationship between a dependent variable and independent variable. Dependent variables are also called target variable and independent variables are also called predictors.



Linear regression



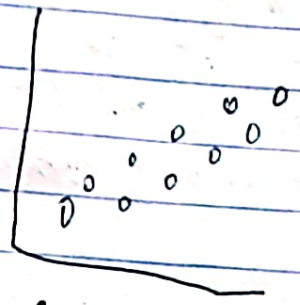
Non-linear regression

Interpolation Vs Regression

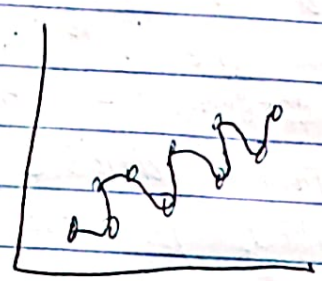
In interpolation we are given some data points and we are supposed to find a curve which fits the input / output relationship perfectly. In case of interpolation we don't have to worry about variance of the fitted curve. That means given the set of n data points (x_i, y_i) we look for the function $f(x)$ that satisfies the relation $y_i = f(x_i)$ for all given data points.

When we do regression we look for a function that minimizes some cost, usually sum of squares of errors. We don't require the function to have the exact values at given points, we just want good approximation. In general, we search for a function $y = f(x)$. That might not satisfied the relation $y_i = f(x_i)$ for any given data point but the

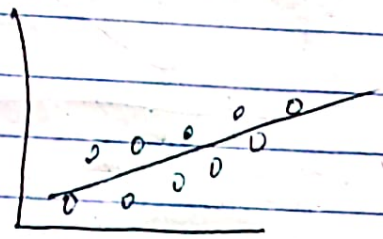
cost function $\sum (f(x_i) - y_i)^2$ will be smallest possible of all the functions of given form.



Given data points



Interpolation



Regression

Parameter Estimation

is the process of estimating values of parameters of the model based on the observed pair of values and minimizing certain objective function. Some methods of parameter estimations are as follows:

i) Minimize the sum of errors :-
i.e. minimize $\sum e_i = \sum y_i - f(x_i)$, $i = 1, 2, 3, \ldots, n$

ii) Minimize the sum of absolute value of errors
i.e. minimize $\sum |e_i| = \sum |y_i - f(x_i)|$, $i = 1, 2, 3, \ldots n$

iii) Minimize the sum of squares of errors (Least Square Method)
i.e minimize $\sum e_i^2 = \sum (y_i - f(x_i))^2$, $i = 1, 2, 3 \ldots n$

Scanned by CamScanner

least Square Method

- This is the most popular method of parameter estimation for coefficient of regression method.
- → Ideally if all the residual (errors) e_i are zero, one may have found an equation in which all the points lie on a model. In the least square method, estimation of the constants of models are chosen such that minimization of the sum of the squared residual is achieved.
- → In this method we need to minimize following objective function or error function.

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - f(x_i))^2 \quad , i = 1, 2, \ldots, n$$

Linear Regression

- → Filling a straight line is the ~~smallest~~ simplest approach of regression analysis which is called linear regression.
- → A straight line can be represented by using the mathematical equation $y = f(x) = a + bx$, where a and b are regression coefficient to be determined
- → let sum of squares of individual errors can be expressed as

$$E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

- → We should choose regression coefficient a and b such that E is minimum. Necessary condition for E to be minimum is :

Scanned by CamScanner

$$\frac{\partial E}{\partial a} = 0 \quad \& \quad \frac{\partial E}{\partial b} = 0$$

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{n} 2(y_i - a - bx_i)(-1) = 0 \quad \&$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{n} 2(y_i - a - bx_i)(-x_i) = 0$$

Simplifying above equation, we get

$$-\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} a + \sum_{i=1}^{n} bx_i = 0 \quad \&$$

$$-\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} ax_i + \sum_{i=1}^{n} bx_i^2 = 0$$

Since,

$$\sum_{i=1}^{n} a = a + a + \dots + a = na$$

Then above equation becomes

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \quad \text{——— (i)}$$

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \quad \text{——— (ii)}$$

Solving the above equation (i) and (ii) we get

$$b = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

Scanned by CamScanner

$$a = \sum_{i=1}^{n} y_i - b \frac{\sum_{i=1}^{n} x_i}{n} = \bar{y} - b\bar{x}$$

where, $\bar{y} = \sum_{i=1}^{n} y_i/n$ & $\bar{x} = \sum_{i=1}^{n} x_i/n$

Fit a st. line to the following set of data

x	1	2	3	4	5
y	3	4	5	6	8

Sol/

The various summations are as

x_i	y_i	$x_i y_i$	x_i^2
1	3	3	1
2	4	8	4
3	5	15	9
4	6	24	16
5	8	40	25
15	26	90	55

$$b = \frac{5 \times 90 - 15 \times 26}{5 \times 55 - (15)^2} = 1.20$$

$$a = \frac{26}{5} - (1.20)\frac{15}{5} = 1.60$$

Therefore linear equation is $y = 1.20 + 1.60x$

Scanned by CamScanner

Algorithm

1.) Read the number of data point, say n
2.) Read data point say x[i] & y[i]
3.) Calculate summation values:

for i=0 to n-1
sx = sx + x[i]
sy = sy + y[i]
sxy = sxy + x[i] * y[i]
sx2 = sx2 + x[i] * x[i]
End for

4.) Calculate the values of parameters
b = ((n * sxy) - (sx + sy)) / ((n * sx2) - (sx)^2)
a = (sy / n) - b * (sx / n)

5.) Display the equation a + bx
6.) Stop

Scanned by CamScanner

Non-Linear Regression

n is a form of regression analysis in which observational data are modeled by a function which is non-linear combination of model parameters and depends on one or more dependend variables. n model can be described in two ways :-

i) By fitting exponential model

→ An exponential model is a model described by the equation $y = ae^{bx}$. In this eqn coefficients a & b are the constants of the exponential model.

→ Non-linear regression can be obtained by fitting exponential through the given data points $(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$

→ Taking the natural log of both sides of the above equation we get

$$\log y = \log ae^{bx}$$
$$= \log a + \log e^{bx}$$
$$= \log y = \log a + bx \Rightarrow y = a + bx$$

This eqn is similar to the eqn $y = a + bx$, Thus we can evaluate parameters a and b by using eqn of linear regression model as

$$b = \frac{n\sum_{i=1}^{n} x_i \log y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} \log y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$\log a = \frac{\sum_{i=1}^{n} \log y_i}{n} - \frac{b\sum_{i=1}^{n} x_i}{n}$$

Scanned by CamScanner

let $\log a = R = \dfrac{\sum_{i=1}^{n} \log y_i}{n} - b \dfrac{\sum_{i=1}^{n} x_i}{n}$

Then, $\log a = R$

$a = e^R$

Fit the following data by using ~~app~~ exponential eqⁿ & Find the value of $x = 13$

x	0	1	3	5	7	9
$f(x)$	1.00	0.891	0.708	0.562	0.447	0.355

Sol^n

x_i	y_i	$\log y_i$	$x_i \log y_i$	x_i^2
0	1.00	0.00	0.00	0
1	0.891	-0.115	-0.115	1
3	0.708	-0.345	-1.035	9
5	0.562	-0.576	-2.88	25
7	0.447	-0.805	-5.635	49
9	0.355	-1.035	-9.315	81
		-2.876	-18.975	165

Now

$$b = \frac{n \sum_{i=1}^{n} x_i \log y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} \log y_i}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$= \frac{6 \times (-18.975) - 25 \times (-2.876)}{6 \times 165 - (25)^2} = -0.115$$

Scanned by CamScanner

$$R = \log a = \frac{\sum_{i=1}^{n} \log y_i}{n} - b \frac{\sum_{i=1}^{n} x_i}{n}$$

$$= \frac{-2876}{6} - (-0.115)\frac{25}{6}$$

$$= -0.0001$$

$$\therefore a = e^{R} = e^{(-0.0001)} = 0.99$$

Then the exponential equation becomes

$$y = (0.99)\, e^{(-0.115)x}$$

For $x = 13$,

$$y = (0.99)\, e^{(-0.115)\,13}$$

$$= 0.22$$

Scanned by CamScanner

Algorithm of Exponential Regression Analysis

1.) Read the number of points, say n
2.) Read the data points, say x[i] & y[i]
3.) Calculate needed summation as below:

```
for i=0 to n-1
    sx = sx + x[i]
    slogy = slogy + logy[i]
    sxy = sxy + x[i] * logy[i]
    sx2 = sx2 + x[i] * x[i]
End for
```

