Journal of Physics: Conference Series

PAPER • OPEN ACCESS

GeoGebra as learning tool for the search of the roots of functions in numerical methods

To cite this article: S Arceo-Díaz *et al* 2020 *J. Phys.: Conf. Ser.* **1672** 012001

GeoGebra as learning tool for the search of the roots of functions in numerical methods

S Arceo-Díaz[1], **E E Bricio Barrios**[1], **J Aréchiga Maravillas**[1], **and J Salazar-Torres**[2]

[1] Instituto Tecnológico de Colima/Tecnológico Nacional de México, Colima, México
[2] Facultad de Ciencias Básicas y Biomédicas, Universidad Simón Bolívar, San José de Cúcuta, Colombia

E-mail: elena.bricio@colima.tecnm.mx

Abstract. Physics is capable of describing, through equations, phenomena on a micro and macroscopic scale. However, most of these equations are non-linear and the identification of their roots requires the use of approximation methods, with numerical methods being a proposal based on a systematic and iterative process, that conclude only when a pre-established tolerance is satisfied. Traditional teaching of numerical methods involves the memorization of algorithms. However, this hinders student's ability to understand the important aspects and then apply them for solving applied problems in subjects such as kinematics, dynamics, electromagnetism, etc. Therefore, this work proposes the use of GeoGebra, as a didactic tool to illustrate the functioning of single root searching algorithms. By using the dynamical graphic's view of GeoGebra, a series of abstract and applied problems where solved by engineering students taking a numerical methods course. The scores of this test group was then compared to a test group, taught trough algorithm memorization. Results show can improve their understanding of how the bisection, false position, secant, and Newton-Raphson methods are able to find approximated solutions to polynomial and trigonometric equations. The results are compared against traditional learning, based on memorizing the steps of the algorithm for each method and the representation of the convergence of successive roots by numerical tables.

1. Introduction

Numerical methods are one of the most relevant courses within the curricular subjects of engineering students [1]. In one instance, it serves to provide complementary solving strategies for linear and non-linear equations, systems of linear and non-linear equations, and ordinary differential equations (allowing the calculation solutions by arithmetic operations instead of applying algebraic manipulations) [2]. Also, within a typical numerical methods course, students are shown how the recollection of experimental data are used to propose mathematical models to a wide variety of phenomena, through data fitting and interpolation techniques [1,3]. However, in many cases, the main emphasis of the course is given to the memorization of algorithms and the development of the necessary skills for allowing students to create computational programs. This results, in many instances, in students capable of creating a program to numerically solve a problem but not retaining much understanding of the characteristics of the corresponding algorithm [1].

Solving polynomial equations by employing root searching algorithms is of the most useful skills learned by engineering students [4]. All the classical algorithms (*e.g.* the bisection, false position, secant, and Newton-Raphson) involve the calculation of roots by employing iterative formulas that, provided an initial suggestion for the root (a seed), give approximations whose accuracy increases after many iterations [5]. Once a condition over the value of the numerical approximation is met (*i.e.* the error being smaller than an arbitrarily defined tolerance) the algorithm is considered successful. If the required condition is not met, then the value obtained by the iterative formula is reinserted again to get a new approximation, repeating itself until the condition is met (the sequence of roots and their associated error representing the convergence of the method) [6].

Among other factors, like the continuity and the differentiability of the function defining a polynomial equation, the success of the numerical root searching methods usually depends on the values of the seeds [7]. Closed methods (like bisection and false position) require seeds to be defined within an interval in which the function is continuous and changes its sing. In contrast, open methods (like Newton-Raphson and secant) are more permissive with the value of the seed but they also benefit if the seed is close enough to the exact root (and can also diverge if certain initial values are selected, as with points in which the derivative of the polynomial function is zero for the Newton-Raphson method) [1,4,6]. Selecting the seeds for any numerical root finding method by plotting the polynomial function is already a common suggestion to students (what is known as the graphical method) and, although is using it mostly restricted to seed selection based on visual inspection of the function.

The incorporation of mathematical software, like MatLab, Maple, or Octave within a typical numerical methods course is customary nowadays, as it reduces the time required to calculate a solution with a desirable accuracy [8]. Usually, students are instructed to memorize all of the steps of the algorithm that defines each method (being flux diagrams a common tool for visualizing the steps involved in the algorithm). Once this competence is gained, students are taught on how to translate a given algorithm into any of the many existing programming languages and it is considered that the student has developed an acceptable understanding of a specific method if he or she is capable of translating its algorithm into a fully functional computer program [1]. However, as the output information that can be obtained by a numerical software is a vector composed of the values for the approximated roots (and maybe their corresponding error) students are not able to analyze the characteristic ways in which numerical methods approach a desired root [6].

