



Computer Scientists: Why Numerical Instead of Analytical Mathematics?

Herbert Wanga¹; Stephano Sanga²; Shangweli Kituma³; Thobius Joseph⁴; Rose Mzilangwe⁵

^{1,3,4,5}Department of Information Technology, University of Iringa (UoI), Tanzania

²Department Mathematics, University of Iringa (UoI), Tanzania

wangahp@gmail.com, stephanosanga@gmail.com, shangwelikituma93@gmail.com, josephthobius@gmail.com, kallunde@gmail.com

DOI: 10.47760/ijcsmc.2020.v09i10.014

Abstract - *Computer scientists deal with computer science. They have strong mathematical components such as automata theory, computational complexity, numerical mathematics, and symbolic mathematics. The single most important skill for a computer scientist is problem solving. Problem solving means the ability to formulate problems, think creatively about solutions, and express a solution clearly and accurately. This article seeks to answer the question as to why should computer scientists deal with numerical mathematics, which give estimates and cause errors, instead of analytical mathematics which give exact answers. Numerical mathematics is a very broad field. In this paper we focus on aspects of numerical mathematics which are related to computer science. Generally, numerical methods require a series of iterations until you come to an estimate close enough to the answer. Computer programs are very efficient in making iterations quickly and correctly. Therefore, computer scientists learn numerical methods so that they can enable people in other fields find solutions to mathematical problems. Computer scientists can use computers to generate the estimates, they can perform 1000, or 10000 iterations in a split of a second and hence get a result of high accuracy. The procedures can be coded easily and hence are well suited for computers. Authors present four basic numerical methods for equation solving: bisection method, Newton-Raphson method, regula falsi and Secant method. Results show that Many practical problems are beyond the scope of analytical mathematics.*

Keywords - *Computer scientist, Numerical method, approximation of a root, Bisection method, Newton-Raphson method, Secant method, regula falsi method, Secant method*

I. INTRODUCTION

Computer scientists are people dealing with computer science. As suggested by [1], Computer science has strong mathematical components such as automata theory, computational complexity, numerical mathematics, and symbolic mathematics.

Like mathematicians, computer scientists use formal languages to denote ideas (specifically computations). Like engineers, they design things, assembling components into systems and evaluating trade-offs among alternatives. Like scientists, they observe the behaviour of complex systems, form hypotheses, and test predictions [2]. The single most important skill for a computer scientist is problem solving. Problem solving means the ability to formulate problems, think creatively about solutions, and express a solution clearly and accurately. As it turns out, the process of learning to program is an excellent opportunity to practice problem solving skills [2]. Computer scientists work in three distinguishable areas: (1) design of hardware components and especially total systems; (2) design of basic languages and software broadly useful in applications, including

monitors, compilers, timesharing systems, etc.; (3) methodology of problem solving with computers [3]. Computer science should concentrate on finding and explaining the principles of problem solving [3].

Mathematics is a merely mental abstraction that serves useful purposes [4]. There are two approaches when finding solutions to Mathematical problems. These are Analytical and Numerical. With analytic approach, given a mathematical problem, we use a procedure and come up with an exact answer. Meanwhile, Numerical methods deal with estimates. They are procedures for solving mathematical problems whose solution is a number. They usually involve a sequence of relatively simple arithmetic procedures. Numerical mathematics is divided into direct methods, which lead to exact answers, such as Gauss elimination method, and iterative methods, by which most numerical methods are. According to [1], Numerical mathematics is the theory and practices of the efficient calculation and error appraisal of approximate solutions of continuous mathematical problems. An equivalent name for the subject is the Analysis of Continuous Algorithms.

The question is, why should computer scientists deal with numerical mathematics, which give estimates and cause errors, instead of analytical mathematics which give exact answers? This article comes with answers.

The remainder of the paper is organized as follows. Section II identifies related works. Section III presents analytical mathematics. Section IV presents numerical methods. Next section V gives numerical methods for computer scientists. Lastly section VI offers some concluding remarks, followed by acknowledgement.

