Numerical Methods

Swarnadeep Biswas

Assam University Silchar

April 16, 2019

Presentation Outline

1. Finding roots of non linear equation
2. Direct Solution of Linear Equation
3. Interpolation and Curve fitting
4. Curve fitting :Regression
5. Numerical Intregation
6. Numerical solution of Ordinary Differential equation
7. References

Horner's Rule

$$
\begin{aligned}
f(x) &= a_n x^n + a_{n-1}x^{n-1} + ... + ax + a_0 \\
&= ((...((a_n x + a_{n-1})x + ... + a)x + a_0 \\
p_n &= a_n \\
p_{n-1} &= p_n x + a_{n-1} \\
&\cdots \\
&\cdots \\
p_1 &= p_2 x + a_1 \\
p_0 &= p_1 x + a_0
\end{aligned}
$$

Question

Evaluate the ploynomial

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

using Horner's rule at $x = 2$

Write a fortran program to evaluate the above problem

Bisection Method

- If $f(x)$ is real and continious in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs ie
$$f(a)f(b) < 0$$
then there is one real root between a and b.

Bisection Method

- If $f(x)$ is real and continious in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs ie
$$f(a)f(b) < 0$$
then there is one real root between a and b.
- We define a new point x_0 to be mid point of a and b ie
$$x_0 = \frac{a+b}{2}$$

Bisection Method

- If $f(x)$ is real and continious in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs ie
$$f(a)f(b) < 0$$
then there is one real root between a and b.
- We define a new point x_0 to be mid point of a and b ie
$$x_0 = \frac{a+b}{2}$$
- Three condition arises
  1. if $f(x_0) = 0$, we have a root at x_0
  2. if $f(x_0)f(a) < 0$, there is a root between x_0 and a
  3. if $f(x_0)f(b) < 0$, there is a root between x_0 and b

Question

Find the root of the equation

$$e^x - x - 2 = 0$$

using Bisection Method.

Write a fortran program to evaluate the above problem

Question

Find the root of the equation

$$x^2 + x - 2 = 0$$

using Bisection Method.

Write a fortran program to evaluate the above problem

False Position

- If $f(x)$ is real and continious in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs ie
$$f(a)f(b) < 0$$
then there is one real root between a and b.

False Position

- If $f(x)$ is real and continious in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs ie
$$f(a)f(b) < 0$$
then there is one real root between a and b.
- We join the points a and b by a straight line.The point of intersection of this line with the x axis(x_0) give the improved estimate of the root and is called the false position of the root.

$$x_0 = a - \frac{f(a)(b-a)}{f(b)-f(a)}$$

False Position

- If $f(x)$ is real and continious in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs ie
$$f(a)f(b) < 0$$
then there is one real root between a and b.
- We join the points a and b by a straight line.The point of intersection of this line with the x axis(x_0) give the improved estimate of the root and is called the false position of the root.
$$x_0 = a - \frac{f(a)(b-a)}{f(b)-f(a)}$$
- Three condition arises
  1. if $f(x_0) = 0$, we have a root at x_0
  2. if $f(x_0)f(a) < 0$, there is a root between x_0 and a
  3. if $f(x_0)f(b) < 0$, there is a root between x_0 and b

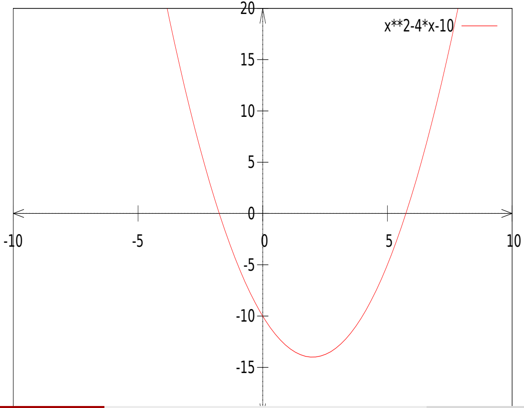
Question

Find the root of the equation

$$x^2 - 4x - 10 = 0$$

using False Position Method.

Write a fortran program to evaluate the above problem



Newton-Raphson Method

Let us assume that x_1 is an approximate root of $f(x) = 0$.
Draw a tangent at the curve $f(x)$ at x_1, the point of intersection of this tangent with the x-axis gives the second approximation of the root.Let the point of intersection be x_2.Then

$$tan(\alpha) = \frac{f(x_1)}{x_1 - x_2} = f'(x_1) \tag{1}$$

Solving for x_2 we obtain

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \tag{2}$$

This is called the *Newton-Raphson* formula. The next approximation would be

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} \tag{3}$$

In general,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4}$$

This method of successive approximation is called the *Newton-Raphson method*. The process will be terminated when the difference between two successive values is within a prescribed limit.