Besides numerical software, like Octave, communication, and information technologies (ICT's) have characteristics that can be used as an advantage for the teaching of numerical methods. Djamila [8] and Fernandez *et al.* [9] reported the use of excel as a helping tool for teaching students how to calculate the roots of polynomial equations. Kiusalaas [2] and Handayani [10] developed algorithms for each root searching method that can be executed in Phyton and Maple, respectively, and Gomez *et al.* [6] developed a web page in which students can read about the characteristics of each of the root searching methods, do exercises and calculate the root of a given polynomial equation. In recent years, GeoGebra has become very popular for the teaching of basic arithmetic, analytic geometry, differential, integral, and vector calculus [11]. Specifically, GeoGebra is characterized for being an ideal platform for basic programming, as it utilizes simple commands, has a user-friendly interface for data input, and can be used to transform any program developed with it into a dynamical HTML web page [12]. The review done by Wassie and Gurju [11], on the usage of GeoGebra on the teaching of different branches of mathematics, only reported the work by Martin *et al.* [5], developing applications that, by automatically generating all of the iterations for calculating the solution of a given equation, allowed to visualize the progression of approximated roots. However, the results of this study were not validated by doing experimental research with students.

The present work describes how GeoGebra was used within a course in numerical methods as a tool for improving the understanding of students on how common root searching methods are able to calculate accurate numerical approximations for a given polynomial equation. A quantitative quasi-experimental design [13] in which the understanding of relevant concepts, related to the numerical solution of polynomial equations, of an test group of engineering students is compared against a control group.

2. Methodology

Three groups of engineering students were selected for this study. The first group (labeled as control) was composed of 36 students of computer systems engineering. The second group (labeled as alternative) was composed of 12 mechatronics students and 19 students of mechatronics were selected for the third group (labeled as test). Although the individuals involved in the study had different careers, they had the same background knowledge in both mathematics and computing courses (as the necessary competencies for taking the numerical methods are taught in the first semesters of each career). The selection of the mechatronics group as the test group was also motivated by the academic disadvantages they had (as 16 out the 19 students had already failed once or, in some cases, twice to pass the course). In this way, the group that presented less proficiency was selected to take the proposed alternative didactic strategies.

The didactic strategies presented in this work were applied during the second thematic unit of the numerical methods course (whose objective is for students to learn methods for calculating roots of linear and non-linear equations), as an intermediate phase between the learning of the algorithms and their coding into computer programs. The selected methods were the bisection and false position methods (two bracketing methods) and the Newton-Raphson and secant methods (two open methods). These are taught as part of the content material of most courses in numerical methods. The basic algorithm for each method was taught, mostly, by the use of flux diagrams, analysis, and repetition of all the involved mathematical steps and by solving examples by hand, whereas the required programming languages taught to code the algorithms were based in the Octave language.

The working hypotheses of the study where that evidence can be found that graphical tools(as the graphics view of GeoGebra) improve the understanding of students on how each method converges to the exact roots of a given polynomial, either by a significant difference on the test group's average scores on a test (hypothesis a) or by the proportion of successful students (hypothesis b) when compared to the scores of the other two groups.

The control group followed the traditional path of memorizing the algorithms of each method, followed by calculating the first iterations of each method to approximate the solutions to a given polynomial equation (assisted by a handheld calculator). The alternative group incorporated excel to calculate the approximated roots on each of the iterations (emulating the proposal of previous works that suggested that excel could improve the learning of the root-finding methods [8,9]. Finally, the test group was introduced to applications (developed by the professor) which allowed to plot a polynomial function and to define the value of the required seeds (as the Newton-Raphson method requires only one seed while the bisection, false position, and secant methods need two). As output information, these applications show the approximated value of the root, its associated error (defined as the absolute value of the function evaluated at the approximated root, and where the exact root corresponds to the value of the variable x that makes $f(x) = 0$) and, also, gives instruction on how to use this approximated value to calculate the next iteration, corresponding to the usual rules followed by each method [1]. Although previous works, and most numerical methods courses, prefer to construct applications that automatically calculate all the required iterations to reach the desired error tolerance, for this work was left to the student to reinsert the approximated root, obtained at each iteration, to calculate the next one. This was done in purpose since it is believed that allowing students to substitute the values by themselves will give them time to analyze the variations of the position of the interval boundaries (for the bisection and False position method) and the secant and tangent lines (for the secant and Newton-Raphson methods, respectively) and gain insight in the convergence (or divergence) of each method towards the exact root. At the end of each practice, students were asked to make conclusions on each method's convergence speed, accuracy, and how approximations changed at each iteration. The score of each group at that unit's exam was analyzed to test the working hypotheses.