4 Calculate a & b by using formula as

$$b = ((n * sxy) - (sx * slogy)) / ((n * sx2) - (sx * sx))$$

$$r = (slogy / n) - b * (sx / n)$$

$$a = e^{r}$$

5 Display the equation ae^{bx}

6 Stop

Note :- If the given number is in the form as:

$$y = ax^{b}$$

If we take logarithm to both side, we get,

$$\log y = \log x + b \log x$$

This is similar to the form of linear equation $y = a + bx$ and therefore we can evaluate the parameters as:

$$b = \frac{n \sum_{i=1}^{n} \log x_i \log y_i - \sum_{i=1}^{n} \log x_i \sum_{i=1}^{n} \log y_i}{n \sum_{i=1}^{n} (\log x_i)^2 - \left(\sum_{i=1}^{n} \log x_i\right)^2}$$

$$\log a = R = \frac{\sum_{i=1}^{n} \log y_i}{} - b \frac{\sum_{i=1}^{n} \log x_i}{}$$

Scanned by CamScanner

$a = e^R$

Eg:- Given the data table

x	1	2	3	4	5
y	0.5	2	4.5	8	12.5

Fit the power function of the form $y = ax^b$

Sol:

Various summations for the equations are as follows:

x_i	y_i	$\log x_i$	$\log y_i$	$(\log x_i)^2$	$(\log x_i \log y_i)$
1	0.5	0	-0.6931	0	0
2	2	0.6931	0.6931	0.4805	0.4804
3	4.5	1.0986	1.5041	1.2069	1.6524
4	8	1.3863	2.0794	1.9218	2.8827
5	12.5	1.6094	2.5257	2.5903	4.0649
		4.7878	6.1092	6.1995	9.0804

Now

$$b = \frac{5 * (5.0804) - (4.7878) * (6.1092)}{5 * 6.1995 - 4.7878 * 4.7878} = 1.9998$$

$$R = \log a = \frac{6.1092}{5} - 1.9998 * \frac{4.7878}{5}$$

$$= -0.69925$$

$$a = e^R = e^{-0.69925} = 0.5001$$

Scanned by CamScanner

Thus we can obtain power function equation as

$$y = 0.5001 * 7^{1.9998}$$

Fitting the Polynomial

Polynomial is a form of regression analysis in which the relationship between the independent variable x and dependent variable y is modeled as n^{th} degree polynomial in x.

- Let n data points are $(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$ & we want to fit an nth order polynomial through the given data points.
- General form of polynomial of degree m is:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m, \quad m < n$$

→ Now the residual (error) at each data point is:

$$e_i = y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m$$

→ Then the sum of square of residuals is

$$E = \sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m)$$

To find the constant of the polynomial regression model, we equate derivative of residual with respect a_i to zero as follows below:

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{n} 2 (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m) = 0$$

$$\frac{\partial E}{\partial a_i} = \sum_{i=1}^{n} 2 (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m)(-x_i)$$

$\vdots$

Scanned by CamScanner

$$\frac{\partial E}{\partial a_m} = \sum_{i=1}^{n} 2\left(y_i - a_0 - a_1 x_i - a_2 x_i^2 \ldots - a_m x_i^m\right)\left(-x_i^m\right) = 0$$

This gives,

$$\sum_{i=1}^{n} \left(a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_m x_i^m\right) = \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} \left(a_0 x_i + a_1 x_i^2 + a_2 x_i^3 + \ldots + a_m x_i^{m+1}\right) = \sum_{i=1}^{n} x_i y_i$$

$$\vdots$$

$$\sum_{i=1}^{n} \left(a_0 x_i^m + a_1 x_i^{m+1} + a_2 x_i^{m+2} + \ldots + a_m x_i^{m+m}\right) = \Sigma$$

Again

$$n a_0 + a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 + \ldots + a_m \sum_{i=1}^{n} x_i^m = \sum_{i=1}^{n} y_i$$

$$a_0 \sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 + a_2 \sum_{i=1}^{n} x_i^3 + \ldots + a_m \sum_{i=1}^{n} x_i^{m+1} = \sum_{i=1}^{n} x_i y_i$$

$$\vdots$$

$$x_0 \sum_{i=1}^{n} x_i^m + a_1 \sum_{i=1}^{n} x_i^{m+1} + a_2 \sum_{i=1}^{n} x_i^{m+2} + \ldots + a_m \sum_{i=1}^{n} x_i^{2m} = \sum_{i=1}^{n} x_i^m y_i$$

Representing there equations in matrix form as

$$\begin{bmatrix} n & \Sigma x_i & \Sigma x_i^2 & \ldots & \Sigma x_i^m \\ \Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 & \ldots & \Sigma x_i^{m+1} \\ \vdots & & & & \\ \Sigma x_i^m & \Sigma x_i^{m+1} & \Sigma x_i^{m+2} & \ldots & \Sigma x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \\ a_m \end{bmatrix} = \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \\ \vdots \\ \Sigma x_i^m y_i \end{bmatrix}$$

$$E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 \ldots - a_m x_i^m\right)^2$$

Eg:- Fit the second order polynomial to the data in the table below

x	1.0	2.0	3.0	4.0
y	6.0	11.0	18.0	27.0

Sol:

The second order polynomial can be represented as:

$$a_0 n + a_1 \Sigma x_i + a_2 \Sigma x_i^2 = \Sigma y_i$$

$$a_0 \Sigma x_i + a_1 \Sigma x_i^2 + a_2 \Sigma x_i^3 = \Sigma x_i y_i$$

$$a_0 \Sigma x_i^2 + a_1 \Sigma x_i^3 + a_2 \Sigma x_i^4 = \Sigma x_i^2 y_i$$

Then

$$\begin{bmatrix} n & \Sigma x_i & \Sigma x_i^2 \\ \Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 \\ \Sigma x_i^2 & \Sigma x_i^3 & \Sigma x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \\ \Sigma x_i^2 y_i \end{bmatrix}$$

The sum of powers & product are evaluated as below

	x	y	x^2	x^3	x^4	xy	x^2y
	1	6	1	1	1	6	6
	2	11	4	8	16	22	44
	3	18	9	27	54	54	162
	4	27	16	64	108	108	432
Σ	10	62	30	100	354	190	644

Substituting these values

$$\begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 62 \\ 190 \\ 644 \end{bmatrix}$$

Solving these equations

$a_0 = 3$, $a_1 = 2$, $a_2 = 1$

Scanned by CamScanner

Algorithm

1.) Read the number of points, say n
2.) Read the order of polynomial, say m
3.) Read the data points, say x[i], y[i]
4.) Calculate Required summations as below:

```
for i=1 to 2m
    for j=0 to n-1
        sx[i] = sx[i] + pow(x[j], i)
    End for
End for
for i=1 to m
    for j=0 to n-1
        sxy[i] = sxy[i] + y[j] * pow(x[j], i)
    End for
End for
```

5) Compute the LHS coefficient matrix of order (m+1) x (m+1)
6.) Construct the RHS matrix of order (m+1) x 1
7) Solve ~~the~~ for coefficients $a_0, a_1, a_2 \ldots a_m$ using Gauss Elimination method
8) Display the equation $a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$

Scanned by CamScanner

Scanned by CamScanner

Solving system of linear Equations

System of linear Equation

A general set of m linear equations & n unknown is

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_m$$

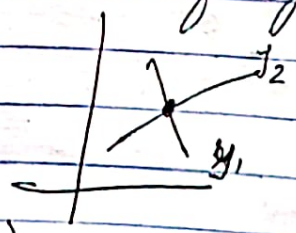
This can be rewritten in matrix form as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

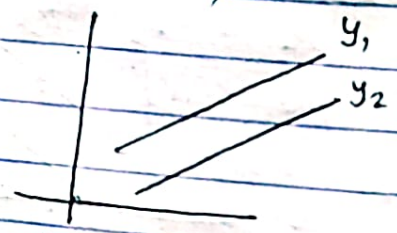
$$\therefore AX = B$$

where A is coefficient matrix, B is called right hand side matrix & x is called solution vector