Newton-Raphson from Taylor series expansion

Consider a small interval such that

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

We can express $f(x_{n+1})$ using Taylor series expansion as follows

$$f(x_{n+1}) = f(x_n) + f'(x_n)h + f''(x_n)\frac{h^2}{2!} + \cdots$$

Neglecting higher order terms from second order derivative we have

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

If x_{n+1} is a root of $f(x)$, then

$$f(x_{n+1}) = 0 = f(x_n) + f'(x_n)h$$

Then

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

Therefore,

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

Secant Method

Secant method,like false position and bisection methods, uses two initial estimates but doesnot require that they must bracket the root.

slope of the secant line passing through x_1 and x_2 is given by

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

solving we get

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

By adding and substracting $f(x_2)x_2$ to the numerator and rearranging we get,

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

So the general form is

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

Secant Algorithm

1. Decide two initial points x_1 and x_2, accuracy level required
2. Compute $f_1 = f(x_1)$ and $f_2 = f(x_2)$
3. Compute

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

4. Test for the accuracy of x_3
   if $abs((x_3 - x_2)/x_3) > E$ then
   set $x_1 = x_2$ and $f_1 = f_2$
   set $x_2 = x_3$ and $f_1 = f(x_3)$
   go to step 3
   otherwise,
   root=x_3
   print result
5. stop

Fixed Point Iteration Methods

Any function in the form of

$$f(x) = 0 \tag{1}$$

can be manupluted such that x is on the left-hand side of the equation as shown below

$$x = g(x) \tag{2}$$

Equation (1) and Equation (2) are equivalent and therefore the roots of equation (2) is also roots of equation (1)

The root of equation (2) is the point of intersection of the curves $y = x$ and $y = g(x)$

This intersection point is known as the *fixed point* of $g(x)$

Fixed Point Iteration Methods

Any function in the form of

$$f(x) = 0 \tag{1}$$

can be manupluted such that x is on the left-hand side of the equation as shown below

$$x = g(x) \tag{2}$$

Equation (1) and Equation (2) are equivalent and therefore the roots of equation (2) is also roots of equation (1)

The root of equation (2) is the point of intersection of the curves $y = x$ and $y = g(x)$

This intersection point is known as the *fixed point* of $g(x)$

For example

$$x^2 - 4x - 10 = 0 \tag{3}$$

can be writen as

$$x = \frac{x^2 - 10}{4}$$

Fixed Point Iteration Methods

Any function in the form of

$$f(x) = 0 \tag{1}$$

can be manupluted such that x is on the left-hand side of the equation as shown below

$$x = g(x) \tag{2}$$

Equation (1) and Equation (2) are equivalent and therefore the roots of equation (2) is also roots of equation (1)

The root of equation (2) is the point of intersection of the curves $y = x$ and $y = g(x)$

This intersection point is known as the *fixed point* of $g(x)$

For example

$$x^2 - 4x - 10 = 0 \tag{3}$$

can be writen as

$$x = \frac{x^2 - 10}{4}$$

If x_0 is the initial guess to a root, then the next approximation is given by

$$x_1 = g(x_0)$$

Further approximation is given by

$$x_2 = g(x_1)$$

This iteration can be expressed in general form as

$$x_{i+1} = g(x_i)$$

which is called the *fixed point iteration formula*

How to find Multiple roots ?

A polynomial of degree n can be expressed as

$$\boxed{p(x) = (x - x_r)q(x)} \tag{1}$$

where x_r is the root of the polynomial $p(x)$ and $q(x)$ is the quotient polynomial of degree $n - 1$.
Synthetic division is performed as follows.
Let

$$p(x) = \sum_{i=0}^{n} a_i x^i$$

and

$$q(x) = \sum_{i=0}^{n-1} b_i x^i$$

.
Putting in equ(1) we have

$$a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0 = (x - x_r)(b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + ... + b_1 x + b_0) \tag{2}$$

Comparing the coefficient of like powers we have

How to find Multiple roots ?

$$
\begin{aligned}
a_n &= b_{n-1} \\
a_{n-1} &= b_{n-2} - x_r b_{n-1} \\
&\vdots \\
a_1 &= b_0 - x_r b_1 \\
a_0 &= -x_r b_0
\end{aligned}
$$

That is

$$\boxed{a_i = b_{i-1} - x_r b_i, \quad i = n, n-1, \cdots 0; \quad b_n = 0}$$

Then

$$\boxed{b_{i-1} = a_i + x_r b_i, \quad i = n \cdots 1; \quad b_n = 0}$$

Question

The polynomial equation

$$p(x) = x^3 - 7x^2 + 15x - 9 = 0$$

has a root at $x = 3$. Find the quotient poynomial $q(x)$ such that

$$p(x) = (x-3)q(x)$$

Write a fortran program to find the cofficient of $q(x)$

Multiple roots by Newton's Method

We can locate all real roots of a polynomial by repeatadly applying Newton-Rapson method and Polynomial deflation to obtain polynomials of lower and lower degrees.
After $(n-1)$ deflations, the quotient is a linear polynomial of type

$$\boxed{a_1 x + a_0 = 0}$$

and therefore the final root is given by

$$\boxed{x_r = -\frac{a_0}{a_1}}$$

Algorithm

1 Obtain the degree and coefficient of polynomial

2 Decide an initial estimate for the first root (x_0) and error citerion

3 Initiate do loop for n > 1

4 Find the root using Newton-Raphson algorithm

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

5 Root (n)=x_r

Algorithm

6 Deflate the polynomial using synthetic division algorithm and make the factor polynomial as the new polynomial of order $n-1$.

7 set $x_0 = x_r$ (initial value of next root)

8 end do

9 root(1) $= \frac{-a_0}{a_1}$

10 stop

Question

Find all the roots of the polynomial equation

$$f(x) = x^2 - 4x - 10 = 0$$

using Newton-Raphson method and synthetic division method.
Also write a fortran code for it.

