International Electronic Journal of Mathematics Education, 2015, **10**(2), 53-65

Teaching Numerical Methods for Non-linear Equations with GeoGebra-Based Activities

Ana M. Martín-Caraballo & Ángel F. Tenorio-Villalón
Pablo de Olavide University, SPAIN

•Received 20 April 2015•Revised 24 May 2015 •Accepted 6 June 2015

This paper exemplifies the potential of GeoGebra as didactic resource for teaching Mathematics not only in High School but even in University. To be more precise, our main goal consists in putting forward the usefulness of GeoGebra as working tool so that our students manipulate several numerical (both recursive and iterative) methods to solve nonlinear equations. In this sense, we show how Interactive Geometry Software makes possible to deal with these methods by means of their geometrical interpretation and to visualize their behavior and procedure. In our opinion, visualization is absolutely essential for first-year students in the University, since they must change their perception about Mathematics and start considering a completely formal and argued way to work the notions, methods and problems explained and stated. Concerning these issues, we present some applets developed using GeoGebra to explain and work with numerical methods for nonlinear equations. Moreover, we indicate how these applets are applied to our teaching. In fact, the methods selected to be dealt with this paper are those with important geometric interpretations, namely: the bisection method, the secant method, the regula-falsi (or false-position) method and the tangent (or Newton-Raphson) method, this last as example of fixed-point methods.

Keywords: college mathematics, mathematics activities, numerical algebra, mathematics instruction, computer-assisted instruction

INTRODUCTION

Pablo de Olavide University (UPO) has been developing its teaching activities according to the framework of European Higher Education Area (EHEA) for the last six years. With the implementation of this then-new methodology and philosophy, both teachers and students have been gradually modifying and updating their approach about the traditional performance and role for each of these two actors in the teaching/learning process. Essentially, we can observe the following three major changes in college education: a) competence-based learning and assessment; b) students as leading characters in the teaching/learning process and teachers come to as advisors in students' knowledge development; and c) "learning to learn" as educational tool to update students' knowledge throughout their professional and social life.

We want to use a classroom model in which students and teachers make a

Correspondence: Ana M. Martín-Caraballo,
Pablo de Olavide University, Ctra. Utrera Km. 1, Seville, 41013, Spain.
Phone: 0034954978044
E-mail: ammarcar@upo.es
doi: 10.29333/iejme/291

ISSN: 1306-3030

working group together, allowing them a correct development and construction of notions and topics in the syllabus; as well as promoting a correct acquisition of professional training skills associated with our courses. Both being competent and having good qualifications are compulsory requirements for our students' educational success. In 2010/2011 academic year, UPO implemented the Bachelor's degree in Computer Science Engineering in Information Systems (CSEIS).

Bermudo et al. (2006) proposed a planning and organizational model for Mathematics and Statistics courses. This model was based on students' autonomous work by using several techniques and tools to asses and to monitor students' academic activities. Indeed, they emphasized that teachers should be considered as advisors and managers for students' knowledge, since students turn into leading actors in their education and training, following recommendations in official documents about degree implementation in Spain – see ANECA (2005) and MECD (2003). Additionally, since students must carry out their courses by autonomous work, it is completely necessary to encourage and promote individualized attention for students in terms of integrated and personalized tuition. To do so, teachers must use any didactic tool being appropriate so that each student works out autonomously syllabuses. The importance of tuition in this methodology can be consulted in Hernández-Jiménez et al. (2008a, 2008b) or Sáenz-Castro (2001).

Nowadays, we are organizing and preparing our Mathematics courses so that students do not only learn contents by heart, but their learning is based on competences. Thus, our undergraduate students must learn to learn in order to continuously update their knowledge in relation with their profession. However, well into the 21st century, our students must compulsorily develop digital competences using software existing to deal with and solve mathematical problems – e.g. Tenorio-Villalón (2008, 2010); which is more relevant when referring to Computer Science. Moreover, teaching and assessing as in the 19th century do not make sense and this is what usually happens with Mathematics. Teachers usually are more interested in their students learning and correctly repeating algorithms with pencil and paper instead of using computational software to do it; as can be seen in Montero (2006) or Pérez-Jiménez (2005). Despite the existence and use of these digital resources, students must learn to correctly apply traditional problem-solving algorithms, but not with problems involving to spend more time than necessary and consisting in computation-based exercises. These latter do not allow us to check and assess if students have assimilated the notions and procedures or if they can modelize real-world problems as mathematical problems, translating mathematical solutions into real-world solutions.