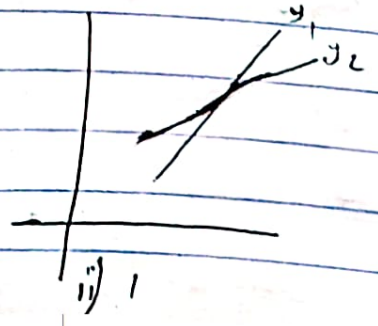
Solution of system of linear Equation



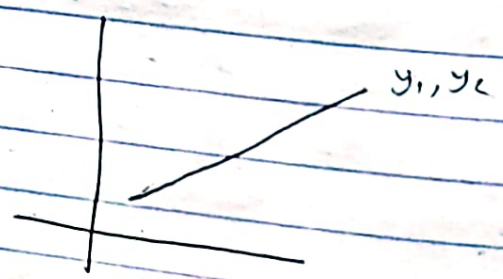
i) unique solution



ii) no solution



iii) 1



iv) [illegible]

Scanned by CamScanner

Solution techniques of system of linear equations can ~~found~~ fundamentally be divided in two groups.

i) Elimination Approach
ii) Iterative Approach

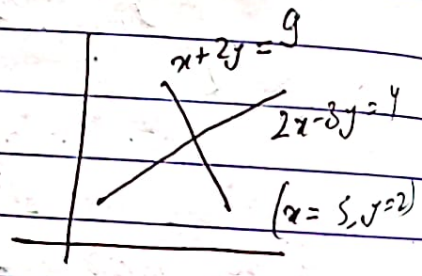
Existence of Solution of System of linear equations

There are four conditions in solution of system of linear equations :-

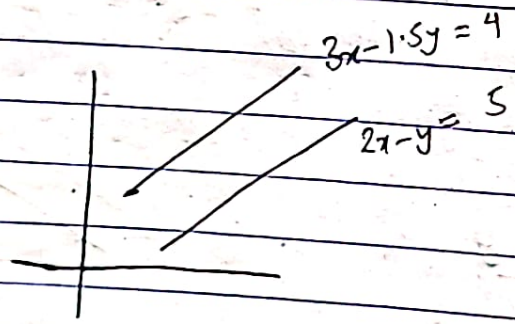
i) Unique Solution
ii) No solution
iii) Ill - condition
iv) No - unique solution

Unique Solution

Since, no other solution exist, so the solution is said to be unique



ii) No Solution

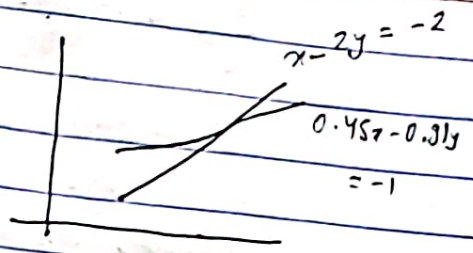


The two lines are parallel, so they never intersect, thus solution doesnot exist.

Scanned by CamScanner

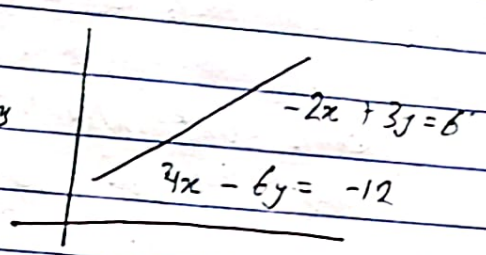
Ill - Condition

It has a solution but it is very difficult to identify the exad point the line intersect.



No - Unique Solution

In this case, there are many solutions



Scanned by CamScanner

Basic Gauss Elimination Method :-

→ Gauss elimination method consists of two steps :-

i) Forward elimination of unknown :-
In this step, them unknown are eliminated from each equations starting with the first equation.

ii) Back substitution :- In this step, starting from the last equation, value of ~~data~~ each of the unknown is found.

Eg :- Use the basic Gauss Elimination method to solve the following system of linear equations:

$$x_1 - 3x_2 + x_3 = 4$$
$$2x_1 - 8x_2 + 8x_3 = -2$$
$$-6x_1 + 3x_2 - 15x_3 = 9$$

Soln/

Given equations are

$$x_1 - 3x_2 + x_3 = 4$$
$$2x_1 - 8x_2 + 8x_3 = -2$$
$$-6x_1 + 3x_2 - 15x_3 = 9$$

Representing these equations in matrix form, we get

$$\begin{bmatrix} 1 & -3 & 1 \\ 2 & -8 & 8 \\ -6 & 3 & -15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 9 \end{bmatrix}$$

or,
$$\begin{bmatrix} 1 & -3 & 1 \\ 0 & -2 & 6 \\ 0 & -15 & -9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -10 \\ 33 \end{bmatrix}.$$

$R_2 = R_2 - 2R_1$
$R_3 = R_3 + 6R_1$

Scanned by CamScanner

or, $\begin{bmatrix} 1 & -3 & 1 \\ 0 & -2 & 6 \\ 0 & 0 & 54 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -10 \\ -108 \end{bmatrix}$ $R_3 = R_3 - \frac{15}{2} R_2$

Now

$54 x_3 = -108 \Rightarrow x_3 = -2$

Substituting the value of x_3 in second row

$-2x_2 + 6x_3 = -10$

$-2x_2 = 2$

$\Rightarrow x_2 = -1$

Substituting the values of x_2 & x_3 in first row

$x_1 - 3x_2 + x_3 = 4$

$\Rightarrow x_1 = 3$

Here, the solution is :-

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ -2 \end{bmatrix}$$

$$\begin{array}{c|ccc} k & 1 & 2 & 3 \\ \hline 1 & 1 & -5 & 1 \\ 2 & 2 & -8 & 8 \\ 3 & -6 & 3 & -15 \end{array} \quad \begin{array}{c} X \\ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{array} = \begin{array}{c} b \\ \begin{bmatrix} 4 \\ -2 \\ 9 \end{bmatrix} \end{array}$$

pivot $= a_{11} = 3$

term $= a_{21} / a_{11} = 2/3$

term $=$ term $\times a_{11} - a_{21} = \frac{2}{3} \times 3 -$

Algorithm for basic Gauss Elimination Method

1) Read the dimension of the system of equation say n
2) Read the coefficient matrix row-wise
3) Read RHS vector
4) Perform forward elimination

```
For k = 1 to n-1
    pivot = a[k][k]
    if (pivot < 0.00000 1)
        Display "Method failed"
    Else
        For i = k+1 to n
            * term = a[i][k] / pivot
            * Multiply row k of coefficient matrix by "term" &
              subtract it from i.
              - term = a[i][k] - term * a[k][k]
            * Multiply row k of b matrix by "term" & subtract
              it from row i
              - term = b[i] - t
        End for
End for
```

5) Perform back substitution as below

```
x[n] = b[n] / a[n][n]
for i = n-1 to 1
    sum = 0
    for j = i+1 to n
        sum = sum = a[i][j] * x[j]
    End for
    x[i] = (b[i] - sum) / a[i][i]
End for
```

Scanned by CamScanner

6) Display solution vector
7) Stop

Drawbacks

Division by 0

May Basic Gauss Elimination may suffer from division by 0 problem at the beginning of each step of forward elimination. This occurs when pivot element is 0.

Round off error

Basic Gauss Elimination method is prone to round off error. This can be serious problem when there are large number of equations as error propagate.

Scanned by CamScanner

Gauss Elimination with partial pivoting.

→ If there are n equations, then there are n-1 forward elimination step.

→ At the beginning of kth step of forward elimination, one finds the maximum of

$$|a_{kk}|, |a_{k+1,k}|, \ldots |a_{nk}|$$

→ Then if the maximum of these values is $|a_{pk}|$ in the pth row, $k \le p \le n$, then swap row p & k

→ The other steps of forward elimination are same as the basic gauss elimination method.

→ The back substitution step is exactly the same as the basic Gauss Elimination method. For eg:-

Solve the following system of linear equations by using Gauss elimination method with partial pivoting

$$2x_1 + x_2 + x_3 = 5$$
$$4x_1 - 6x_2 = -2$$
$$-2x_1 + 7x_2 + 2x_3 = 9$$

Representing these equations in matrix form, we get

$$\begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \\ 9 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -6 & 0 \\ 2 & 1 & 1 \\ 2 & 7 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 9 \end{bmatrix} \quad \{R_1 \leftrightarrow R_2\}$$

Scanned by CamScanner

Algorithm

1.) Read the dimension of system of equations say n
2.) Read coefficient matrix row wise
3.) Read RHS vector
4.) Perform the forward elimination as below

```
For k=1 to n-1
    - Find the largest of a[p][k] for p=k, k+1, ... n
    - swap row k & row p in coefficient matrix
    - swap row k & row p in RHS vector
    - pivot = a[k][k]
    for i=k+1 to n
        * term = a[i][k] / pivot
        * multiply row k of coefficient matrix by
          "term" & subtract it from row i
        * Multiply row k of b matrix by "term"
          & substract it from row i.
    End for
End for
```

6.) Perform the back substitution as below

```
x[n] = b[n] / a[n][n]
for i=n-1 to 1
    sum = 0
        for j=i+1 to n
            sum = sum + a[i][j] * x[j]
        End for
    x[i] = b[i] - sum / a[i][i]
End for
```

7) Display solution vector
8) Stop

Scanned by CamScanner

Gauss - Jordan Method

It is the variation of Gauss Elimination method. The difference between Gauss Elimination & Gauss Jordan are described below.