3. Results

As part of the training activities for engineering students, a simple parabolic shooting practice with drag was designed. This problem was stated as: "A batter hits a ball at a height (h_0) of 1 mt high with a velocity of $v_0 = 60\ m/s$ and an angle $\theta = \pi/8$, relative to the horizontal ground. The ball moves through the air with a drag coefficient of $b = 0.5$. Calculate the time when the ball hits the ground without and with the effect of friction". Solving this problem only requires substituting the data in the equations for a parabolic trajectory (Equation (1)) and parabolic draft with drag (Equation (2)):

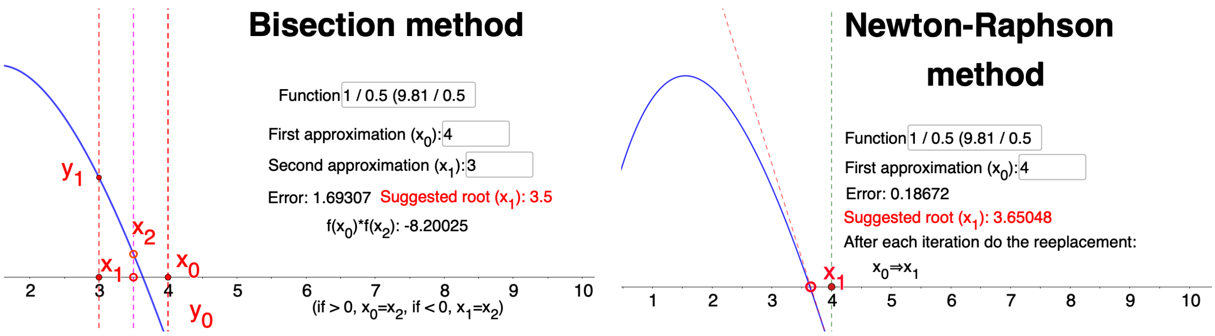
$$y = \frac{1}{2} * g * t^2 + v_0 * sin(\theta) * t + h_0, \tag{1}$$

$$y_{drag} = \frac{1}{b}\left(\frac{g}{b} + v_0 sin(\theta) - \frac{v_0}{2} sin(\pi)\right)\left(1 - e^{-bt}\right) - \left(\frac{g}{b} - \frac{v_0}{2} sin(\pi)\right) * t + h_0, \tag{2}$$

where v_0 is the initial velocity (m/s), θ is the launch angle of the projectile, t is the time in seconds, g is the gravitational constant (9.81 m/s^2) and h_0, measured in meters, is the launch height of the object [14].

Although it is possible to identify the roots through the graphic method, the precision of the resulting estimation is limited. In some instances, it is possible to neglect some factors to simplify the problems. For example, if air drag is ignored the problem can be solved analytically by considering the simple parabolic shot, in which (Equation (1)) describes height as a function of time and whose roots, corresponding to a second-degree polynomial, are $x_1 = 0$ (launching point) and $x_2 = 3.63582$ (impact point). However, eliminating any simplifying assumption requires solving Equation (2) (showing highly non-linear dynamics). Therefore, numerical methods provide a simpler alternative instead of algebraic manipulation. Figure 1 shows the identification of one of the roots by the bisection and Newton-Raphson methods.

Figure 1 shows the first iteration in each of the Geogebra applications designed to approximate roots: Figure 1(a) display the bisection method, where the root of the function is the mean value of two pre-established intervals. Figure 1(b) shows the Newton-Raphson method, that calculates the numerical derivative of the function and then uses the intersection between the tangent line and the horizontal axis to define the approximate root. On the first iteration, students within the test group used the definition of the function as an input to produce the plot and from the plot, they were asked possible values that could be used as a seed to approximate the roots (while the control and alternative groups obtained the graphs with aid of Octave). Each app allowed students to state the values of the seeds and, from them, the plots of different geometric objects where shown: the boundaries of the bracketing interval (for the bisection method) and the secant and tangent line (for the Newton-Raphson method).