Presentation Outline

Introduction

Mathematical model of many of the real world problems are either linear or can be approximately resonably well using linear relationships.A linear equation involving two variables x and y has the standard form

$$\boxed{ax + by = c}$$

A linear equation with n variables has the form

$$\boxed{a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n = b}$$

where $a_i(i = 1, 2, \cdots, n)$ are real numbers and at least one of them is not zero.The main concern is to solve for $x_i(i = 1, 2, \cdots, n)$,given the valus of a_i and b.
A system of n linear equations is represented generally as

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

In matrix notation above equation can be expressed as

$$\boxed{Ax = b}$$

Introduction

Where A is an $n \times n$ matrix,b is an n vector, and x is a vector of n unknowns,given by

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

There are two different approach to solve *simultaneous equations*

- Elimination approach
- Iterative approach

Elimination approach,also known as *direct method*,reduces the given system of equation to a form from which the solution can be obtained by simple substitution.There are following methods

- Basic Gauss elimination method
- Gauss elimination with pivoting
- Gauss-Jordan method
- LU decomposition methods
- Matix inversion method

Example

Solve the following system of equations by the process of elimination

$$\begin{aligned} 3x + 2y + z &= 10 \\ 2x + 3y + 2z &= 14 \\ x + 2y + 3z &= 14 \end{aligned}$$

Basic Gauss Elimination Method

Gauss elimination method proposes a systematic strategy for reducing the system of equations to upper triangular form using the *forward elimination* approach and then for obtaining values of unknown using the *back substitution* process.

Consider a general set of n equations in n unknowns:

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

Algorithm

(1) Arrange equation such that $a_{11} \neq 0$

(2) Eliminate x_1 from all but the first equation.This can be done as follows

  (i) Normalise the first equation by dividing it by a_{11}

  (ii) Substract from the second eq a_{21} times the normalised first equation.

Algorithm

The result is

$$\left[a_{21} - a_{21}\frac{a_{11}}{a_{11}}\right]x_1 + \left[a_{22} - a_{21}\frac{a_{12}}{a_{11}}\right]x_2 + \cdots = b_2 - a_{21}\frac{b_{11}}{a_{11}}$$

The new second equation is

$$0 + a'_{22}x_2 + \cdots + a'_{2n}x_n = b'_2$$

(iii) Similarly we get for other equations

$$\begin{aligned} a'_{22}x_2 + \cdots + a'_{2n}x_n &= b'_2 \\ a'_{32}x_2 + \cdots + a'_{3n}x_n &= b'_3 \\ &\vdots \\ a'_{n2}x_2 + \cdots + a'_{nn}x_n &= b'_n \end{aligned}$$

Algorithm

(3) In the similar fasion eliminate x_2 from the third to the last equation in the new set this process will continue untill the last equation contains only one unknown,namely x_n. The final form of the equations will look like this

$$\begin{aligned} a_{11}x_1 + a_{21}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a'_{22}x_2 + \cdots + a'_{2n}x_n &= b'_2 \\ &\vdots \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)} \end{aligned}$$

This process is called *triangularisation*.The number of primes indicate the number of times the coefficient has been modified.

(4) Obtain the solution by back substitution.

$$\boxed{x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}}$$

This can be substituted back in the (n-1)th equation to obtain the solution for x_{n-1}. This back substitution can be continued till we get the solution for x_1

General form

The co-efficient of the k th derived system has the general form

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)}$$

where

$$i = k+1 \quad \text{to} \quad n$$
$$j = k+1 \quad \text{to} \quad n$$

$a_{ij}^{(0)} = a_{ij}$ for $i = 1$ to n, $j = 1$ to n

The k th equation, which is multiplied by the factor a_{ik}/a_{kk}, is called the *pivot equation* and a_{kk} is called the pivot element. The process of dividing the k th equation by a_{ik}/a_{kk} is referred to as *normalisation* .

The k th unknown x_k has the general form

$$x_k = \frac{1}{a_{kk}^{(k-1)}} \left[b_k^{(k-1)} - \sum_{j=k+1}^{n} a_{kj}^{(k-1)} x_j \right]$$

where $k = n-1$ to 1 and

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}$$

Gauss Elimination With Pivoting

In gauss Elimination method each row is normalised by dividing the coefficients of that row by its pivot element.

$$a_{kj} = \frac{a_{kj}}{a_{kk}} \quad \text{where} \quad j = 1, \cdots, n$$

if $a_{kk} = 0$,k th row cannot be normalised.
One way to overcome this problem is to interchange this row with another row below it which doesnot have a zero element in that position.
It is suggested that the row with zero pivot element should be interchanged with the row having the largest coefficient in that position.This process is referred as *partial pivoting*.

In *complete pivoting* at each stage the largest element in any of the remaining rows is used as pivot.

Question

Solve the following system of equations using partial pivoting technique

$$\begin{aligned} 2x_1 + 2x_2 + x_3 &= 6 \\ 4x_1 + 2x_2 + 3x_3 &= 4 \\ x_1 - x_2 + x_3 &= 0 \end{aligned}$$

Gauss-Jordan Method

Gauss-Jordan method like Gauss elimination method uses the process of elimination of variables, but it is eliminated form all other rows (both below and above).This process thus eliminates all the off-diagonal terms producing a diagonal matrix rather than a triangular matrix.Also all the rows are normalised by dividing them by their pivot elements. Consequently,we can obtain the values of the unknowns directly from the b vector,without employing back-substitution.

