Review Paper on Numerical Method and Necessity of Computers for a High Speed Calculation

Vishal Vaman Mehtre[1], Akash Sudhakar Bhavsar[2]

[1]Assistant Professor, Department of Electrical Engineering, Bharati Vidhyapeeth Deemed (to be) University College of Engineering Pune India

[2]Student, Department of Electrical Engineering, Bharati vidhyapeeth Deemed (to be) University College of Engineering Pune India

***Abstract*- This paper focuses on efficient high-order methods suitable for practical applications, with particular emphasis on new methods, new skills. There are many basic principles of solving numerical methods. E.g. The necessity of computers for high speed calculations: One of the most difficult and most important jobs performed by computers is the solution of complicated problems involving numbers. Computers can solve those problems amazingly and quickly. The computer can perform a simple numerical problem to complicated numerical problem. Computation saves a lot of time and energy.**

***Index terms*- Numerical methods, high speed computers**

INTRODUCTION

To obtain roots of given equation

Solve f(x) = 0 for x, when an explicit analytical solution is impossible. Following method is used for root finding.

Bisection Method:-

The bisection method is the easiest to numerically implement and almost always works. The main disadvantage is that convergence is slow. If the bisection method results in a computer program that runs too slow, then other faster methods may be chosen; otherwise it is a good choice of method.

We want to construct a sequence x0, x1, x2, .that converges to the root x = r that solves f(x) = 0. We choose x0 and x1 such that x0 < r < x1. We say that x0 and x1 bracket the root. With f(r) = 0, we want f(x0) and f(x1) to be of opposite sign, so that f(x0)f(x1) < 0. We then assign x_2 to be the midpoint of x0 and x1, that is x2 = (x0 + x1)/2, or

$$x_2 = x_0 + \frac{x_1 - x_0}{2}$$

The sign of f(x2) can then be determined. The value of x3 is then chosen as either the midpoint of x0 and x2 or as the midpoint of x2 and x1, depending on whether x0 and x2 bracket the root, or x2 and x1 bracket the root. The root, therefore, stays bracketed at all times. The algorithm proceeds in this fashion and is typically stopped when the increment to the left side of the bracket (above, given by (x1 − x0)/2) is smaller than some required precision. [1]

Numerical

Question: _ $f(x) = x^3-1.8x^2-10x+17$

Answer: _

Iteration no. 1

$f(a) = f(1) = (1)^3-1.8(1)^2-10x+17$

$= 6.2$

$f(b) = f(2) = (2)^3-1.8(2)^2-10x+17$

$= -2.2$

C = a+b/2 = 1+2/2 = (1.5)

$f(c) = f(1.5) = (1.5)^3-1.8(1.5)^2-10(1.5)+17$

$=1.325$

Here $f(2)\,f(5) <0$

or $f(b)\,f(c) <0$

Hence the root lies between b & c. therefore replace ‘a’ by ‘c’ in a new interval

:. New interval become, [a,b] = [1.5, 2]

Iteration No 2

We know form iteration no 1 [a=1.5, b=2]

C= a+b/2 = 1.5+2/2 =1.75

$f(c)=f(1.75) =(1.75)^3-1.8(1.75)^2-10(1.75)+17$

$=-0.653125$

Hear $f(1.5)\,f(1.75) <0$

$f(a)\,f(c) <0$

Hence root lies between a & c Therefore replace ‘b’ by ‘c’ in new interval

: .New interval become [a, b] = [1.5, 1.75]

Iteration No 3

Form iteration No. 2 we have a=1.5, b=1.75

C=a+b/2 =1.5+1.75/2 =1.6225

$f(c)= f(1.625)=(1.625)^3-1.8(1.625)^2-10(1.625)+17$

$=0.287891$

Hear $f(1.75)\ f(1.625) <0$

$f(b)\ f(c)<0$

Hence root lies between b & c Therefore replace 'a' &c' in new interval

: .New interval become [a, b] = [1.625, 1.75]

Iteration No 3

We have a=1.625, b= 1.75

C= a+b/2 =1.625+1.75/2 =1.6875

$f(c)=f(1.6875)=(1.6875)^3-1.8(1.6875)^2-10(1.6875)+17$

$=-0.195361$

Here $f(1.625)\ f(1.6875) <0$

or $f(a)\ f(b) <0$

Hence the root lies between b & c. therefore replace 'a' by 'c'

Here $f(1.625)\ f(1.6875) <0$

Or $f(a)\ f(c) <0$

Hence the root lies between b & c. therefore replace 'a' by 'c' in a new interval. New interval become,

[a, b] = [1.625, 1.6875]