Bisection method

Function 1 / 0.5 (9.81 / 0.5

First approximation (x_0): 4

Second approximation (x_1): 3

Error: 1.69307 Suggested root (x_1): 3.5

f(x_0)*f(x_2): -8.20025

(if > 0, x_0=x_2, if < 0, x_1=x_2)

Newton-Raphson method

Function 1 / 0.5 (9.81 / 0.5

First approximation (x_0): 4

Error: 0.18672

Suggested root (x_1): 3.65048

After each iteration do the reeplacement:

x_0⇒x_1

(a)

(b)

Figure 1. Initial settings for finding the roots of Equation (2) that describes the height of a projectile with air drag, as a function of time, trough the Bisection (a), and Newton-Raphson (b) methods.

To evaluate the significant learning of students on the concept of convergence, it was asked to each group to describe it in terms of the tools they had used. Both, the control and alternative groups described convergence by looking at the successive values of roots and the size of the errors in the numerical tables they had generated. Students from the test group described convergence by using a GeoGebra applet developed by the professor (see Figure 2). Within this applet, the absolute error ($\epsilon = |f(x)|$) of each approximate root, calculated by any of the root-finding methods, is used as the radial coordinate in a polar plot. At each point, the distance to the origin corresponds to the associated error. The angle defining the position of each point is calculated by the formula $\theta = 360/n$, with n defined as the number of iteration used to approximate the root (so, if six iterations are required to get an acceptable approximation, then six points are plotted). A slider, corresponding to the tolerance (τ), allows observing how many iterations are needed to fulfill the condition $\epsilon \leq \tau$.

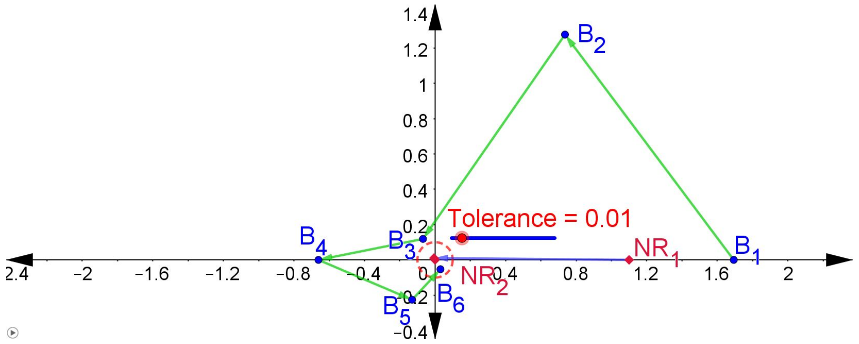


Figure 2. Polar plot that illustrates the convergence of the Bisection and Newton-Raphson method towards finding the positive root of Equation (2).

Figure 2 shows a polar plot of the iterations required for the bisection method to find the positive root of the curve described by Equation (2). Each point corresponds to one iteration of a particular method (labeled as B_1 to B_6, for the Bisection method, and NR_1 and NR_2, for Newton-Raphson) and its polar coordinates are given by the absolute error of that iteration's approximated root (radial coordinate) and the number of degrees with a full circle divided by the number of iterations (polar coordinate). The minimum tolerance is represented as the radius of the dotted circle centered at the origin. While the bisection method requires six iterations to get an approximation whose error falls within the acceptable region (defined by the tolerance to be within a distance of 0.01 from the root), the Newton-Raphson method requires only two iterations to meet the condition.

After a series of practices like the one described above where carried on, during multiple sessions, a test was applied to the three groups. The results of the test where used to evaluate if there was any statistical evidence to support any of the working hypothesis stated above. Table 1 and Table 2 shows the results from the t-score hypothesis test looking for a difference in average scores (Table 1) and a difference in the proportion of students getting a score equal or higher than 70 out of 100 points (Table 2).

Table 1. Relevant parameters for the hypothesis test looking for a significant difference on average scores within each group.

Group	Control	Alternative	Test
Number of students	36.00	12.00	19.00
Average score	62.72	70.58	75.12
Standard deviation	29.87	19.14	14.63
t-score	–	0.852	1.697
P-value	–	0.399	0.096

Table 1 shows the average score for each group. The first row corresponds to the number of students with each group. The second row shows the average score: the control group obtained the lowest average score while the test group got the highest. Also, as can be seen in the third row, these groups show, respectively, the largest and lowest score dispersion. After doing calculating the t-scores for the alternative group and the test group (with their average scores contrasted to that of the control group), it appears that there is slight evidence that the use of GeoGebra improves the learning of students (as the P-value, corresponding to the probability that the difference on average scores does not result from a random casualty is around 9%). The results for the hypothesis test between the average scores of the alternative and control groups are less optimistic, as the P-value is almost 40%.

