IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Comparative Study Of FORTRAN And MATLAB Programming For Newton- Raphson Method

Bhagvat K. Kumthekar[1] and Vijay B. Patare[2]

[1]*Department of Physics, Nutan Mahavidyalaya, Sailu Dist Parbhani-431503*

[2]*Department of Mathematics, Nutan Mahavidyalaya, Sailu Dist Parbhani-431503*

Abstract:

In the present paper, we would like to discuss about Newton-Raphson Method. This method is a powerful technique for finding the roots of given equations numerically. It is based on the simple idea of linear approximation using direct iteration method. Here, we have studied the roots polynomial equations using Programming of FORTRAN & MATLAB and compared their results. We found that this method is faster than Bisectional method as well as False position method.

Keywords: Newton-Raphson Method, iteration method, FORTRAN & MATLAB Programming

1. Introduction:

Newton-Raphson method is named after Isaac Newton and Joseph Raphson. It is a popular iterative method to find the roots of a polynomial equation. Newton's method was first published in 1685 in *A Treatise of Algebra both Historical and Practical* by John Wallis. In 1690, Joseph Raphson published a simplified description in *Analysis aequationum universalis*. Raphson observed that the Newton's method purely as an algebraic method and applicable to polynomials, but he describes the method in terms of the successive approximations x_n instead of the more complicated sequence of polynomials used by Newton. Finally, in 1740, Thomas Simpson described Newton's method as an iterative method for solving general nonlinear equations using calculus, essentially giving the description above. In the same publication, Simpson also gives the generalization to systems of two equations and notes that Newton's method can be used for solving optimization problems by setting the gradient to zero [1].

2. Newton-Raphson Method:

This method is based on the process of successive approximation by iteration. For the successive approximation, we start from a point $x = x_0$ which in the close vicinity of a root of the equation $f(x) = 0$. The new of x is obtained as the following.

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

Here, $f'(x_0)$ is differential of the function at $x = x_0$. We can repeat the process and can find out successive values of x, so that two successive values x_i and x_{i+1} are related through the following relation.

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

Each successive value of x becomes more and more close to the root of the equation. The process of iteration continues till the difference between two successive values is less than the required accuracy.

To find out a root in this method, one should obviously know the value x_0, which is close to the root of the equation. Since the convergence of the Newton-Raphson method is faster than that of the bisectional method as well as of the false position method, we would prefer to use it. It can be explained by solving one example.

Example: Find the root of the given equation $x^3-x-1 = 0$ for x

Letus consider $f(x) = x^3-x-1$ and $f'(x) = 3x^2-1$

$$x_{i+1} = x_i - \frac{x_i^3 - x_i - 1}{3x_i^2 - 1}$$

For this formulation we need to decide an appropriate initial x_i value. This value lies between positive and negative value of function $f(x)$ i.e. $f(1) = -1<0$ and $f(2) = 5 >0$. Therefore the must be lies between 1 and 2 . So that take initial value x_0 as a 1. Then

$$x_1 = x_0 - \frac{x_0^3 - x_0 - 1}{3x_0^2 - 1}$$

Now, we get $x_1 = 1.5$. Therefore, root lies between 1 and 1.5

$$x_2 = x_1 - \frac{x_1^3 - x_1 - 1}{3x_1^2 - 1}$$

Now, we get $x_2 = 1.34783$ and root lies between 1.5 and 1.34783. For next stage we have,

$$x_3 = x_2 - \frac{x_2^3 - x_2 - 1}{3x_2^2 - 1}$$

We find that $x3 = 1.32520$ and same procedure carried out next two values x_4 and x_5 we get 1.32472 as constant when the digit stops changing to the required accuracy. We can take that value as a root. In this example we have taken five decimal place accuracy hence the root of above equation is 1.32472.

MATLAB:

MATLAB stands for matrix laboratory. It is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by Math Works MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

MATLAB users come from various backgrounds of engineering, science, and economics. It is now also used in education, in particular the teaching of linear algebra, numerical analysis, and is popular amongst scientists involved in image processing

Code of Newton-Raphson Method in MATLAB

```
% Program Code of Newton-Raphson Method in MATLAB
a=input('Enter the function in the form of variable x:','s');
x(1)=input('Enter Initial Guess:');
error=input('Enter allowed Error:');
f=inline(a)
dif=diff(sym(a));
d=inline(dif);
for i=1:100
x(i+1)=x(i)-((f(x(i))/d(x(i))));
err(i)=abs((x(i+1)-x(i))/x(i));
if err(i)<error
break
end
end
root=x(i)
```

Fortran

- Fortran stands for "Formula Translation". It is a general-purpose, imperative programming language that is especially suited to numeric computation and scientific computing. Originally developed by IBM in the 1950s for scientific and engineering applications, Fortran came to dominate this area of programming. It has early versions Fortan I , II III, IBM 1401 FORTRAN and IV. The FORTRAN defined by the first standard, officially denoted X3.9-1966, became known as FORTRAN 66. After this a number of significant features are added in FORTRAN 66 and become a FORTRAN 77. Now a days Fortran 2008 is used for various purpose.