Question

Solve the following system of equations using Gauss-Jordan technique

$$\begin{aligned} 2x_1 + 4x_2 - 6x_3 &= -8 \\ x_1 + 3x_2 + x_3 &= 10 \\ 2x_1 - 4x_2 - 2x_3 &= -12 \end{aligned}$$

Matrix Inverse Method

By Matrix algebra, the system of linear equation can be writen as

$$\mathbf{Ax} = \mathbf{b}$$

multiply each side of equation by the inverse of $\mathbf{A}$, we have

$$\mathbf{A^{-1}Ax} = \mathbf{A^{-1}b}$$

Since $\mathbf{A^{-1}A} = \mathbf{I}$, the identy matrix, above equation becomes

$$\boxed{\mathbf{x} = \mathbf{A^{-1}b}}$$

Question

Solve the following system of equations using Matrix-Inversion technique

$$\begin{aligned} 4x_1 + 3x_2 - x_3 &= -6 \\ x_1 + x_2 + x_3 &= 10 \\ 3x_1 + 5x_2 + 3x_3 &= -12 \end{aligned}$$

Presentation Outline

Introduction

If we have a table of discreate data points $(x_i, y_i), i = 0, 1, 2, \cdots, n$,we are often interested to find the value of dependent variable y for an intermediate data points.This task can be accomplished by constructing a function y(x) that will pass through the given sets of points and then evaluating y(x) for the specified value of x. The process of construction of y(x) to fit table of data points is called *Curve fitting*.A table of data are of two types

- Table of values of well defined function
- Data tabulated from measurements made during experiment.

In the first case, the function is constructed such that it passes through all data points. This process of contructing a function and estimting values of non-tabular points is called *interpolation*.The function is called interpolation polynomial.

In the second case, the values are not accurate and therefore, it will be meaningless to pass the curve through every point. The best strategy would be to construct a single curve that would represent the general trends of the data, without necessarily passing through the individual points.Such function are called approximating functions.One such approximation function to fit a given set of experimental data is called *least-square regression*.The approximating polynomial are known as least square polynomial.

Polynomial

The most common from of nth order polynomial is

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

This form,known as power form ,is very convinient for differentating and intregating the polynomial function and therefore are most widely used in mathematical analysis.
Polynomial obtained from power form may not always produce accurate results. So we have shifted power form as shown below

$$p(x) = a_0 + a_1(x - c) + a_2(x - c)^2 + \cdots + a_n(x - c)^n \tag{1}$$

Where c is a point somewhere in the interval of interest, this form increase the accuracy.
Equation(1) is the Taylor expansion of p(x) around the point c , when the coefficients a_i are replaced by appropiate function derivatives.
There is a third form known as Newton form as shown below

$$p(x) = a_0 + a_1(x - c_1) + a_2(x - c_1)(x - c_2) + \cdots + a_n(x - c_1)(x - c_2)\cdots(x - c_n)$$

Polynomial can also be expressed in the form

$$\boxed{p_2(x) = b_0(x - x_1)(x - x_2) + b_1(x - x_0)(x - x_2) + b_2(x - x_0)(x - x_1)}$$

In general form,

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

Lagrange interpolation Polynomial

The points $(x_0, f_0), (x_1, f_1), \cdots, (x_n, f_n)$ can be imagined to be data values connected by curve. Any function $p(x)$ satisfying the conditions

$$p(x_k) = f_k \quad \text{for} \quad k = 0, 1, \cdots, n$$

is called interpolation function.
Let us consider a second-order polynomial of the form

$$p_2(x) = b_1(x - x_0)(x - x_1) + b_2(x - x_1)(x - x_2) + b_3(x - x_2)(x - x_0)$$

If $(x_0, f_0), (x_1, f_1), (x_2, f_2)$ are the three interpolating points, then we have

$$\begin{aligned} p_2(x_0) &= f_0 = b_2(x_0 - x_1)(x_0 - x_2) \\ p_2(x_1) &= f_1 = b_3(x_1 - x_2)(x_1 - x_0) \\ p_2(x_2) &= f_2 = b_1(x_2 - x_0)(x_2 - x_1) \end{aligned}$$

substituting for b_1, b_2 and b_3 in the above equation we get

$$p_2(x) = f_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f_1 \frac{(x - x_2)(x - x_0)}{(x_1 - x_2)(x_1 - x_0)} + f_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Above equation can be represented as

$$\begin{aligned} p_2(x) =& f_0 l_0(x) + f_1 l_1(x) + f_2 l_2(x) \\ =& \sum_{i=0}^{2} f_i l_i(x) \end{aligned}$$

Lagrange interpolation Polynomial

where

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

In general, for $n+1$ points we have nth degree polynomial as

$$\boxed{p_n(x) = \sum_{i=0}^{n} f_i l_i(x)} \tag{A}$$

where

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

Eqn (A) is called the *Lagrange interpolation polynomial*. The polynomial $l_i(x)$ are known as *Lagrange basis polynomials*

Question

The table gives below square roots for integers

x	1	2	3	4	5
f(x)	1	1.4142	1.7321	2	2.2361

Find the square root of 2.5 using second order Langrage interpolation polynomial.