→ When an unknown is eliminated from an equation is also eliminated from all other equations. All rows are normalized by dividing them by their pivot element. Hence the elimination steps results an identity matrix rather than a triangular matrix

→ Back substituting is not required. We can obtain solution directly from identity matrix obtained from elimination step.

a_{11}

Gauss - Jordan - method

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1' \\ b_2' \\ b_3''' \end{bmatrix}$$

Scanned by CamScanner

Algorithm for Gauss Jordan Method

1.) Normalize the 1st equation by dividing it by its pivot element.
2.) Eliminate x_1 terms from all the other equations.
3.) Now, normalize the second eqn by dividing its by pivot element.
4) Eliminate x_2 from all other eq's above and below the normalize pivotal equation.
5.) Repeat this process until x_n is eliminated from all but the last equation..
6.) The resultant b vector is the solution vector.

Scanned by CamScanner

Matrix Inversion

Inverse matrix can be computed by using Gauss Jordan method

1. Augment the coefficient matrix with identity matrix as below

$$\left[\begin{array}{ccc:ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}\right]$$

2.) Apply Gauss-Jordan method to the augmented matrix to reduce coefficient matrix & identity matrix as below

$$\left[\begin{array}{ccc:ccc} 1 & 0 & 0 & a'_{11} & a'_{12} & a'_{13} \\ 0 & 1 & 0 & a'_{21} & a'_{22} & a'_{23} \\ 0 & 1 & 1 & a'_{31} & a'_{32} & a'_{33} \end{array}\right]$$

3.) Now the right hand side of the augmented matrix is the inverse of original coefficient matrix.

Eg:- Find the inverse of the following matrix

$$\begin{bmatrix} 7 & 3 \\ 5 & 2 \end{bmatrix}$$

The augmented matrix of given matrix is

$$\left[\begin{array}{cc:cc} 7 & 3 & 1 & 0 \\ 5 & 2 & 0 & 1 \end{array}\right]$$

$$\left[\begin{array}{cc:cc} 1 & 3/7 & 1/7 & 0 \\ 5 & 2 & 0 & 1 \end{array}\right] \qquad R_1 = \frac{R_1}{7}$$

$$\left[\begin{array}{cc:cc} 1 & 3/7 & 1/7 & 0 \\ 0 & -1/7 & -5/7 & 1 \end{array}\right] \qquad R_2 = R_2 - 5R_1$$

Scanned by CamScanner

$$\left[\begin{array}{cc:cc} 1 & 3/7 & 1/7 & 0 \\ 0 & 1 & 5 & -7 \end{array}\right] \quad R_2 = -7R_2$$

$$\left[\begin{array}{cc:cc} 1 & 0 & -2 & 3 \\ 0 & 1 & 5 & -7 \end{array}\right] \quad R_1 = R_1 - 3/7\, R_2$$

$\therefore$ Inverse matrix is $\begin{bmatrix} -2 & 3 \\ 5 & -7 \end{bmatrix}$

Triangular Factorization Method

- The coefficient matrix A of a system of linear equations can be factorized (or decomposed) into two triangular matrices L and U such that

$$A = LU$$

where,

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \quad \& \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & u_{2n} \\ \vdots & & & \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

L is known as lower triangular matrix and U is known as upper triangular matrix.

Once A is factorized into L & U, the system of equations $AX = B$ can be expressed as

$$LUX = B$$

$$L(UX) = B \quad \text{—— (i)}$$

Let us assume that

$$UX = Z \quad \text{—— (ii)}$$

Scanned by CamScanner

where Z is an unknown vector.

Substituting equation ② in equation ① we get

$$LZ = B$$

Now, we can solve the system $AX = B$ into two stages.

i) Solve the equation $LZ = B$ for Z by forward substituting

ii) Solve the equation $UX = Z$ for X by back substitution.

A) Dolittle LU Decomposition:

→ Coefficient matrix A of a system of linear equations can be decomposed into two triangular matrices L and U such that,

$$A = LU$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ l_{31} & l_{32} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & & & \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

It implies:

$$a_{ij} = l_{i1}u_{1j} + l_{i2}u_{2j} + l_{i3}u_{3j} + \cdots + l_{ii}u_{ij} \quad \text{for } i<j$$

$$a_{ij} = l_{i1}u_{1j} + l_{i2}u_{2j} + l_{i3}u_{3j} + \cdots + l_{ii}u_{jj} \quad \text{for } i=j$$

$$a_{ij} = l_{i1}u_{1j} + l_{i2}u_{2j} + l_{i3}u_{3j} + \cdots + l_{ij}u_{jj} \quad \text{for } i>j$$

Where

$$u_{ij} = 0 \text{ for } i>j \text{ and } l_{ij} = 0 \text{ for } i<j$$

→ The Dolittle algorithm assumes that all the diagonal elements of L are unity (ie 1) i.e.

$$l_{ii} = 1 \text{ for all } i = 1, 2, \ldots n$$

Scanned by CamScanner

Using eq (i), (ii) & (iii) we can successively determine the elements of U and L as follows.

if $i \le 1$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \qquad j = 1, 2, \ldots n$$

if $i > 1$

$$l_{ij} = \frac{1}{u_{ij}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) \quad j = 1, 2 \ldots i-1$$

where, $l_{11} = 1$, $l_{22} = 1$, $\ldots$ $l_{nn} = 1$ &

$$l_{i1} = \frac{a_{i1}}{u_{11}} \quad \text{for all } i = 2, \ldots n$$

Solve the following systems of linear equation ~~in matrix form~~ by using LU Decomposition method

$$3x_1 + 2x_2 + x_3 = 10$$
$$2x_1 + 3x_2 + 2x_3 = 14$$
$$x_1 + 2x_2 + 3x_3 \quad 14$$

Sol/

Representing the given system of linear equation in matrix form as

$$\begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 14 \\ 14 \end{bmatrix}$$

For $i = 1$:

$$l_{11} = 1$$
$$u_{11} = a_{11} = 3$$
$$u_{12} = a_{12} = 2$$
$$u_{13} = a_{13} = 1$$

For $i = 2$

$$l_{21} = \frac{1}{u_{11}} (a_{21}) = \frac{1}{3} \times 2 = \frac{2}{3}$$

$$l_{22} = 1$$

$$u_{22} = a_{22} - \sum_{k=1}^{1} (l_{2k} u_{kj})$$

$$= a_{22} - l_{21} u_{12}$$

$$= 3 - \frac{2}{3} \times 2 = \frac{5}{3}$$

Scanned by CamScanner

$$u_{23} = a_{23} - l_{21} u_{13} = 2 - \tfrac{2}{3} \cdot 1 = \frac{4}{3}$$

For $i = 3$

$$l_{31} = \frac{1}{u_{11}} (a_{31}) = \frac{1}{3}$$

$$l_{32} = \frac{1}{u_{22}} (a_{32} - l_{31} u_{12})$$

$$= \frac{3}{5} \left(2 - \frac{1}{3} \cdot 2\right) = \frac{4}{5}$$

$$l_{33} = 1$$

$$u_{33} = a_{23} - l_{31} u_{13} - l_{32} u_{23}$$

$$= 3 - \frac{1}{3} \times 1 - \frac{4}{5} \times \frac{4}{3}$$

$$= \frac{24}{15}$$

Thus we have

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 1/3 & 4/5 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 3 & 2 & 1 \\ 0 & 5/3 & 4/3 \\ 0 & 0 & 24/15 \end{bmatrix}$$

Forward Substitution

Solve $LZ = B$ by forward substitution

$$\begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 1/3 & 4/5 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 15 \\ 14 \\ 14 \end{bmatrix}$$

Scanned by CamScanner

Algorithm for solving of linear equations by using LU Decomposition

1) Read A, n, B
2) Set $u_{1j} = a_{1j}$ for $j = 1 \ldots n$
   set $l_{ii} = 1$ for $i = 1 \ldots n$
   set $l_{i1} = a_{i1} / u_{11}$ for $i = 1 \ldots n$
3) For $j = 2$ to n do
   i) For $i = 2$ to j
      Compute $u_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}$

      Repeat i
   ii) For $i = j+1$ to n
      Compute $l_{ij} = \frac{1}{u_{ij}} \times \left[a_{ij} - \sum_{k=1}^{j-1} l \right]$

      Repeat i

   Repeat j
4.) Set $z_1 = b_1$
5.) For $i = 2$ to n
      set $sum = \sum_{j=1}^{i-1} l_{ij} z_j$

      set $z_i = b_i - sum$
   Repeat i
6.) Set $x_n = z_n / u_{nn}$
7.) For $i = n-1$ to 1
      set $sum = \sum_{j=i+1}^{n}$

      set $x_i = (z_i - sum) / u_{ii}$
8) Write result.