Code of NEWTON-RAPHSON METHOD in FORTRAN

```
DIMENSION  ROOT(100)
OPEN(UNIT = 2, FILE = $'$OUTPUT$'$)
AN =
AN1 =
AN2 =
H =
ACCU =
XMIN = $-$ SQRT((AN1/AN)**2 $-$ 2*AN2/AN)
XMAX = $-$ XMIN
M = 0
```

```
N = 0
XA = XMIN
10   A = F(XA)
20   XB = XA + H
B = F(XB)
IF( ABS(B) .LT. ACCU ) GO TO 50
IF((A*B) .LT. 0.0) GO TO 30
A = B
XA = XB
IF (XA  .GE. XMAX)  GO TO 70
GO TO 20
30    M = M + 1
IF(M  .EQ. 15) STOP
XI  = (XA + XB)/2.
C = F(XI)
IF ( ABS(C) .LT. ACCU ) GO TO 60
CA = C*A
IF (CA .LT. 0.0) B = C
IF (CA  .LT.  0.0) XB  = XI
IF (CA .GT. 0.0) A = C
IF (CA  .GT.  0.0) XA  = XI
IF (M .LT. 5) GO TO 30
40 X0  = XI
XI = X0 $-$ F(X0) / FP(X0)
IF ( ABS  (XI $-$ X0)  .LT. ACCU ) GO TO 60
GO TO 40
50    XI = XB
60    N = N + 1
XA = XI + H
ROOT(N) = XI
IF (XA .LT. XMAX) GO TO 10
70   WRITE(2, 80) (ROOT(I), I=1, N)
80   FORMAT(2X, $'$ROOTS ARE =  $'$, 6F8.2)
STOP
END
```

Comparison between FORTRAN & MATLAB

Fortran	MATLAB
FORTRAN Is a free software	The MATLAB programming language is part of the commercial MATLAB software
FORTRAN stands for “formula translation”. It is a language.	MATLAB Stands for “Matrix Laboratory”. It is like mathematical tool or An APP
In FORTRAN we have to write subroutine or function which is required for the execution of the program	MATLAB has already inbuilt subroutines to perform the mathematical operations; which could be called when required
Static error can be detected in Fortran	In Mat lab errors can be dynamically checked
Both are useful to solve mathematical problems	

DISCUSSION

- Both coding gives same results. FORTRAN used to solve only numeric problems.
- MATLAB allows its users to accurately solve problems. And Produce graphics easily.

MATLAB advantages

- MATLAB is a platform-independent interpreted language
- optimized for numerical (matrix and array) computation:
- It allows one to perform numerical calculations easily:
- Simple High Level Syntax
- It allows one to visualize the results without the need for
- complicated and time consuming programming:
- One or two line of MATLAB code in most simple cases
- Optimized for matrix and array structures—all our multimedia
- data structures are arrays or matrices
- Simple yet powerful MATLAB Integrated Development Environment (IDE)
- Rich support of multimedia formats
- Rich support of computational algorithms and tools

Acknowledgments:

I am heartily thankful to Prof. Suresh Chandra for his valuable guidance. Also thankful to Dr. Sharad S. Kulkarni for providing me necessary facilities for my research work and his kind cooperation.

References:

1. https://en.wikipedia.org/wiki/Newton's_method
2. *"FORTRAN". American Heritage Dictionary of the English Language (5 ed.). The Free Dictionary. 2011. Retrieved 2016-02-08.*
3. *John Backus. "The history of FORTRAN I, II and III" (PDF). Softwarepreservation.org. Retrieved 19 November 2014.*
4. *"Doctor Fortran in "One Door Closes"". Software.intel.com. Retrieved 21 September 2015.*
5. https://en.wikipedia.org/wiki/Fortran
6. *Cleve Moler (December 2009). "The Origins of MATLAB". Retrieved 15 April 2009.*
7. Suresh Chandra, Computer Applications in Physics, 2nd Edition, (2006) Narosa Publishing House, New Delhi