Also write fortran code for it.

Newton Interpolation polynomial

The Newton form of polynomial is

$$p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \quad (1)$$

Where the interpolation points $x_0, x_1, \cdots, x_{n-1}$ act as centres.
Let us assume that (x_0,f_0),(x_1,f_1),...,(x_{n-1},f_{n-1}) are the interpolation points. i.e

$$p_n(x_k) = f_k \quad \text{for} \quad k = 0, 1, \cdots, n$$

now at $x = x_0$,

$$p_n(x_0) = \boxed{a_0 = f_0}$$

now at $x = x_1$,

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

substituting a_0 we get

$$\boxed{a_1 = \frac{f_1 - f_0}{x_1 - x_0}}$$

at $x = x_2$,

$$p_n(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = f_2$$

Substituting a_0 and a_1 and rearranging we get,

$$\boxed{a_2 = \frac{[(f_2 - f_1)/(x_2 - x_1)] - [(f_1 - f_0)/(x_1 - x_0)]}{x_2 - x_0}}$$

Newton Interpolation polynomial

We can write in terms of *divide differences*

$$
\begin{aligned}
a_0 &= f_0 = f[x_0] \\
a_1 &= \frac{f_1 - f_0}{x_1 - x_0} = f[x_0, x_1] \\
a_2 &= \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} \\
&= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\
&= f[x_0, x_1, x_2]
\end{aligned}
$$

Thus

$$a_n = f[x_0, x_1, x_2, \cdots, x_n]$$

a_1 represent *first divided difference* and a_2 the *the second divided difference* and so on
Putting the coefficient of a_i coefficients in the equation (1) we get

$$
\begin{aligned}
p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\
+ f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1})
\end{aligned}
$$

Newton Interpolation polynomial

This can be written in compact form

$$p_n = \sum_{i=1}^{n} f[x_0, x_1, \cdots, x_i] \prod_{j=0}^{i-1} (x - x_j) \quad (2)$$

Equation (2) is called *Newton's divided difference interpolation polynomial.*

Question

Given below is a table for log x. Estimate log 2.5 using second order Newton interpolation polynomial.

i	0	1	2	3
x_i	1	2	3	4
log x_i	0	0.3010	0.4771	0.6021

Newton Gregory forward difference formula

We have Newton form of polynomial as

$$p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

We can write in terms of *divide differences* as

$$\begin{aligned} a_0 &= f_0 = f[x_0] \\ a_1 &= \frac{f_1 - f_0}{x_1 - x_0} = f[x_0, x_1] \\ a_2 &= \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} \\ &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ &= f[x_0, x_1, x_2] \end{aligned}$$

Thus

$$a_n = f[x_0, x_1, x_2, \cdots, x_n]$$

putting the above coefficients in the main equation

$$\begin{aligned} p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ + f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned}$$

Newton Gregory forward difference formula

Consider the case where the function valued are given for equidistant points

$$x_k = x_0 + kh$$

We can write

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

$$f[x_0, x_1, x_2] = \frac{\Delta f_1 - \Delta f_0}{x_2 - x_0} = \frac{\Delta^2 f_0}{2h^2} = \frac{\Delta^2 f_0}{2!\, h^2}$$

$$\vdots$$

$$f[x_0, x_1, x_2, \cdots, x_n] = \frac{\Delta^n f_0}{n!\, h^n}$$

Then the equation

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

can be writen as

$$p_n(x) = \sum_{j=0}^{n} \frac{\Delta^j f_0}{j!\, h^j} \prod_{k=0}^{j-1} (x - x_k)$$

Newton Gregory forward difference formula

Let us set $x = x_0 + sh$ so that $p_n(s) = p_n(x)$ and also $(x - x_k) = (s - k)h$
Substituting we get,

$$p_n(s) = \sum_{j=0}^{n} \frac{\Delta^j f_0}{j!\, h^j} \prod_{k=0}^{j-1} (s - k)h$$

$$p_n(s) = \sum_{j=0}^{n} \frac{\Delta^j f_0}{j!\, h^j} [s(s-1)(s-2)\cdots(s-j+1)]h^j$$

Thus

$$\boxed{p_n(s) = \sum_{j=0}^{n} \binom{s}{j} \Delta^j f_0}$$

The above equation is Newton Gregory forward difference formula and can be expanded as follows

$$p_n(s) = f_0 + \Delta f_0 s + \frac{\Delta^2 f_0}{2!} s(s-1) + \frac{\Delta^3 f_0}{3!} s(s-1)(s-2) + \cdots$$

Forward Difference Table

x	f	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$	$\Delta^5 f$
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					

Question

Estimate the value of $\sin\theta$ at $\theta = 25°$ using the following 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

Presentation Outline

Curve fitting :Regression

Regression analysis is a methods of curve fitting of experimental data.*Least square regression* is used when

- Relationship is linear
- relationship is transcendental
- Relationship is polynomial
- Relationship involves two or more independent variables.