Scanned by CamScanner

Cholesky Decomposition Method

Incase A is symmetric, the LU Decomposition can be modified so that the upper factor is the transpose of lower one (or vice versa) & we can factorize A as

$$A = LL^T \quad \text{or} \quad A = U^T U$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} u_{11} & 0 & \cdots & 0 \\ u_{12} & u_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ u_{1n} & u_{2n} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Just as for Dolittle Decomposition, multiplying the matrices of right hand side and comparing it with coefficient matrix of left hand side, we get.

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2} \qquad i = 1 \text{ to } n$$

$$u_{ij} = \frac{1}{u_{ii}} \left[a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj} \right] \qquad j > i$$

This decomposition is called the Cholesky's factorization or the method of square roots

~~Exam~~

Scanned by CamScanner

Eg #1 Factorize the following algorithm by using Choleskys algorithm

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 8 & 22 \\ 3 & 22 & 82 \end{bmatrix}$$

Sol^n/

For $i=1$, $u_{21} = \sqrt{a_{11}} = \sqrt{1} = 1$

$u_{12} = \frac{1}{1}[2] = 2$

$u_{13} = \frac{1}{1}[3] = 3$

For $i=2$

$u_{22} = \sqrt{a_{32} - a_{12}^2}$

$= \sqrt{8-4} = 2$

$u_{23} = \frac{1}{u_{22}}[22 - u_{12} \times u_{13}]$

$= \frac{1}{2}[22 - 2 \times 3] = 8$

For $i=3$

$u_{33} = \sqrt{a_{33} - u_{13}^2 - u_{23}^2}$

$= \sqrt{82 - 9 - 64} = 3$

Eg 2# Solve the following system of linear eq^{n}s by using Cholesky decomposition technique

$$4x_1 + 10x_2 + 8x_3 = 44$$
$$10x_1 + 26x_2 + 26x_3 = 128$$
$$8x_1 + 26x_2 + 61x_3 = 214$$

Sol^n/

Here $\begin{bmatrix} 4 & 10 & 8 \\ 10 & 26 & 26 \\ 8 & 26 & 61 \end{bmatrix}$

For $i=1$ $u_{11} = \sqrt{a_{11}} = \sqrt{4} = 2$

$u_{12} = \frac{1}{2}[10] = 5$

$u_{13} = \frac{1}{2}[8] = 4$

For $i=2$

$u_{22} = \sqrt{a_{22} - u_{12}^2}$

$= \sqrt{26 - 25}$

$= 1$

$u_{23} = \frac{1}{u_{22}}[26 - 5 \times 4]$

$= \frac{1}{1}[26 - 20] = 6$

For $i=3$

$u_{33} = \sqrt{a_{33} - u_{13}^2 - u_{23}^2}$

$= \sqrt{61 - 4^2 - 6^2}$

$= \sqrt{9}$

$= 3$

Scanned by CamScanner

Algorithm for Cholesky method

1.) Read n, A
2.) Set $u_{11} = \sqrt{a_{11}}$
3) set $u_{1j} = a_{1j} / u_{11}$ for $j = 2$ to n
4.) For $j = 2$ to n
    for $i = 2$ to j
        $sum = a_{ij}$
        For $k = 1$ to $i-1$
            $sum = sum - u_{ik} u_{kj}$
        set $u_{ij} = sum / u_{ii}$ if $i < j$
        set $u_{ij} = \sqrt{sum}$ if $i = j$
5.) End of factorization

Scanned by CamScanner

Iterative method for solving system of linear equations.

- Elimination method of solving system of linear equations requires large number of floating point operations therefore severely effect the accuracy of the solution due to round off errors.
- In such cases, iterative methods provide an alternative
- Following are the two iterative method that will discuss

1.) Jacobi iteration method

Jacobi Iteration Method

- Jacobi method is one of the simple iterative method.
- The basic idea behind this method ~~that, will discuss~~ is essentially the same as that for the fixed point method.

Let us consider a system of n equations in n unknowns

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

We rewrite the original systems as

$$x_1 = \frac{b_1 - (a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n)}{a_{11}}$$

$$x_2 = \frac{b_2 - (a_{21}x_1 + a_{23}x_3 + \dots + a_{2n}x_n)}{a_{22}}$$

$$x_n = b_n - (a_{n1}x_1 + a_{n2}x_2 + \dots + a_{n(n-1)}x_{n-1})$$

Scanned by CamScanner

→ Now we can compute $x_1, x_2 \ldots x_n$ by using initial guesses for these values. These new values are again used to compute the next set of x values. The process will continue till we obtain a desired level of accuracy in the x values

→ In general, an iteration for x_i can be obtained from the ith equation as follows

$$x_i^{(k+1)} = \frac{b_i - (a_{i1}x_1^{(k)} + a_{i,i-1}x_{i-1}^{(k)} + a_{i,i+1}x_{i+1}^{(k)} + \ldots + a_{in}x_n^{k})}{a_{ii}}$$

Eg# Obtain the solution of the following system using the Jacobi Iteration Method

$$2x_1 + x_2 + x_3 = 5$$
$$3x_1 + 5x_2 + 2x_3 = 15$$
$$2x_1 + x_2 + 4x_3 = 8$$

Sol

Rewriting given system of equations for unknown as:

$$x_1 = \frac{5 - x_2 - x_3}{2}$$

$$x_2 = \frac{15 - 3x_1 - 2x_3}{5}$$

$$x_3 = \frac{8 - 2x_1 - x_2}{4}$$

<u>Iteration 1</u>

Assume initial guess values of x_1, x_2 & x_3 to be zero

Then we get

$$x_1^{(1)} = \frac{5}{2} = 2.5 \quad ; \quad E_1 = \left|\frac{x_1^{(1)} - x_1^{(0)}}{x_1^{(1)}}\right| = \left|\frac{2.5 - 0}{2.5}\right| = 1$$

Scanned by CamScanner

$x_2^{(1)} = \frac{15}{2} = 3 \qquad E_2 = \left|\frac{3-0}{3}\right| = 1$

$x_3^{(1)} = \frac{8}{4} = 2 \qquad E_3 = \left|\frac{2-0}{2}\right| = 1$

Iteration 2

For the second iteration we have,

$x_1^{(2)} = \frac{5-3-2}{2} = 0 \qquad E_1 = \frac{0-2.5}{0} = \infty$

$x_2^{(2)} = \frac{15-3\times2.5-2\times2}{5} = \frac{3.5}{5} = 0.7 \qquad E_2 = \left|\frac{0.7-3}{0.7}\right| = -3.286$

$x_3^{(2)} = \frac{8-2\times2.5-3}{4} = 0 \qquad E_3 = \left|\frac{0-2}{0}\right| = \infty$

The process can be continued till the values of x reach a desired level of accuracy.

Algorithm for Jacobi Iteration Method

1.) Read n, A & B
2.) Set $x_{0i} = b_i / a_{ii}$ for $i = 1, 2, \ldots n$
3.) For $i = 1, 2 \ldots n$
   i) Set $sum = b_i$
   ii) For $j = 1, 2 \ldots n$ $(j \neq i)$
      set $sum = sum - a_{ij} x_{0j}$
   iii) set $x_i = sum / a_{ii}$
   iv) if $\left|\frac{x_i - x_{0i}}{x_i}\right| > error$, then
      set $x_{0i} = x_i$
      go to step 3
4.) Write result.