Iteration no. 4

We have a=1.625, b=1.6875

C= a+b/2 =1.625+1.6875/2 =1.65625

$f(1.65625)=f(c)=(1.65625)^3-1.8(1.65625)^2-10(1.65625)+17$

$=0.04317$

Here $f(1.625)\ f(1.6875)<0$

$f(b)\ f(c) <0$

Hence root lies between b & c

Iteration No 5

Form iteration No 4 [a=1.625, b= 1.6875]

C= a+b/2 =1.625+1.6875/2 =1.65625

$f(c)=f(1.65625)=(1.65625)^3-1.8(1.65625)^2-10(1.65625)+17$

Hear $f(1.6875)\ f(1.65625) <0$

$f(b)\ f(c) <0$

Hence root lies between b & c.

Therefore, the root at the end of 5^{th} iteration approximate value of roots will be at the center of 'b' &'c'.

Roots =1.6875+1.65625/2 =1.671875

Thus approximate value of roots at the end of 5^{th} iteration= 1.671875

Newton's Method

This is the fastest method, but requires analytical computation of the derivative of f(x). Also, the method may not always converge to the desired root. We can derive Newton's Method graphically, or by a Taylor series. We again want to construct a sequence x0, x1, x2, . .that converges to the root x = r. Consider the xn+1 member of this sequence, and Taylor series expand f(xn+1) about the point xn. We have $f(xn+1) = f(xn) + (xn+1 - xn)f\ 0(xn) + . . .xn$ To determine xn+1, we drop the higher-order terms in the Taylor series, and assume f(xn+1) = 0. Solving for xn+1, we have [2]

$x_{n+1} = xn -f\ (xn)/f\ 0(xn)$

Numerical

Question $f(x)=x^3-5x+3$

Answer $f(0)=3>0$ $f(1)=-1<1$

roots lies between 0 & 1

$f(0.1) = >0.$

$f(0.2) = >0.$

$f(0.3) = >0.$

$f(0.6) = >0.$

$f(0.7) = <0.$

take x=0.6

$f(x)=x^3-5x+3$

$=0.216-3+3$

$=0.216$

$f'(x0)= 3x^2-5$

$=.3.92$

Iteration No 1

$x1 =x0-((f(x0)) / (f'(x0)))$

$=0.6-0.216/-3.92$

$= 0.6551$

$f(x)=x^3-5x+3$

$=.28142728-3.2755102+3$

$= 0.005632$

$f'(x1)=3x^2-5$

$= -3.712523952$

Iteration No 2

$x2 =x1-((f(x1))/(f'(x1)))$

$= 0.65510204+0.005632528/3.712523952$

$=0.656619209$

Secant Method

The Secant Method is second best to Newton's Method, and is used when a faster convergence than Bisection is desired, but it is too difficult or impossible to take an analytical derivative of the function f(x). We write in place of f 0(xn)

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

Numerical

Question: - $f(x) = x^3-5x+3$

Answer: - $f(0) = 3>0$

$f(1) = -1 <0$

Therefore roots lies between 0&1

Iteration No 1

$x2 = x1 - (\frac{x1-x0}{f(x1)-f(x0)})f(x1)$

x2 = 1-(1-0/-1-3)(-1)

= 0.75

Iteration NO 2

$f(x2) = -0.3281$

Therefore roots lies between 0&0.75

$x3 = x1 - (\frac{x2-x1}{f(x2)-f(x1)})f(x2)$

x3=0.75(0.75-0/-0.3281-3)(0.3281)

= 0.6760

SYSTEM OF EQUATIONS

Gaussian Elimination

The standard numerical algorithm to solve a system of linear equations is called Gaussian Elimination. We can illustrate this algorithm by example.

Consider the system of equations

$-3x + 2x2 - x3 = -1,$

$6x - 6x2 + 7x3 = -7,$

$3x - 4x2 + 4x3 = -6.$

To perform Gaussian elimination, we form an Augmented Matrix by combining the matrix A with the column vector b:

$$\begin{pmatrix} -3 & 2 & -1 & -1 \\ 6 & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{pmatrix}$$

Row reduction is then performed on this matrix. Allowed operations are (1) multiply any row by a constant, (2) add multiple of one row to another row, (3) interchange the order of any rows. The goal is to convert the original matrix into an upper-triangular matrix.

We start with the first row of the matrix and work our way down as follows. First we multiply the first row by 2 and add it to the second row, and add the first row to the third row:

$$.\begin{pmatrix} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & -2 & 3 & -7 \end{pmatrix}$$ [3]

LOWER UPPER DECOMPOSITION

We then go to the second row. We multiply this row by −1 and add it to the third row.