Curve fitting :Fitting a Linear equation

Supppose to describe a experimenta data we are using Mathematical equation for a straight line ,which is

$$y = a + bx = f(x)$$

Where a is the intercept and b is the slope of the line. If (x_i, y_i) represent the set of data and q_i represent the error of data points. Then

$$\begin{aligned} q_i &= y_i - f(x_i) \\ &= y_i - a - bx_i \end{aligned}$$

Sum of the squares of individual errors can be expressed as

$$Q = \sum_{i=1}^{n} q_i^2 = \sum_{i=1}^{n} [y_i - f(x_i)]^2$$

Curve fitting :Fitting a Linear equation

$$= \sum_{i=1}^{n} [y_i - a - bx_i)]^2$$

We choose the values of a and b such that Q is minimised.So the necessary condition is

$$\frac{\partial Q}{\partial a} = 0 \quad \text{and} \quad \frac{\partial Q}{\partial b} = 0$$

Then

$$\frac{\partial Q}{\partial a} = -2 \sum_{i=1}^{n} [y_i - a - bx_i)] = 0$$

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

Thus

$$\sum y_i = na + b \sum x_i$$

$$\sum x_i y_i = a \sum x_i + b \sum x_i^2$$

These are called normal equations. Solving for a and b , we get

Curve fitting :Fitting a Linear equation

$$b = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

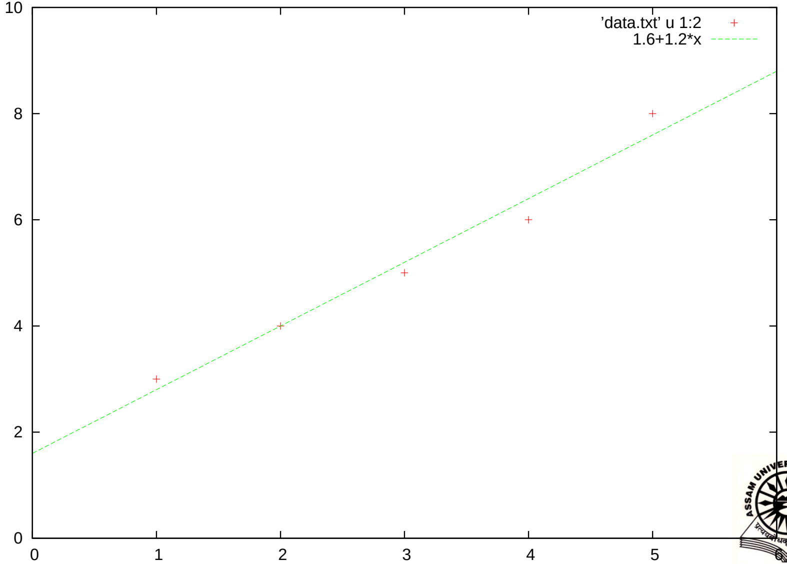
$$a = \frac{\sum y_i}{n} - b\frac{\sum x_i}{n} = \bar{y} - b\bar{x}$$

Where $\bar{x}$ and $\bar{y}$ are the averages of x values and y values respectively.

Question

Fit a straight line to the following set of data

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


10
8
6
4
2
0
0
1
2
3
4
5
6
'data.txt' u 1:2
1.6+1.2*x

Presentation Outline

Numerical Intregation

A definite intregal of the form

$$I = \int_a^b f(x)dx$$

can be treated as area under the curve $y = f(x)$,enclosed between the limits $x = a$ and $x = b$. The problem of intregation is then simply reduced to the problem of finding the shaded region.

Numerical intregation methods uses an interpolating polynomial $p_n(x)$ in place of $f(x)$.Thus

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx \tag{1}$$

polynomial $p_n(x)$ can be easily intregated analytically.

equation(1) can be expressed as

$$\int_a^b p_n(x)dx = \sum_{i=0}^{n} w_i p_n(x_i)$$

where $a = x_0 < x_1 < x_2 \cdots < x_n = b$

Since $p_n(x)$ coincides with $f(x)$ at the points $x_i, i = 0, 1, 2, \cdots, n$ we can say

$$\boxed{I = \int_a^b f(x)dx \approx \sum_{i=0}^{n} w_i p_n(x_i)}$$

Numerical Intregation

There is a set of methods known as Newton-Cotes rules in which the sampaling points are equally spaced.
For approximating the function $f(x)$ Newton or Langrage interpolation polynomial is used. We use Newton-Gregory formula which is given below:

$$\begin{aligned} p_n(s) &= f_0 + \Delta f_0 s + \frac{\Delta^2 f_0}{2!}s(s-1) + \frac{\Delta^3 f_0}{3!}s(s-1)(s-2) + \cdots \\ &= T_0 + T_1 + T_2 + \cdots \end{aligned} \tag{1}$$

where x_0 is called reference point given by

$$s = (x - x_0)/h$$

and h is called step size given by

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

Tapezoidal Rule

Trapezoidal rule is a two-point formula, it uses the first order interpolation polynomial $p_1(x)$ for approximating the function $f(x)$ and assumes $x_0 = a$ and $x_1 = b$.