Scanned by CamScanner

Gauss Seidel Method

→ Gauss-Seidel Method is an improved version of Jacobi Iteration Method

→ It is difference with Jacobi iteration method is that with the Jacobi method the value of $x_i^{(k)}$ obtained in the kth iteration remains unchanged until the entire $(k+1)^{th}$ iteration has been calculated. But with the Gauss-Seidel Method, we use the new values as soon as they are known

Fig #

→ Given a general set of n equations and n unknowns we have

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n$$

→ We rewrite the original system of equations as

$$x_1 = \frac{b_1 - (a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n)}{a_{11}}$$

$$x_2 = \frac{b_2 - (a_{21}x_1 + a_{23}x_3 + \dots + a_{2n}x_n)}{a_{22}}$$

$$\vdots$$

$$x_n = \frac{b_n - (a_{n1}x_1 + a_{n2}x_2 + \dots + a_{n-1}x_{n-1})}{a_{nn}}$$

→ Now, we start with initial guesses and use above equations to find new estimates for x_i's

→ Remainder, unlike Jacobi Iteration Method Gauss-Seidel method always uses the most recent estimates to calculate the ext estimates for x_i.

→ During the $(k+1)^{th}$ iteration of Gauss-Seidel Method

Scanned by CamScanner

x_i take the form :-

$$x_i^{(k+1)} = \frac{b_i - a_{i1}x_1^{(k+1)} + \dots + a_{i,i-1}x_{i-1}^{(k+1)} + a_{i,i+1}x_{i+1}^{(k)} + \dots + a_{in}x_n^{(k)}}{a_{ii}}$$

for all $k = 0, 1, 2 \dots$

Obtain the solution of the following system using Gauss-Seidel iteration method

$$2x_1 + x_2 + x_3 = 5$$
$$3x_1 + 5x_2 + 2x_3 = 15$$
$$2x_1 + x_2 + 4x_3 = 8$$

Sol^n

Rewriting the system of equations for unknown as

$$x_1 = \frac{5 - x_2 - x_3}{2} \qquad x_2 = \frac{(15 - 3x_1 - 2x_3)}{5}$$

$$x_3 = \frac{(8 - 2x_3 - x_2)}{4}$$

Assuming initial guess values as $x_1 = 0, x_2 = 0, x_3 = 0$

Iteration 1

$$x_1 = \frac{5}{2} = 2.5 \qquad E_1 = \left|\frac{2.5 - 0}{2.5}\right| = 1$$

$$x_2 = \frac{15 - 3\times 2.5 - 0}{5} = 1.5 \qquad E_2 = \left|\frac{1.5 - 0}{1.5}\right| = 1$$

$$x_3 = \frac{8 - 2\times 0 - 1.5}{4} = 0.4 \qquad E_3 = \left|\frac{0.4 - 0}{0.4}\right| = 1$$

Iteration 2

$$x_1 = \frac{5 - 1.5 - 0.4}{2} = 1.6 \qquad E_1 = \left|\frac{1.6 - 2.5}{1.6}\right| =$$

$$x_2 = \frac{(1.5 - 3\times 1.6 - 2\times 0.4)}{5} = 1.9 \qquad E_2 = \left|\frac{1.9 - 1.5}{1.9}\right|$$

$$=$$

Scanned by CamScanner

$$x_3 = \frac{8 - 2\times 1.6 - 1.9}{4} = 0.7 \qquad E_3 = \left|\frac{0.2 - 0.4}{0.7}\right| =$$

We can continue this process until we get $x_1 = 1.0$, $x_2 = 2.0$ & $x_3 = 1.0$

Algorithm of Gauss - Seidel Method :-

1.) Read A, B & n
1.) Set $x_i = b/a_{ii}$ for $i = 1$ to n
3.) Set key = 0
For $i = 1$ to n
i) Set sum = b_i
ii) For $j = 1$ to n $(j \neq i)$
Set sum = sum $- a_{ij} x_j$
iii) set dummy = sum $/ a_{ii}$
iv) If key = = 0 then
if $\left|\frac{\text{dummy} - x_i}{\text{dummy}}\right| >$ error, then
set key = 1 ;
v) set x_i = dummy
4.) If key = 1 then
goto step 3
6.) Write result

Scanned by CamScanner

Eigen vector and Eigen Value of a number

$$Ax = \lambda x$$

$x \rightarrow$ eigen vector

$\lambda \rightarrow$ eigen value

$A \rightarrow$ Matrix

$$Ax - \lambda x = 0$$
$$(A - \lambda I)x = 0$$

Power Method :-

$$y = Ax$$
$$x = \frac{1}{k} y$$

Eg :- $A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

Sol^n/ Assume $x = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

Iteration 1 ;

$$y = Ax = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$$

$k = 2$

$$x = \frac{1}{2} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix}$$

Scanned by CamScanner

$$E_1 = \left|\frac{1-0}{1}\right| = 1 \quad E_2 = \left|\frac{0.5-1}{0.5}\right| = 1 \quad E_2 = \left|\frac{0-0}{0}\right| = \infty$$

Iteration 2

$$y = Ax = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.5 \\ 0 \end{bmatrix}$$

$k = 2.5$

$$x = \frac{1}{2.5} \begin{bmatrix} 2 \\ 2.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1 \\ 0 \end{bmatrix}$$

$$E_1 = \left|\frac{0.8-1}{0.8}\right| = 0.25 \quad E_2 = \left|\frac{1-0.5}{1}\right| = 0.5 \quad E_3 = \left|\frac{0-0}{0}\right| = \infty$$

Iteration 3

$$y = Ax = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0.8 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.8 \\ 2.6 \\ 0 \end{bmatrix}$$

$k = 2.8$

$$x = \frac{1}{2.8} \begin{bmatrix} 2.8 \\ 2.6 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.9286 \\ 0 \end{bmatrix}$$

$$E_1 = \left|\frac{1-0.8}{1}\right| = 0.2 \quad E_2 = \left|\frac{0.9286-1}{0.9286}\right| = 0.0769$$

$$E_3 = \left|\frac{0-0}{0}\right| = \infty$$

Iteration 4

$$y = Ax = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0.9286 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.86 \\ 2.93 \\ 0 \end{bmatrix}$$

$k = 2.93$

Scanned by CamScanner

$$x = \frac{1}{2.93}\begin{bmatrix} 2.86 \\ 2.93 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.98 \\ 1 \\ 0 \end{bmatrix}$$

$$E_1 = \left|\frac{0.98 - 1}{0.98}\right| = 0.02 \quad E_2 = \left|\frac{1 - 0.9286}{1}\right| = 0.07$$

$$E_3 = \left|\frac{0-0}{0}\right| = \infty$$

Iteration 5

$$y = Ax = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 0.98 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.98 \\ 2.96 \\ 0 \end{bmatrix}$$

$$k = 2.98$$

$$x = \frac{1}{2.98}\begin{bmatrix} 2.98 \\ 2.96 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}$$

$$E_1 = \left|\frac{1 - 0.98}{1}\right| = 0.02 \quad E_2 = \left|\frac{0.99 - 1}{0.99}\right| = 0.01$$

$$E_3 = \left|\frac{0-0}{0}\right| = \infty$$

Iteration 6

$$y = Ax = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.98 \\ 2.99 \\ 0 \end{bmatrix}$$

$$k = 2.99$$

$$x = \frac{1}{2.99}\begin{bmatrix} 2.98 \\ 2.99 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.99 \\ 1 \\ 0 \end{bmatrix}$$

$$E_1 = \left|\frac{0.99 - 1}{0.99}\right| = 0.01 \quad E_2 = \left|\frac{1 - 0.99}{1}\right| = 0.01 \quad E_3 = \left|\frac{0-0}{0}\right| = \infty$$

Scanned by CamScanner

Iteration #7

$$y = Ax = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0.99 \\ \cancel{0.991} \\ 0 \end{bmatrix} = \begin{bmatrix} 2.99 \\ 2.96 \\ 0 \end{bmatrix}$$

$$\cancel{\left| \varepsilon_1 = \left| \frac{0.99 - 1}{0.99} \right| = 0.01 \quad \varepsilon_2 = \left| \frac{-0.99}{1} \right| = 0.01 \quad \varepsilon_3 = \left| \frac{0 - 0}{0} \right| = \infty \right.}$$

$$k = 2.99$$

$$x = \frac{1}{2.99} \begin{bmatrix} 2.99 \\ 2.96 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}$$

$$\therefore x = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix}$$

Scanned by CamScanner

Algorithm of Power Method

1.) Read A, n
2.) Read guess vector x
3.) Compute y vector as below
    for i = 1 to n
        y[i] = 0
        for j = 1 to n
            y[i] = y[i] + a[i][j] * x[j]
4.) Set k = fabs (y[1])
5.) Find the largest absolute element of y vector and put that element in k
    for i = 2 to n
        if (k < fabs (y[i]))
            k = y[i]
6.) Find the new value of x vector as below :
    for i = 1 to n
        nx[i] = $\frac{1}{k}$ * y[i]
7.) Compute and check error precision
    for i = 1 to n
    { $E[i] = \frac{(nx[i] - x[i])}{nx[i]}$
        if (E[i] > E)
            break;
    }
8.) If i == n+1
        Display eigen value & eigen vector
    else
        for i = 1 to n
            x[i] = nx[i];
        goto step 3;
9. STOP

Scanned by CamScanner

Unit 4 : Numerical Differential & Integration

Numerical Differentiation :

used in :

1.) Tabulated function (Discrete function)
2.) Complicated function (Continuous function)

Differentiating continuous function :-

$f'(x)$ can be find when function $f(x)$ itself is available.