The resulting equations can be determined from the matrix and are given by

$-3x1 + 2x2 - x3 = -1$

$-2x2 + 5x3 = -9$ $-2x3 = 2.$

These equations can be solved by backward substitution, starting from the last equation and working backwards. We have

$-2x3 = 2 \rightarrow x3 = -1$

$-2x2 = -9 - 5x3 = -4 \rightarrow x2 = 2,$

$-3x1 = -1 - 2x2 + x3 = -6 \rightarrow x1 = 2.$

Therefore,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix}$$ [6]

Least-squares approximation

The method of least-squares is commonly used to fit a parameterized curve to experimental data. In general, the fitting curve is not expected to pass through the data points, making this problem substantially different from the one of interpolation.

We consider here only the simplest case of the same experimental error for all the data points. Let the data to be fitted be given by (xi, yi), with i = 1 to n.[4]

Interpolation

Consider the following problem: Given the values of a *known* function $y = f(x)$ at a sequence of ordered points $x_0, x_1. . . x_n$, find $f(x)$ for arbitrary x. When $x_0 \leq x \leq x_n$, the problem is called interpolation. When $x < x_0$ or $x > x_n$ the problem is called extrapolation.

With $y_i = f(x_i)$, the problem of interpolation is basically one of drawing a smooth curve through the known points $(x_0, y_0), (x_1, y_1), . . . , (x_n, y_n)$. This is not the same problem as drawing a smooth curve that approximates a set of data points that have experimental error. This latter problem is called least-squares approximation.

Here, we will consider three interpolation algorithms:
(1) Polynomial interpolation;
(2) Piecewise linear interpolation, and;
(3) Cubic spline interpolation.

Integration
We want to construct numerical algorithms that can perform definite integrals of the form Z b
I =f(x) dx.
Calculating these definite integrals numerically is called numerical integration, numerical quadrature, or more simply quadrature.

NECESSITY OF COMPUTERS FOR HIGH SPEED CALCULATIONS

The evolution of large electronic computers has been so rapid that the importance of these machines for the optical sciences is not widely recognized. Calculations have already been made with such equipment in the fields of lens design, color, and spectroscopy.
Enormous speeds, up to a hundred thousand times that of a desk calculator, and flexible logic of recent general-purpose computers now make possible numerical solution of partial differential equations, synthesis of simple lens systems, correlation of complex infrared spectra with known data, mechanized literature searching, and real-time control instrumentation.
One of the most difficult and most important jobs performed by computers is the solution of complicated problems involving numbers. Computers can solve those problems amazingly and quickly. The computer can perform a simple numerical problem to complicated numerical problem.
In today's ever-more digitalized world, we all have a tale or two to share about how personal computers have let us down: like how they refused to let us run different programs at the same time or how the data was so heavy that the damned device kept us on hold forever before conducting even the most trivial operation
Well, there is one machine in the world — and it's in Japan — that is absolutely free of such concerns, being the fastest computer on Earth and capable of handling a mind-boggling number of tasks in far less than the blink of an eye.
The K computer — jointly developed by the IT giant Fujitsu and housed at the RIKEN Advanced Institute for Computational Science (AICS) in Kobe — was ranked No. 1 in the TOP500 list of the world's fastest computers. The ranking is announced twice a year at the SC conference of supercomputing experts — also known as the International Conference for High Performance Computing, Networking, Storage, and Analysis [5]

CONCLUSION

In this review paper we learn about basic principles of numericals methods and necessity of computers for high speed calculations without any mistakes we can get correct decimal answer.
In this paper we learn about IEE arithmetic operations. For finding roots of equation we compare 3 methods (bisection method, newton's method, secant method). And we find that secant method is more appropriate to find actual answer.

ACKNOWLEDGMENT

We would like to express our special thanks of gratefulness to Dr D S Bankar, head of the department of electrical engineering for their able guidance and support for completing our research paper I would also like to thanks the faculty member of the department of electrical engineering who helped us with extended support.

REFERENCE

[1] Biswa Nath Datta , Lecture Notes on Numerical Solution of root Finding Problem (2012)
[2] P. Deaflhard, "Newton Methods for Nonlinear Problem and Adaptive Algorithms", Springer Series in Computational Mathematics, Vol. 35. Springer, Berlin, 2004
[3] Galdino, *Sergio* "A family of newton raphson root-finding methods". *11 July 2017*
[4] *Smithe, D. E.*, History of Mathematics, II, *Dover(2010)*
[5] Ehiwario, J.C., Aghamie, "Comparative study of Bisection method, Newton-Raphsons method", S.O, ISOR Journal of Engineering, ISSN (e): 2250-3021, Vol. 04, Issue 04, April 2014.
[6] J.S.Chitode "Computational techniques, technical publication (2001)