According to equation (1) $p_1(x)$ consist of first two terms T_0 and T_1. Therefore the integeral for trapezoidal rule is given by

Tapezoidal Rule

$$I_t = \int_a^b (T_0 + T_1) dx$$
$$= \int_a^b T_0 \; dx + \int_a^b T_1 \; dx$$

T_i has to expressed in terms of s
$dx = h \times ds \quad x_0 = a \; x_1 = b$ and $h = b - a$
At $x = a \quad s = 0 \quad$ and at $x = b \quad s = 1$
Hence

$$I_{t1} = \int_a^b T_0 \; dx = \int_0^1 hf_0 \; dx = hf_0$$
$$I_{t2} = \int_a^b T_1 \; dx = \int_0^1 \Delta f_0 sh \; ds = h\frac{\Delta f_0}{2}$$

Therefore

$$I_t = h\left[f_0 + \frac{\Delta f_0}{2}\right] = h\left[\frac{f_0 + f_1}{2}\right]$$

Since $f_0 = f(a)$ and $f_1 = f(b)$ we have

$$\boxed{I_t = h\frac{f(a) + f(b)}{2} = (b - a)\frac{f(a) + f(b)}{2}}$$

Area=width of the segment $(b - a) \times$ average height of the points $f(a) and f(b)$

Question

Evaluate the integral

$$I = \int_a^b (x^3 + 1)\, dx$$

for the intervals

(a) (1,2)

(b) (1,1.5)

Simpson's 1/3 rule

In this rule the function $f(x)$ is approximated by a second order polynomial $p_2(x)$ which passes through three sampaling points given by $x_0 = a$, $x_2 = b$ and $x_1 = (a+b)/2$. The width of the segments h is given by

$$h = \frac{b-a}{2}$$

The integral for the simpson's 1/3 rule is obtained by intregating the first three terms of equation (1) i.e

$$\begin{aligned} I_{s1} &= \int_a^b p_2(x)\ dx = \int_a^b (T_0 + T_1 + T_2)\ dx \\ &= \int_a^b T_0\ dx + \int_a^b T_1\ dx + \int_a^b T_2\ dx \\ &= I_{s11} + I_{s12} + I_{13} \end{aligned}$$

where

$$\begin{aligned} I_{s11} &= \int_a^b f_0\ dx \\ I_{s12} &= \int_a^b \Delta f_0 s\ dx \\ I_{s13} &= \int_a^b \frac{\Delta^2 f_0}{2} s(s-1)\ dx \end{aligned}$$

Simpson's 1/3 rule

Where $dx = h \times ds$ and s varies from 0 to 2 (as x varies from a to b).Thus,

$$I_{s11} = \int_0^2 f_0 h \; ds = 2hf_0$$

$$I_{s12} = \int_0^2 \Delta f_0 sh \; ds = 2h\Delta f_0$$

$$I_{s13} = \int_0^2 \frac{\Delta^2 f_0}{2} s(s-1)h \; ds = \frac{h}{3}\Delta^2 f_0$$

Therefore,

$$I_{s1} = h\left[sf_0 + 2\Delta f_0 + \frac{\Delta^2 f_0}{3}\right]$$

Since $\Delta f_0 = f_1 - f_0$ and $\Delta^2 f_0 = f_2 - 2f_1 + f_0$ above equation reduces to

$$\boxed{I_{s1} = \frac{h}{3}[f_0 + 4f_1 + f_2] = \frac{h}{3}[f(a) + 4f(x_1) + f(b)]}$$

This equation is called *Simson's 1/3 rule.*

Composite Simpson's 1/3 rule

Usually Simpson's 1/3 rule is employed by dividing the interval into n number of segments of equal width.Then the step size is

$$h = \frac{b-a}{n}$$

where $x_i = a + ih, i = 0, 1, 2, \cdots, n$. now to each $n/2$ pairs or subintervals i.e $(x_{2i-2}, x_{2i-1}), (x_{2i-1} - x_{2i})$ Simpson's 1/3 rule is applied which gives

$$\begin{aligned} I_{cs1} &= \frac{h}{3}\sum_{i=1}^{n/2}[f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})] \\ &= \frac{h}{3}[f(a) + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{n-2} + 4f_{n-1} + f(b)] \end{aligned}$$

On regrouping terms ,we get

$$\boxed{I_{cs1} = \frac{h}{3}\left[f(a) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i}) + f(b)\right]}$$

Question

Evaluate the integral

$$I = \int_0^{\pi/2} \sqrt{sin(x)}\, dx$$

applying Simpson's 1/3 rule for $n = 4$ and $n = 6$ with an acccuracy to five decimal places. Also write a fortan code for it.

FORTRAN code for Simpson's 1/3 rule

Write a FORTRAN code to evaluate the integral

$$I = \int_{-1}^{1} e^x \, dx$$

applying Simpson's 1/3 rule for $n = 4$ and $n = 6$ with an acccuracy to five decimal places.

FORTRAN code for Simpson's 1/3 rule

Write a FORTRAN code to evaluate the integral

$$I = \int_{-1}^{1} e^x \, dx$$

applying Simpson's 1/3 rule for $n = 4$ and $n = 6$ with an acccuracy to five decimal places.