II. RELATED WORKS

In their study [5] prepared a suite of motivational examples which illustrate numerical methods for equation solving. Fixed point iteration, Newton's method, secant method and regula falsi method were implemented as GeoGebra tools in their experience in teaching of numerical mathematics in "Jovan Jovanović Zmaj" high school in Novi Sad is presented.

The advantages introduced by a computer are not only in quicker calculation and drawing. The computer does all the tedious work, which leaves the teacher and the pupils with enough time to discuss the problem, try out multiple ideas and approaches to solving, and, finally, compare and analyze them. The method of solving a problem is as important as its solution. Use of a computer is particularly important when working with pupils who have difficulties understanding all the aspects of solving a mathematical problem [5].

A study by [6] suggest that the MATH as an application provides easy to use tool for calculating roots of nonlinear equations, roots of system of linear equations, differentiate, integrate, approximation, matrix calculation using mentioned numerical methods, to calculate results, estimate errors and much more. The MATH is completed with graphical visualization capable of producing publication quality figures. Future development will be aimed on expanding the numerical methods and graphical capabilities. Engineers, computer scientists and others, often needs to know which method will be most efficient or most precise, and the MATH is capable of providing such information.

Authors [7] developed and used programs allowing students to experiment with well-chosen examples which bring out qualities, good or bad, of numerical techniques. Introducing graphics should make (selected) numerical techniques more meaningful.

III. ANALYTICAL MATHEMATICS

Analytical mathematics is the mathematics that we have grown accustomed to in primary and high school where a mathematical equation is solved using "pen and paper". Mathematical problems that can be solved using analytical mathematics is usually relatively simple. On the other hand, computational mathematics is an approximation technique that leads to algorithms that must be implemented on a computer to solve complex mathematical equations [8]. Many practical problems are beyond the scope of analytical mathematics [9].

IV. NUMERICAL METHODS

Numerical mathematics is the branch of mathematics that proposes, develops, analyzes and applies methods from scientific computing to several fields including analysis, linear algebra, geometry, approximation theory, functional equations, optimization and differential equations. Other disciplines such as physics, the natural and biological sciences, engineering, and economics and the financial sciences frequently give rise to problems that need scientific computing for their solutions[10].

As such, numerical mathematics is the crossroad of several disciplines of great relevance in modern applied sciences, and can become a crucial tool for their qualitative and quantitative analysis. This role is also emphasized by the continual development of computers and algorithms, which make it possible nowadays, using scientific computing, to tackle problems of such a large size that real-life phenomena can be simulated providing accurate responses at affordable computational cost [10].

Numerical mathematics is a very broad field. In this paper we focus on aspects of numerical mathematics which are related to computer science [1]. An important element in numerical mathematics is the appraisal of error [1].

In the process of learning the mathematics and especially in the case of undergraduates, understanding the basics of numerical mathematics plays the key role. In order to be able to choose the best suited numerical method, one has to be aware of how those various numerical methods work, their advantages and disadvantages in a form of a calculation speed, precision and complexity [6].

V. NUMERICAL METHODS FOR COMPUTER SCIENTISTS

In modern devices, due to the complexity of design, we no longer resort to analytical calculations; instead, electromagnetic simulation programs that use numerical methods are now the standard approach [4]. When we carry out engineering in different circumstances, the way we perform mathematics changes. Often the reality is that when analytical methods become too complex, we simply resort to empirical models and simulations [4].

A. Why Numerical methods for Computer Scientists?

Why do we opt to use numerical methods, which gives us an estimate of the answer, instead of analytic methods, which output the accurate answer? Here we come up with three reasons;

- a. When we do not know how to solve them analytically.
- b. When the existing analytic method is much more difficult to carry out, for example $x^5 - 2x - 8 = 0$
- c. For problems for which analytic methods do not exist.

Generally, numerical methods require a series of iterations until you come to an estimate close enough to the answer. Computer programs are very efficient in making iterations quickly and correctly. Therefore, computer scientists learn numerical methods so that they can enable people in other fields find solutions to mathematical problems.

