Graphical Approach to Solve the Transcendental Equations

Salim Akhtar[1] Ms. Manisha Dawra[2]
[1]M.Tech. Scholar [2]Assistant Professor
[1,2]Department of Computer Science & Engineering,
[1,2]Al-Falah School of Engineering & Technology, Faridabad, Haryana, India

Abstract— Transcendental equations are equations containing trigonometric, algebraic, exponential, logarithmic, etc. terms. Many analytical/iterative methods are used to solve transcendental equations.In this paper we present a graphical approach to solve the transcendental equations and find the approximate solution using Computer Based Numerical Techniques. We can easily find out the approximate solution of the transcendental equations.

Key words: CBNT, Transcendental Equation, Analytical Method, Genetic Algorithm.

I. INTRODUCTION

Transcendental equations are equations containing trigonometric, algebraic, exponential, logarithmic, etc. terms. Many analytical/iterative methods are used to solve transcendental equations. Though these methods are capable of solving many transcendental equations they suffer from many common disadvantages. Usually transcendental equations have many solutions in a given range, and analytical methods are not able to find all these roots in a given interval, even when they find several solutions, it is not possible to conclude that the given method has found the complete set of roots/solutions, and has not missed any particular solution. Also, these methods fail in case of misbehaved or discontinuous functions. Hence, though these methods may work very well in some situations, they are not general in nature and need a lot of homework from the Analyst.

An analysis of some common methods used for solving transcendental equations, their disadvantages and cases of failures are discussed below.

A. Newton Raphson Method:

This is a commonly used method for solving transcendental equations. The method makes use of the slope of the curve at different points. Therefore, if the function is non differentiable at points or has a point of inflexion, the method is not able to find the root. Secondly, if the function changes its slope very quickly (frequently achieves slope of zero), or is discontinuous, cannot be solved by this method. If the function is discrete, the derivative has no meaning for it and this method cannot be used. Also there is no straightforward way to find all the roots in an interval or even ascertain the number of roots in the interval.

B. Bisection Method:

This method needs two points on the graph such that $f(a)*f(b)<0$. There is no straightforward analytical method to find these points. Another problem lies in choosing the distance between the points ‘a’ and ‘b’. For the method to work, ‘a’ and ‘b’ should be close enough, such that the function behaves monotonously in these limits. At the same time, a small difference in values of ‘a’ and ‘b’ makes it difficult to search the sample space. An algorithm to ascertain such points for all roots of the equation has to be essentially random in nature and can be other applications of GA. This will be discussed later. Further still, the method fails for discontinuities in function.

C. Method of False Position:

This method suffers from same problems as Bisection method.Hence it can be concluded that analytical methods cannot find all the roots of a Transcendental equation reliably.In the field of graphics there are so many tools are available, the main aim of our project is to developed a geometrical solution tool using the concept of_co-ordinate geometry .Actually we are proving that we can switch from concept of co-ordinate geometry to the geometrical solutions.

D. Some examples of Transcendental equation:

$x = \sin(x)$
$x = e^{-x}$
$x = \cos(x)$
$e^{x} = x*\sin(x)$
$x*\cos(x) = 1$
$e^{x}*\sin(x)=1$
$x*\sin(x) – \cos(x) = 0$

II. INTRODUCTION TO GENETIC ALGORITHMS

Genetic Algorithms (GAs) were first presented by J. H. Holland in his book “Natural and Artificial Systems” in the year 1975 and developed further by his students. With time, many changes and improvements have been suggested. Here, we discuss the present form of implementation of Genetic Algorithms.

Genetic algorithms are a class of algorithms inspired by evolution. These algorithms encode solutions to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information.

An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. Then these structures are evaluated and allocated reproductive opportunities in such a way that those chromosomes which represent a better solution to the problem are given more chances to reproduce than those chromosomes which are poorer solutions. The "goodness" of a solution is typically defined with respect to the current population.

The basic steps involved in a GA are the following.

(1) Build an initial population of samples (solutions) created randomly or using some initialization method.
(2) Calculate the fitness (measure of being provided reproductive opportunities) of all the samples and select individuals for the reproduction process. The selection of the individual is though based on fitness, but it is a probabilistic mechanism. Roulette wheel selection, Rank Tournament Selection, Stochastic Universal Selection are some of the selections used.

(3) Apply the genetic operators of crossover, mutations, inversions, etc. to the selected individuals to create new individuals and thus a new generation. Crossover exchanges some of the bits of the two chromosomes and mutation inverts any bit(s) of the chromosome depending on a probability. Crossovers are a distinguishing feature of Genetic Algorithms. Many evolutionary algorithms were earlier used, but they basically worked on mutations and no crossovers took place. In GAs crossovers 'explore' around the already found good solutions and mutations help 'exploiting' the search space for new solutions.