Two point forward Difference formula

Consider a small increment $\Delta x = h$ in x

According to Taylor's theorem

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)h^2}{2!} + \dots$$

$$\text{or, } f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(x)h}{2!} - \dots$$

$$\therefore f'(x) = \frac{f(x+h) - f(x)}{h} - E$$

Thus if h is chosen to be sufficiently small $f'(x)$ can be approximated by

$$f'(x) = \frac{f(x+h) - f(x)}{h} \quad \text{--- (i)}$$

with the truncation error of

$$E = -\frac{f''(h)h}{2!} \quad \text{--- (ii)}$$

The equation (i) is called two point forward differen formula. This equation is also called forward difference quotient.

Scanned by CamScanner

Eg:- Find the value of derivative at $x = 1$ for the function $f(x) = x^2$ by using $h = 0.2$ & 0.05

Sol'n

For $h = 0.2$

We know, $f'(x) = \frac{f(x+h) - f(x)}{h}$

$$f'(1) = \frac{f(1+0.2) - f(1)}{0.2} = \frac{1.44 - 1}{0.2} = 2.2$$

True value of derivative at $x = 1$ is

$f'(x) = 2x = 2$

$\therefore$ Error $= \left|\frac{2 - 2.2}{2}\right| = \left|\frac{0.2}{2}\right| = 0.1$

$\therefore$ % error $= 0.1 \times 100\% = 10\%$

For $h = 0.05$

$$f'(1) = \frac{f(1+0.05) - f(1)}{0.05} = \frac{(1.05)^2 - 1}{0.05} = 2.05$$

True value of derivative at $x = 1$ is

$f'(x) = 2x = 2$

$\therefore$ % Error $= \left|\frac{2 - 2.05}{2}\right| \times 100\% = 2.5\%$

From this example, it is clear that error decreases as the value of h becomes smaller.

Scanned by CamScanner

Two point Backward difference formula:

Consider a small change $\Delta x = h$ in x. According to Taylor's theorem, we have

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)h^2}{2!} - \dots$$

$$f'(x) = \frac{-f(x-h) + f(x)}{h} + \frac{f''(x)h^2}{2!} - \dots$$

$$f'(x) = \frac{f(x) - f(x-h)}{h} + E$$

Thus if h is chosen to be sufficiently small $f'(x)$ can be approximated by

$$f'(x) = \frac{f(x) - f(x-h)}{h} \quad \text{——} \quad (1)$$

with a truncation error of

$$E = \frac{f''(x)h}{2} \quad \text{——} \quad (2)$$

Eqn (1) is two point Backward difference formula.

Eg:-

$x = 1$. for $f(x) = x^2$ using $h = 0.2$ & 0.05

Sol/

For $h = 0.2$

We know, $f'(x) = \frac{f(x) - f(x-h)}{h}$

$$f'(1) = \frac{f(1) - f(1-0.2)}{0.2} = \frac{1 - 0.64}{0.2} = 1.8$$

$$E = \frac{2 - 1.8}{2} = \frac{0.2}{2} = 0.1\% = 0.1 \times 100 = 10\%$$

#

1.)
2.)
3.)
4.)
5.)
6.)

Scanned by CamScanner

For $h = 0.05$

$$f'(1) = \frac{f(1) - f(1-0.05)}{0.05} = \frac{1 - 0.9025}{0.05} = 1.95$$

Algorithm for two point forward difference formula

Read the value at which the derivative value is needed, say x

Read interval gap, say h

Calculate $f(x)$ & $f(x+h)$

Calculate $d = f'(x) = (f(x+h) - f(x)) / h$

Display the value of derivative

Stop.

Scanned by CamScanner

Three point formula

From Taylor's series

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)h^2}{2!} + \dots \quad (1)$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)h^2}{2!} - \dots \quad (2)$$

Subtracting eqn (2) from (1) we get

$$f(x+h) - f(x-h) = f'(x)2h + \dots$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + E$$

Thus if h is chosen to be sufficiently small, $f'(x)$ can be approximated by

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} \quad (3)$$

with a truncation error of

$$E = \frac{f'''(x)\, 2h^2}{3!} \quad \dots (4)$$

The equation (3) is called the three point difference formula. This is also known as central difference quotient.

Scanned by CamScanner

Eg:- Find the value of derivative at $x = 1$ for the function $f(x) = x^2$ by using $h = 0.2$ & 0.05

Sol/

$h = 0.2$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

$$f'(x) = \frac{f(1+0.2) - f(1-0.2)}{2 \times 0.2}$$

$$= \frac{1.44 - 0.64}{0.4}$$

$$= 2$$

$h = 0.05$

$$f'(1) = \frac{f(1+0.05) - f(1-0.05)}{2 \times 0.05}$$

$$= \frac{(1.05)^2 - 0.9025}{2 \times 0.05}$$

$$= \frac{0.2}{0.1}$$

$$= 2$$

Scanned by CamScanner

Differentiating Discrete (Tabulated) Function:

i) Derivative using Newton's divided difference formula.

We know that,

$$P_n(x) = f(x) = f[x_0] + f[x_1, x_0](x-x_0) + f[x_2, x_1, x_0](x-x_0)(x-x_1) + f[x_3, x_2, x_1, x_0](x-x_0)(x-x_1)(x-x_2) + \dots$$

The first derivative of eqn (i) w.r.t x, we get

$$f'(x) = f[x_1, x_0] + f[x_2, x_1, x_0]\{(x-x_1) + (x-x_0)\} + f[x_3, x_2, x_1, x_2]\{(x-x_1)(x-x_2) + (x-x_0)(x-x_2) + (x-x_0)(x-x_1)\} + \dots$$

→ Now, we can find $x = a$, Putting the value of x in eqn (2)

→ Again, differentiating eqn (ii) w.r.t x.

$$f''(x) = 2f[x_2, x_1, x_0] + 2f[x_3, x_2, x_1, x_0]\{(x-x_0) + (x-x_1) + (x-x_2)\} \quad \text{— (iii)}$$

By putting $a = x$ in eqn (3) we get the value of 2nd derivative at $x = a$

Find $f'(10)$ from the following data points

x	3	5	11	27	34
$f(x)$	-13	23	899	17315	35606

Sol:

x	$f(x)$	1st dd	2nd dd	3rd dd	4th dd
3	-13				
		18			
5	23		16		
		146		1	
11	899		40		
		1026		1	0
27	17315		69		
		2613			
34	35606				

We know that

$$f'(x) = f[x_1, x_0] + f[x_2, x_1, x_0]\{(x - x_0) + (x - x_1)\} + f[x_3, x_2, x_1, x_0]\{(x - x_1)(x - x_2) + (x - x_0)(x - x_2) + (x - x_0)(x - x_1)\}$$

$$f'(10) = 18 + 16\{(10-3)(10-5)\} + 1\{(10-5)(10-11) + (10-3)(10-11) + (10-3)(10-5)\}$$

$$= 233$$

Scanned by CamScanner

~~Rea~~ Algorithm :-

1.) Read the number of point, say n
2.) Read n data points
3.) Read the value at which derivative is needed, say x
4.) Compute dividend difference
5.) Compute differentiated value
6.) Print the value of 1st derivative

#

Derivative using Newton forward difference formula

$$P_n(x) = f(x_0) + s\Delta f(x_0) + \frac{1}{2!}s(s-1)\,\Delta^2 f(x_0) + \frac{1}{3!}s(s-1)(s-2)\,\Delta^3 f(x_0) + \cdots \quad \text{—— (1)}$$

Now, derivative of eq (1) w.r.t x

$$f'(x) = \frac{df(x)}{ds}\cdot\frac{ds}{dx}$$

$$= \Delta f(x_0) + \frac{1}{2!}(2s-1)\,\Delta^2 f(x_0) + \frac{1}{3!}(3s^2-6s+2)\,\Delta^3 f(x_0) + \ldots\Big]\frac{ds}{dx}$$