Computer scientists can use computers to generate the estimates, they can perform 1000, or 10000 iterations in a split of a second and hence get a result of high accuracy. The procedures can be coded easily and hence are well suited for computers.

B. Errors in numerical methods

Most numerical methods generate a sequence of estimates, each estimate having an error. The aim of the process is to reduce the error to very small values.

1) Types of errors in numerical methods

- a. The absolute error ϵ , which is the difference between the true value and the estimate. The absolute error ϵ is given by

$$\epsilon = |x - \hat{x}|$$

- b. The relative error ϵ_r which is the ratio of the absolute error to the true value. It is given by

$$\epsilon_r = \frac{|x - \hat{x}|}{x}$$

- c. The percentage error ϵ_p which is the relative error expressed as a percentage. It is given by

$$\epsilon_p = \frac{|x - \hat{x}|}{x} \times 100\%$$

2) Causes of errors in numerical methods

- a. Error due to rounding off. This is an error caused by rounding off of values. To reduce round off error we need to use more decimal places.
- b. Error due to Truncation. It is an error caused by terminating an infinite process of solving a problem.

C. Roots of Nonlinear Functions

In this part we are going to discuss numerical methods of finding roots of functions. A value x_0 is a root of a function f if $f(x_0) = 0$

1) Bisection Method

One of the first numerical methods developed for finding roots of a nonlinear equation $f(x) = 0$ was the bisection method. This is one of the simplest methods [5]. The Bisection tool performs one step of the bisection method. It requires the following input: a function which has real zeros and two points a and b on the x -axis. It produces a new point c on the x -axis with the coordinates $((a+b)/2, 0)$ and it marks the part of the interval which does not contain the zero with a red line [5].

In bisection method we first establish an interval $[a, b]$ in which a root lies. From the Intermediate Value Theorem, it can be shown that for a continuous function f , if $f(a)f(b) < 0$, then we are sure of at least one root in the interval (a, b) . That is, if $f(a)$ and $f(b)$ have different signs, one being positive and the other being negative, then there is root x in the interval between a and b .

Error Bound of the Bisection Method

The error bound of the estimate is the largest possible absolute error in the estimate. Each estimate contains an error, because it is different from the exact solution. Error bound of m_i is given by

$$e_i \leq \frac{b - a}{2^i}$$

2) Newton-Raphson method

The Newton-Raphson method of solving the nonlinear equation $f(x) = 0$ is given by the recursive formula

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

For Newton-Raphson method only one initial approximation of the root is needed to get the iterative process started to find the root of an equation. This method is based on the principle that if the initial guess of the root of $f(x) = 0$ is at x_0 , then if one draws the tangent to the curve at $f(x_0)$, the point x_1 where the tangent crosses the x -axis is an improved estimate of the root [5].

The steps to apply Newton-Raphson method to find the root of an equation $f(x) = 0$ are:

- a. Evaluate $f'(x)$ symbolically.
- b. Use an initial guess of the root, x_0 , to estimate the new value of the root x_1 .
- c. Repeat step 2, using the value x_1 to obtain a new value x_{i+1} .

3) Regula falsi method

Methods such as bisection method and the false position method of finding roots of a nonlinear equation $f(x) = 0$ require bracketing of the root by two guesses. These methods are always convergent since they are based on reducing the interval between the two guesses to zero in on the root [5].

In the regula falsi method, we start with two initial points, x_0 and x_1 , such that $f(x_0)f(x_1) < 0$ so that $f(x) = 0$ has a solution α between x_0 and x_1 . We assume that α is the unique solution to $f(x) = 0$ between x_0 and x_1 . The new approximation x_2 is the point of intersection of the straight line passing through $(x_0, f(x_0))$ and $(x_1, f(x_1))$ with the x -axis:

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

If $f(x_2) = 0$ then $\alpha = x_2$ and we stop. If $f(x_0)f(x_2) < 0$, then we leave x_0 unchanged and continue to the next iteration; otherwise, we set $x_0 = x_1$ and continue to the next iteration in the same way. In case f' and f'' have fixed signs in an interval containing α , which is the situation of interest to us here, the point x_0 ultimately

remains fixed. Therefore, in such a case, the regula falsi method becomes a fixed-point method at some point during the iteration process [5].