Then again Step2 is followed till the condition for ending the algorithm is reached. Many issues such as encoding of problem, the types of selection operator, the type of fitness functions have been discussed and experimented at large by researchers to reach better results.

A. GAs for the present problem:

GAs has been traditionally used for optimization problems. Function optimization implemented by GAs was studied by De Jong in detail and GAs was successful to solve the problem. Solving transcendental equations is also a kind of optimization problem and hence GAs is applicable here.

Also, Melanie Mitchell states, if the space to be searched is large, is known not to be perfectly smooth and unimodal or is not well understood or if the fitness function is noisy and if the task doesn't require a global optimum to be found i.e. quickly finding a sufficient good solution is enough, GAs will have a good chance to be competitive with or other surpassing other weak methods. The present problem has a large search space, a fitness function which might be misbehaved and has more than one solution. All these difficulties indicate that genetic algorithms may be useful in such situations and I will address this problem in this paper.

III. PROBLEM IDENTIFICATION

When we try to solve a transcendental equation, it is very difficult to identify the initial guess or interval where root of transcendental equation lie. When we solve the transcendental equation by any numerical method, the initial guess is given. If the initial guess is not given then it is very difficult to solve the transcendental equation, so in this case first of all we try to find the initial guess, after more iteration we find the initial guess then solve transcendental using any numerical method.

A. Example:

Find the real root of the equation x log10 x = 1.2 by Bisection method
correct to four decimal places.
Sol. $f(x) = x \log10 x - 1.2$
Since $f(2.74) = - .000563$ *i.e.*, –ve
And $f(2.75) = .0081649$ *i.e.*, +ve
Hence, the root lies between **2.74** and **2.75**.
∴ First approximation to the root is
$x_1 = (2.74 + 2.75)/2 = 2.745$
Now $f(x_1) = f(2.745) = .003798$ *i.e.*, +ve
Hence, the root lies between 2.74 and 2.745.
∴ Second approximation to the root is
$x_2 = (2.74 + 2.745)/2 = 2.7425$
Now $f(x_2) = f(2.7425) = .001617$ *i.e.*, +ve
Hence, the root lies between 2.74 and 2.7425.
Note:
In the above example before first approximation of root, it is very difficult to find the initial values 2.74 and 2.75 there root lies, after more and more iteration we will be find these values.

IV. PROPOSED WORK

Actually we are developing a graph plotter and numerical equation solver but the problem is that if we want to represent the geometry system or the concept of coordinate system such as line, 2-D curves, points etc. Then which type of application you will use? So for solving this problem, we are developing the application, which will used to solve the problem such as root or solution/intersect point of the equation of the two curves or two lines.

So we will develop graphical tool by using the concepts of coordinate geometry, trigonometry, mathematical graphs etc. So the simple meaning is that we are implementing the solution of co-ordinate system and so this is a project by which you can generate the graphs of geometrical functions and also we can use this project for making and analyzing the mathematical models.

There are various types of graphical tool but for accuracy there is a need of entering the equation and apply any numerical methods to solve the transcendental equation. This type of graphical is using some different type of concept for developing graphical. However we are providing the just mouse click operations for user interface. There is also a need of a tool of plotting a 2-D curve in x-y plane.

In the field of animation we are showing that how we can use the geometry system for creating system. The new logic for different operation will surely used in the other type and high graphical tool. The other feature of the project on the basis of utility are given below
(1) Maximum resource utilization.
(2) Time management.
(3) Easy to analyze object.
(4) Easy to maintain.
(5)Minimum space required for developing application.

We will present a very simple geometric analysis of how to draw the Geometric Configuration. This Geometric Configuration contains a unique arrangement of the figure of the circle, the triangle, and the square, in such a harmonic way that deserves our special attention.

Our main objective is to develop a tool using concept of co-ordinate Geometry, which provide the solution of 2-D, 3-D Geometry and creation of Geometrical Graphical.

In equation you will enter two equations like the function, but in this case you will able to get the root of this equation also the display of that equation on the 2-D axis.
Solution of transcendental equations by graphical method
Example: (1) x=sin(x)

(2)|x|= ln(x) etc.

We are using to solve algebraic equations as well as transcendental equations by two methods
(1) Progressive Method.

(2) Regula Falsi Method.
(3) Newton-Raphson Method.

Apart from these two standard and well known methods we had also provided a simple method of solving mathematical equations that is Progressive Method.

A. Remark:

Progressive Method is generally used to check the result obtained by one of any method like Newton-Raphson method, Regula-Falsi method etc.

V. COMPUTER BASED UMERICAL METHODS AND ALGORITHMS

A. Bisection Method:

This method is based on the repeated application of intermediate value property.