Since, $s = \frac{x-x_0}{h} \Rightarrow \frac{ds}{dx} = \frac{1}{h}$ —— (2)

Now eq^n (2) becomes

$$f'(x) = \frac{1}{h}\Big[\Delta f(x_0) + \frac{1}{2!}(2s-1)\,\Delta^2 f(x_0) + \frac{1}{3!}(3s^2-6s+2)\,\Delta^3 f(x_0) + \cdots \quad \text{—— (3)}$$

→ By Putting $x=a$ in eq^n (3) we get the 1st derivative value at $x=a$

→ Again, differentiating eq^n (iii) w.r.t x we get

$$f''(x) = \frac{df'(x)}{ds}\,\frac{ds}{dx}$$

Scanned by CamScanner

$$f''(x) = \frac{1}{h}\left[\Delta^2 f(x_0) + (s-1)\Delta^3 f(x_0) + \dots\right]\frac{ds}{dx}$$

$$= \frac{1}{h^2}\left[\Delta^2 f(x_0) + (s-1)\Delta^3 f(x_0) + \dots\right] \quad \text{—— (4)}$$

By putting $x = a$ in eqn (4) we get the value of 2nd derivative at $x = a$.

Find the 1st & 2nd derivative of the function tabulated below

x	1.0	1.2	1.4	1.6	1.8	2.0
fx	0.0	0.128	0.544	1.296	2.432	4.0

Sol: $f'(1.1)$ & $f''(1.1)$ $\qquad s = 0.5$

x	fx				
1	0				
		0.128			
1.2	0.128		0.288		
		0.416		0.048	
1.4	0.544		0.336		0
		0.752		0.048	
1.6	1.296		0.384		0
		1.136		0.048	
1.8	2.432		0.432		
		1.568			
2	4				

$$f'(1) = \frac{1}{0.2}\left[0.128 + \frac{1}{2}(2\times 0.5 - 1)\times 0.288 + \frac{1}{6}(3\times 0.5^2 - 6\times 0.5 + 2)\times 0.048\right]$$

$$= \cancel{\frac{271}{200} = 1.355} \quad 0.63$$

Derivative Using Newton's Bd formula

Backward

We know that Newton's Bd formula:

$$P_n(x) = f(x) = f(x_n) + \nabla f(x_n)\, s + \frac{1}{2!}\nabla^2 f(x_n)\, s(s+1) + \frac{1}{3!}\nabla^3 f(x_n)\, s(s+1)(s+2) + \dots \quad \text{--- (1)}$$

where, $s = \dfrac{(x - x_n)}{h}$

Differentiating eqn (1) w.r.t x, we get

$$f'(x) = \frac{d f(x)}{ds} \cdot \frac{ds}{dx}$$

$$= \left[\nabla f(x_n) + \frac{1}{2!}\nabla^2 f(x_n)(2s+1) + \frac{1}{3!}\nabla^3 f(x_n)(3s^2+6s+2) + \dots\right]\frac{ds}{dx}$$

Since, $s = \dfrac{(x - x_n)}{h} \Rightarrow \dfrac{ds}{dx} = \dfrac{1}{h}$

Thus eqn (2) becomes

$$f'(x) = \frac{1}{h}\left[\nabla f(x_n) + \frac{1}{2!}\nabla^2 f(x_n)(2s+1) + \frac{1}{3!}\nabla^3 f(x_n)(3s^2+6s+2) + \dots\right]$$

→ By putting $x = a$ in eqn (3) we can calculate the first derivative at point $x = a$

→ Again differentiating equation (3) w.r.t x we get

$$f''(x) = \frac{d f'(x)}{ds} \cdot \frac{ds}{dx}$$

$$f''(x) = \frac{1}{h}\left[\nabla^2 f(x_n) + \nabla^3 f(x_n)(s+1) + \dots\right]\frac{ds}{dx}$$

$$f''(x) = \frac{1}{h^2}\left[\nabla^2 f(x_n) + \nabla^3 f(x_n)(s+1) + \dots\right] \quad \text{--- (4)}$$

Scanned by CamScanner

→ By putting $x = a$ in equation (4) we can calculate the 2nd derivative at point $x = a$

Eg:- Find the first derivative of the function tabulated below at $x = 9$

x	5	6	7	8	9
$f(x)$	10.0	14.5	19.5	25.5	32.0

$$s = \frac{9-9}{4} = 0$$

x	$f(x)$				
5	10				
		4.5			
6	14.5		0.5		
		5		0.5	
7	19.5		1		$\boxed{1}$
		6		$\boxed{-0.5}$	
8	25.5		$\boxed{0.5}$		
		$\boxed{6.5}$			
9	$\boxed{32.}$				

$$f'(9) = \frac{1}{1}\left[32 + \frac{1}{2!} \times 6.5\,(2 \times 0 + 1) + \frac{1}{3!} \times 0.5\,(3 \times 0^2 + 6 \times 0 + 2) + \frac{1}{4!} \cdots\right]$$

$=$

Maxima & Minima of Tabulated Function

We know that Newton's forward difference formula :-

$$f(x) = f(x_0) + s\Delta f(x_0) + \frac{1}{2!}s(s-1)\Delta^2 f(x_0) + \frac{1}{3!}s(s-1)(s-2)\Delta^3 f(x_0) + \dots \quad \text{—— (1)}$$

where, $s = \frac{x - x_0}{h}$

Differentiating eq^n (1) w.r.t x we get,

$$f'(x) = \frac{1}{h}\left[\Delta f(x_0) + \frac{1}{2!}(2s-1)\Delta^2 f(x_0) + \frac{1}{3!}(3s^2-6s+2)\Delta^3 f(x_0) + \dots\right] - (2)$$

For maxima & minima, $f'(x) = 0$

Now

$$\frac{1}{h}\left[\Delta f(x_0) + \frac{1}{2!}(2s-1)\Delta^2 f(x_0) + \frac{1}{3}(3s^2-6s+2)\Delta^3 f(x_0)\right] = 0$$

or $\Delta f(x_0) + \frac{1}{2!}(2s-1)\Delta^2 f(x_0) + \frac{1}{3!}(3s^2-6s+2)\Delta^3 f(x_0) = 0$

or $\left\{\Delta f(x_0) - \frac{1}{2}\Delta^2 f(x_0) + \frac{1}{3}\Delta^3 f(x_0)\right\} + \left\{\Delta^2 f(x_0) - \Delta^3 f(x_0)\right\}s + \left\{\frac{1}{2}\Delta^3 f(x_0)\right\}s^2 = 0$

or, $as^2 + bs + c = 0$

where, $a = \frac{1}{2}\Delta^3 f(x_0)$

$b = \Delta^2 f(x_0) - \Delta^3 f(x_0)$

$c = \Delta^2 f(x_0) - \frac{1}{2}\Delta^2 f(x_0) + \frac{1}{3}\Delta^3 f(x_0)$

Equation (3) is quadratic in s and can be solved. The value of x can be computed from the relation $x = x_0 + sh$

Eg :-

Q:- Find Maximum & minimum value of following tabulated function

x	0	1	2	3
$f(x)$	-5	-7	-3	13

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
0	-5			
		~~-2~~ -2		
1	-7			
			6	
		~~-10~~ 4		6
2	-3		12	
		16		
3	13			

$$a = \frac{1}{2}\Delta^3 f(x_0) = \frac{1}{2} \times 6 = 3$$

$$b = \Delta^2 f(x_0) - \Delta^3 f(x_0) = 6 - 6 = 0$$

$$c = \Delta f(x_0) - \frac{1}{2}\Delta^2 f(x_0) + \frac{1}{3}\Delta^3 f(x_0)$$

$$= -2 - \frac{1}{2} \times 6 + \frac{1}{3} \times 6$$

$$= -2 - 3 + 2$$

$$= -3$$

Now

$$as^2 + bs + c = 0$$

$$\Rightarrow 3s^2 - 3 = 0$$

$$\therefore s = \pm 1$$

$$x = x_0 + sh = 0 + 1 \times 1 \text{ or } 0 - 1 \times 1$$

$$= 1 \text{ or } -1$$

$$\Rightarrow x = \pm 1$$

Scanned by CamScanner

$f'(x) = 3s^2 - 3 = 0$

$f''(x) = 6s$

$\therefore$ We have maxima at -1 and minima at 1

Scanned by CamScanner

Date
Page

Scanned by CamScanner