Answer

We have from Simpson's 1/3 rule the value of integeral is

$$I = \frac{h}{3}\left[f_0 + 4f_1 + f_2\right] = \frac{h}{3}\left[f(a) + 4f(x_1) + f(b)\right]$$

where $h = (b - a)/2$, and f_1 is the function value at $x = (a + b)/2$
Now to increase the accuracy the intervals are split up into even no. of intervals and to each interval Simpson's 1/3 rule is applied, the method is known as composite Simpson's 1/3 rule.
For n=2, we have

$$I = \frac{h}{3}\left[f_0 + 4f_1 + f_2 + f_2 + 4f_3 + f_4\right]$$

$$I = \frac{h}{3}\left[f_0 + 4f_1 + +2f_2 + 4f_3 + f_4\right]$$

where $h = (b - a)/4$

Answer

For n=4, we have

$$I = \frac{h}{3}\left[f_0 + 4f_1 + f_2 + f_2 + 4f_3 + f_4 + f_4 + 4f_5 + f_6 + f_6 + 4f_7 + f_8 +\right]$$

$$I = \frac{h}{3}\left[f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + 4f_5 + 2f_6 + 4f_7 + f_8\right]$$

where $h = (b - a)/8$

generalising for n interval we have $h = (b - a)/(2 * n)$ we can now go for writing code

Presentation Outline

Introduction

Mathematical models which uses differential equation to express relationship between variables are known as *differential equations*.
Examples are:

1. **Law of motion**

$$m\frac{dv(t)}{dt} = F$$

2. **Kirchoff's law for an electrical circuit**

$$L\frac{di}{dt} + iR = V$$

3. **Simple Harmonic motion**

$$m\frac{d^2y}{dt^2} + a\frac{dy}{dt} + ky = 0$$

The initial-value problem of an ordinary differential equation has the form

$$\boxed{y'(x) = f(x, y(x)), y(x_0) = y_0}$$

Euler's Method

We can expand a function $y(x)$ about a point $x = x_0$ using Taylor's Theorem of expansion

$$y(x) = y(x_0) + (x - x_0)y'(x_0) + (x - x_0)^2 \frac{y''(x_0)}{2!} + \cdots + (x - x_0)^n \frac{y^n(x_0)}{n!}$$

We have the differential equation of form

$$y' = f(x, y) \quad \text{with} \quad y(x_0) = y_0$$

then we have

$$y'(x_0) = f(x_0, y_0)$$

or we have

$$y(x) = y(x_0) + (x - x_0)f(x_0, y_0)$$

At $x = x_1$ we have

$$y(x_1) = y(x_0) + (x_1 - x_0)f(x_0, y_0)$$

If $h = (x_1 - x_0)$ is the step-size, the above equation becomes

$$y(x_1) = y(x_0) + hf(x_0, y_0)$$

Similarly

$$y(x_2) = y(x_1) + hf(x_1, y_1)$$

Generalising

$$\boxed{y(x_{i+1}) = y(x_i) + hf(x_i, y_i)}$$

Question

Estimate the value of $y(2)$ using Euler's method

$$\frac{dy}{dx} = 3x^2 + 1 \quad \text{with} \quad y(1) = 2$$

for (i) $h = 0.5$ (ii) $h = 0.25$

Runge-Kutta method

Runge-Kutta method is based on the general form of extrapolation equation

$$y_{i+1} = y_i + \text{slope} \times \text{interval size}$$

where m is the weighted averages of the slopes at the various points in the interval h.
m can be writen as

$$m = w_1 m_1 + w_1 m_1 + w_2 m_2 + \cdots w_r m_r$$

where $w_1, w_2, \cdots, w_r$ are weights of the slopes at various points. The slopes $m_1, m_2, \cdots, m_r$ are computed as follows

$$\begin{aligned}
m_1 &= f(x_i, y_i) \\
m_2 &= f(x_i + a_1 h, y_i + b_{11} m_1 h) \\
m_3 &= f(x_i + a_2 h, y_i + b_{21} m_1 h + b_{22} m_2 h) \\
&\vdots \\
m_r &= f(x_i + a_{r-1} h, y_i + b_{r-1,1} m_1 h + \cdots + b_{r-1,r-1} m_{r-1} h)
\end{aligned}$$

Fourth-order Runge-Kutta method

Runge-Kutta method is based on the general form of extrapolation equation

$$y_{i+1} = y_i + \text{slope} \times \text{interval size}$$

slope can be calculated as

$$\boxed{\begin{aligned} m_1 &= f(x_i, y_i) \\ m_2 &= f\left(x_i + \frac{h}{2}, y_i + \frac{m_1 h}{2}\right) \\ m_3 &= f\left(x_i + \frac{h}{2}, y_i + \frac{m_2 h}{2}\right) \\ m_4 &= f\left(x_i + h, y_i + m_3 h\right) \\ y_{i+1} &= y_i + \left(\frac{m_1 + 2m_2 + 2m_3 + m_4}{6}\right) h \end{aligned}}$$

Question

Using 4th order RK method solve the differential equation to obtain the value $y(0.4)$

$$\frac{dy}{dx} = x^2 + y^2 \quad \text{with} \quad y(0) = 0$$

for (i) $h = 0.2$ (ii) $h = 0.1$

Presentation Outline

References

- *Fortran 77 and Numerical Methods*:**C Xavier**,New Age International(p) Limited
- *Numerical Methods*:**E Balaguruswamy**,Tata McGraw-Hill Publishing Company limited,New Delhi
- *Computer Programming in Fortran 77*:**V Rajaraman**,PHI,New Delhi
- *Fortran for Beginners*:**E Balaguruswamy**,Tata McGraw-Hill Publishing Company limited,New Delhi