Let the function F(x) be continuous between ‘a’ and ‘b’ for definiteness, let F(a)*F(b)<0. Then the first approximation to the root is x1 = 0.5(a+b).

1) Algorithm:

Step-1: Start the program
Step-2: Input the variables x1,x2 for the task
Step-3: Check F(x1)* F(x2)<0
Step-4: If yes proceed
Step-5: If no exist and print error message
Step-6: Repeat 7-11 if condition not satisfied
Step-7: X0=0.5*(x1+x2)
Step-8: If F(x0)*F(x1)<0
Step-9: x2=x0
Step-10: Else
Step-11: x1=x0
Step-12: Condition
Step-13: |(x1-x2)/x1|<maximum possible error or F(x0)=0
Step-14: Print output
Step-15: End of the program

B. Method of False Position or Regula-Falsi Method:

The bisection method guarantees that the iterative process will converge. It is however slow. Thus attempts have been made to speed up (order of convergence greater than one), bisection method retaining its guaranteed convergence. Method of doing this is called the method of false position.

It is sometimes known as method of linear interpolation.

This is the oldest method for finding the real roots of a numerical equation and closely resembles the bisection method.

In this method, we choose the two points x0 and x1 such that f(x0) and f(x1) are of opposite signs. Since the graph of y = f(x) crosses the x-axis between these two points, a root must lie in between these points.

1) Algorithm:

Step 1: Start of the program.
Step 2: Input the variables x0, x1, e, n for the task.
Step 3: f0=f(x0)
Step 4: f1=f(x1)
Step 5: for i=1 and repeat if i<=n
Step 6: x2=(x0f1-x1f0)/(f1-f0)
Step 7: f2=x2
Step 8: If|f2|<=e
Step-9: Print “convergent “, x2, f2
Step-10: If sign (f2) != sign(f0)
Step-11: x1=x2 and f1=f2
Step-12 else
Step-13: x0=x2 and f0=f2
Step-14: End loop
Step-15: Print output
Step-16: End of the program.

C. Algorithm of Newton-Raphson Method:

1) Method:

Step-1: Start of the program
Step-2: Input the variable x0, n for the task
Step-3: Input Epsilon and delta.
Step-4: For i=1 and repeat if i<=n.
Step-5: F0 = F(x0)
Step-6: dF0 = dF(x1)
Step-7: If |dF0| <= delta
Print slope too small
Print x0, F0, dF0, i
End pf the program
Step-8: x1= x0 – (F0/dF0)
Step-9: If |(x1-x0)/x1|<Epsilon
Print convergent
Print x1, F(x1), i
End of the program
Step-10: x0=x1
Step-11 End loop

VI. DESIGN AND IMPLEMENTATION OF MAIN PAGE

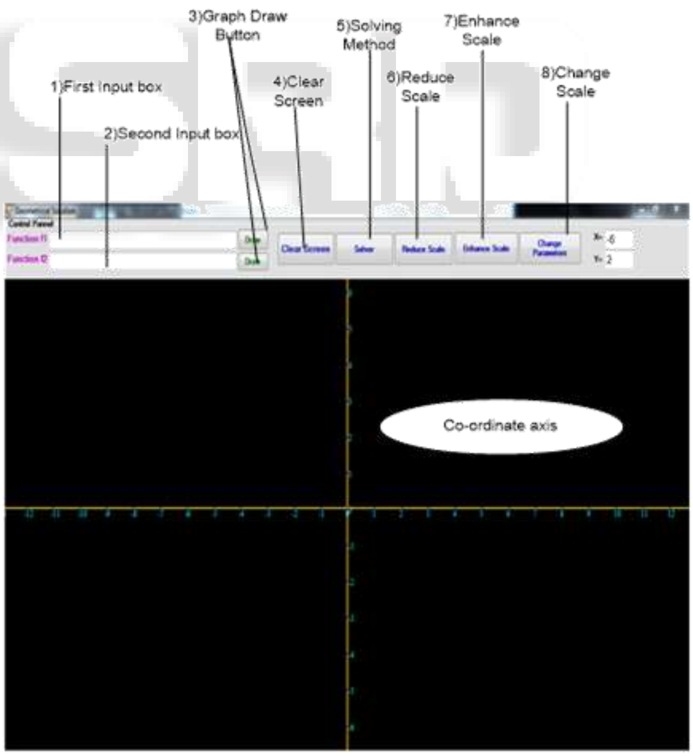

Fig. 1: Design & Implementation on main Page

VII. SOLVER ALGORITHMS PAGE

Fig. 2: Solve Algorithm page

VIII. CONCLUSIONS AND FUTURE WORK

A. Conclusions:

This thesis present the how to solve transcendental equation with the help of graphical approach. Transcendental equations are equations that containing trigonometric, algebraic, exponential, logarithmic, etc. terms. Usually transcendental equations have many solutions in a given range, and analytical methods are not able to find all these roots in a given interval, even when they find several solutions, it is not possible to conclude that the given method has found the complete set of roots/solutions, and has not missed any particular solution. This thesis presents only three numerical methods to solve the transcendental equations easily. Chapter-1 presents all three numerical methods and example of transcendental equation. Chapter-2 presents the analytical/iterative methods to solve transcendental equations and introduction to the Genetic Algorithm and also present the Genetic Algorithm for the present problem as well. Genetic algorithms are a class of algorithms inspired by evolution. These algorithms encode solutions to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information.