Table 2. Relevant parameters for the hypothesis test looking for a significant difference on the proportion of successful students within each group.

Group	Control	Alternative	Test
Number of students	36.00	12.00	19.00
Proportion of success	0.555	0.666	0.789
χ^2	–	10.911	2.887
P-value	–	0.001	0.089

Table 2 shows the results for the hypothesis test on the difference in the proportion of students whose score was higher than 70% (the minimal scored needed for students to approve the unit in the Institution where this work was made). The χ^2 test indicates that: the evidence for a statistically significant difference in the proportion of successful students (to a 95% CL) between the control and test groups is just slight (giving a P-value of 9%), while there is no significant difference between the alternative and control groups. All the applets made for this study can be obtained from the corresponding author.

4. Conclusions

This work proposed the use of GeoGebra for solving physics problems within a course of numerical methods for engineering students. Specifically, this work reports the use of numerical methods for the identification of the roots of non-linear functions, describing the vertical distance traveled by a projectile with air drag, as well as graphically showing the convergence of the bisection and Newton-Raphson methods.

In contrast to previous works, the results from the GeoGebra applets developed were validated with the help of students from three different groups, testing two different hypotheses on the possible improvements that could be induced. Although the sample was small the hypothesis test brings some evidence on the improvement in both the average score of students taking a test on the subject, the proportion of students getting the minimal approbation score, and a reduction on the variability of the scores (measured by the standard deviation of each sample), suggesting that the typical skill differences among students could have been diminished when calculating the numerical approximations and allowed them to understand more easily the convergence of the root searching methods. As future work, the development of applets allowing to calculate roots in systems of equations, using a combination between graphical and numerical methods, will be done.

Acknowledgments

The authors want to thank all the students that cooperated as part of any of the involved groups. This work would have not been possible without them. Also, we appreciate the help of the academic authorities of the National Institute of Technology of México (TecNM), and professors of the Technological Institute of Colima, México,that made it possible to present the results of this work.

References

[1] Chapra S, Canole R 2010 *Numerical Methods for Engineers* (New York: McGraw Hill Higher Education)

[2] Kiusalaas J 2013 *Numerical Methods in Engineering with Python 3* (Cambridge: Cambridge University Press)

[3] Tomás X, Cuadros J, González L 2006 *Introducción al Cálculo Numérico* (Barcelona: Institut Químic de Sarrià)

[4] Wylie C, Barrett L 1960 *Advanced Engineering Mathematics* (Pennsylvania: McGraw Hill Education)

[5] Martín-Caraballo A, Tenorio A 2015 Teaching numerical methods for non-linear equations with Geogebra based activities *International Electronic Journal of Mathematics Education* **10** 67-73

[6] Gómez M, Cervantes J, Báez E García A, Ramos R 2015 A software tool to improve teaching numerical methods *Electronic Journal of Mathematics and Technology* **9** 31-35

[7] Lapidus L, Pinder G 2013 *Numerical Solution of Partial Differential Equations in Science and Engineering* (New Jersey: John Wiley and Sons)

[8] Rice J 2014 *Numerical Methods in Software and Analysis* (Amsterdam: Elsevier)

[9] Fernández S Orosa J, Galan J 2012 A new methodology to teach numerical methods with MS Excel *Journal of Maritime Research* **9** 35-41

[10] Handayani A D, Herman T, Fatimah S 2017 Developing teaching material software assisted for numerical methods *Journal of Physics: Conference Series* **895** 012019:1-7

[11] Wassie Y, Zergaw G 2018 Capabilities and contributions of the dynamic Math software, GeoGebra-A review *North American GeoGebra Journal* **7(1)** 68-78

[12] Wassie Y, Zergaw G 2019 Some of the potential affordances, challenges and limitations of using GeoGebra in mathematics education *Eurasia Journal of Mathematics, Science and Technology Education* **15** 1734-1746

[13] Hernández S, Fernández C, Baptista L 2014 *Metodología de la Investigación* (México: McGraw Hill Education)

[14] Resnick R, Walker J, Halliday D 1988 *Fundamentals of Physics* (Hoboken: John Wiley)