Without loss of generality, we will assume that x_0 remains fixed. In this case the regula falsi method is given by

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_0)}{f(x_k) - f(x_0)}.$$

4) The Secant Method.

Given the following function,

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

We can go about the secant method to numerically estimate the root of a function f by

- a. Selecting two numbers x_{-1} , x_0 as initial estimates of the root. The estimates should be near the root.
- b. Calculating the next estimate using the formula

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

- c. We continue until we reach the required tolerance. The secant method does not always converge. Convergence depends on the nature of the function, and on how near the initial estimates are to the solution.

VI. CONCLUSION AND SUGGESTIONS

In order to achieve the single most important skill for a computer scientist of problem solving, computer scientists are recommended to take numerical mathematics. Numerical methods require a series of iterations until you come to an estimate close enough to the answer. Computer programs are very efficient in making iterations quickly and correctly. Computer scientists can use computers to generate the estimates in a split of a second and get a result of high accuracy. The procedures can be coded easily and hence are well suited for computers. In this article, authors present four basic numerical methods for equation solving: bisection method, Newton-Raphson method, regula falsi and Secant method. Results show that Many practical problems are beyond the scope of analytical mathematics. Computer scientists are expected to be able to write a code that can implement numerical methods. They can write in any language they are comfortable with, such as java, C++, and MATLAB.

ACKNOWLEDGEMENT

Author wishes to acknowledge Mr. Michael Peter from the Open University of Tanzania (OUT) for his support in Computer Mathematics.

REFERENCES

- [1] J. F. Traub, "Numerical Mathematics and Computer Science," *Commun. ACM*, vol. 15, no. 7, pp. 537–542, 1972.
- [2] A. G. de la Nación, C. M. Allen Downey, Jeffrey Elkner, D. Llorens, A. . Fallis, and V. B. Gundersen, *How to think like a computer scientist, learning with Python*, vol. 0, no. 6. 2013.
- [3] G. E. Forsythe, "What to do Till the Computer Scientist Comes," *Am. Math. Mon.*, vol. 75, no. 5, pp. 454–462, 1968, doi: 10.1080/00029890.1968.11971015.
- [4] D. Abbott, "The reasonable ineffectiveness of mathematics," *Proc. IEEE*, vol. 101, no. 10, pp. 2147–2153, 2013, doi: 10.1109/JPROC.2013.2274907.
- [5] D. Herceg and Đ. Herceg, "Numerical mathematics with GeoGebra in high school," *Teach. Math. Comput. Sci.*, vol. 2, no. 2008, pp. 363–378, 2009.
- [6] H. Glaser-Opitz and K. Budajová, "THE MATH – OPEN SOURCE APPLICATION FOR EASIER LEARNING OF NUMERICAL MATHEMATICS," *Acta Didact. Napocensia*, vol. 2, no. 4, pp. 45–50, 2012.
- [7] I. B. Jacques and C. J. Judd, "Use of microcomputers in teaching numerical mathematics," *Int. J. Math. Educ. Sci. Technol.*, vol. 16, no. 3, pp. 347–354, 1985, doi: 10.1080/0020739850160303.
- [8] A. Ooi, "An Analysis of the Teaching of Mathematics in Undergraduate Engineering Courses .," in *AaeE Conference, Melbourne*, 2007, pp. 1–7.
- [9] N. Mullineux, J. R. Reed, and S. J. Harborne, "The interplay of numerical and analytical solutions of singular integral equations," *Int. J. Math. Educ. Sci. Technol.*, vol. 3, no. 3, pp. 291–299, 1972, doi: 10.1080/0020739700030312.
- [10] A. Quarteroni, R. Sacco, and F. Saleri, *Texts in Applied Mathematics*, Second edi., vol. 53, no. 9. Berlin: Springer, 2017.