Chapter-3 presents the problem identification. When we solve the transcendental equation by any numerical method, the initial guess is given. If the initial guess is not given then it is very difficult to solve the transcendental equation, so in this case first of all we try to find the initial guess, after more iteration we find the initial guess then solve transcendental using any numerical method. Problem identification discussed with the help of example also.

Chapter-4 presents the proposed work. In the field of animation we are showing that how we can use the geometry system for creating system. The new logic for different operation will surely used in the other type and high graphical tool.

Chapter-5 presents the platform and language. For providing better user interface we used C# programming language. C# is a fully object oriented programming language in the field of software development.

Chapter-6 present computer based numerical methods and their algorithms. In this chapter we have also present Flow Chart of the numerical methods.

Chapter-7 presents designing of the tool. In this chapter we discussed 0, 1 and 2-level DFD, entity relationship diagram and also flow chart diagrams.

Chapter-8 presents implementation of the tool. First of all we have presented designing and implementation of main page, then solver algorithm page, then designing and implementation of progressive, Regula-Falsi and Newton-Raphson Methods.

Chapter-9 presents graph animation. We presented screen sorts of standard function graph, transformation of graph and composite graphs.

Chapter-10 presents solution of transcendental equation. In this chapter we applied all three methods to solve the transcendental equations with help of graphical approach, and find out the approximate solution.

B. Limitations and Future Work:

In this section we briefly discuss the limitations of our work and then draw some outlines of how future work can be carried out from this research.

There are some limitations in this thesis define as follows:

(1) This proposed system only 2-D not support 3-D.
(2) It has not drag and drop option.
(3) Only three numerical methods are implemented.
(4) Only two equations draw at a time on main page.

In the future work, we will extend the numerical methods and graph of the any equations as well, and we will propose 3-D Co-ordinate geometry system with a very good graphics, because this proposed system has low graphics so that we need to extend this system. In this proposed system only find the single approximate root of the transcendental equation so that we will extend to find out total number of solution and then choose the best solution of transcendental equation. This proposed system is a Windows application of .NET(Network Enable Technology). Dot NET framework was developed by Microsoft, so in future we will convert this system into Web application for many people who want to use these types of system online. We have only implemented three Computer Based Numerical Methods, so in future we will implement more numerical methods.

REFERENCES

[1] Manish Goyal, Computer Based Numerical & Statistical Techniques.
[2] Udit Agrwal, Computer Based Numerical & Statistical Techniques.
[3] Black Book of C#.
[4] De Jong, “An Analysis of the Behavior of a Class of Genetic Adaptive Systems”, 1975.
[5] David E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning” (text), 1989
[6] http://en.wikibooks.org/wiki/Numerical_Methods/Equation_Solving
[7] http://mat.iitm.ac.in/home/sryedida/public_html/caimna/transcendental/bracketing%20methods/bisection/bisection.html.
[8] http://www.wolfram.com/products/mathematica/newin7/content/TranscendentalRoots/
[9] http://www.shodor.org/unchem-old/math/newton/index.html

[10] http://mathematica.stackexchange.com/questions/22445/plotting-the-solutions-to-a-transcendental-equation
[11] http://www.wolfram.com/products/mathematica/newin7/content/TranscendentalRoots/
[12] http://www.webgraphing.com/transcendentaltricksoftrade.jsp
[13] http://www.ehow.com/how_8517694_solve-transcendental-equations-matlab.html
[14] http://en.wikibooks.org/wiki/Numerical_Methods/Equation_Solving
[15] www.math.odu.edu/~jhh/chs2.pdf
[16] http://reference.wolfram.com/mathematica/tutorial/SolvingEquations.html
[17] http://math.stackexchange.com/questions/412927/how-to-solve-transcendental-equations-with-matlab
[18] http://stackoverflow.com/questions/20050367/transcendental-equation-solution-by-numerical-method
[19] http://stackoverflow.com/questions/19770107/how-can-i-solve-transcendental-equation
[20] http://www.mathworks.in/matlabcentral/answers/63374-how-to-solve-non-linear-transcendental-equation-with-many-roots