This article focuses attention on competence-based learning and assessment by using Information and Communication Technology (ICT). More concretely, we show one of the uses given to ICT in our courses: the application of GeoGebra as a didactic resource which allows us to explain some notions (about numerical analysis) to students by manipulation of concepts and problems which makes easier their subsequent assimilation.

USING ICT IN OUR MATHEMATICS COURSES

CSEIS covers three Mathematics courses in the first academic year: Algebra, Calculus and Mathematical Methods for Engineering (MME). Precisely, this latter is the one in which we have carried out the experience, explaining some numerical methods with GeoGebra. The syllabus of MME consists in Numerical Algebra and Calculus.

Teaching is divided in theoretical and practical lectures, both of them being worked on with ICT and focused on meaningful and autonomous learning. Our methodology is eminently based on practical training, to develop students' competences related to theoretical aspects and use of numerical methods and their computational treatment. ICT have become essential for our teaching and our practical lectures are running in computer classrooms to work computationally with problems.

Mathematical software allows students to handle a problem, experimenting with different examples and testing what happens after altering initial data without wasting a lot of time repeating all the computations. Thus, students can evolve in their learning and skills via success/failure experiments in a first stage. Once this is done, students can better assimilate notions and procedures worked during the courses. Note that computers do not solve problems by their own, but need students' instructions when inserting data and interpreting answers returned by the software. In this way, if our questions and problems are adequately oriented and prepared, we can check and assess our students' understanding and assimilation of notions and procedures worked in the course. This allows us to detect if our students' mathematical thinking is correctly and adequately evolving, regardless of possible operational difficulties. For instance, see Tenorio-Villalón (2010), Tenorio Villalón, Paralera-Morales & Martín-Caraballo (2010) or Tenorio-Villalón & Oliver-García (2011).

We mainly use mathematical software for symbolic computation, but complemented by other resources like GeoGebra. Thus, GeoGebra is a resource for our teaching and symbolic computation packages are used for students`' learning process.

REVIEWING NUMERICAL METHODS TO SOLVE NON-LINEAR EQUATIONS

We have just commented that GeoGebra is used to complement our theoretical explanations, allowing students to manipulate notions and methods and to make easier their understanding and assimilation. Before showing how to apply GeoGebra in practice, we review the numerical (both recursive and iterative) methods to solve non-linear equations, which are explained using GeoGebra in Section 4. For a detailed introduction, the interested reader can consult Burden & Faires (2000).

Numerically, any (linear or non-linear) equation can be expressed as $f(x)=0$, where $f:[a,b]\to\Re$ is a real function in a finite interval $[a,b]\subset\Re$. A numerical method to solve this equation consists in obtaining a sequence $\{x_k\}_{k\in\aleph}$, with limit being equal to one (maybe the unique) solution of the equation. Each x_k is called iteration and corresponds to an estimation of the solution. To obtain the subsequent iteration x_{k+1}, we may need only the previous iteration x_k or more than one (in our case, not more than two iterations x_k and x_{k-1}). The computation of x_{k+1} is given by an iteration formula, which may be iterative or recursive depending on using only one or more than one previous iterations, respectively. Note that both the function f and their (higher-order) derivatives may appear in the iteration formula of a numerical method.

Depending on the method to be considered, additional hypotheses must be imposed to the function f so that the method converges to the solution of $f(x)=0$. Firstly, we must find out intervals in the domain of f containing a unique solution of the equation. There are several techniques to obtain such intervals:

- A naive method consists in using the graph of f to determine graphically intervals. In that case, we cannot assess how students apply theoretical mechanisms to separate solutions; but only the notion per se and its graphical application.

- Alternatively, there exist theoretical methods to locate and separate solutions of a given equation. These methods allow us to assess if students understand not only the method and the notion, but also how to appropriately apply them. In the course, we only explain and apply the location method based on determining closed intervals where f is non-increasing or non-decreasing and then checking the existence of opposite signs in the extremes of those closed intervals (i.e. applying Bolzano's Theorem to each of them). Note that f must be continuous to apply Bolzano's Theorem; in fact, f should be continuously differentiable to easily study monotony.

There exists a second theoretical method for non-linear polynomial equations without multiple solutions (or roots): Sturm's Theorem based on constructing Sturm polynomials. This method is not applicable to non-polynomial equations, but does not require determining where the polynomial increases or decreases. Since this method is not applicable to non-polynomial equations and its difficulty is higher for our students, we hardly ever explain it.

After separating all the solutions of the equation, we must apply a numerical method to solve it. We explain several recursive and iterative methods to estimate solutions of non-linear equations, which will be reviewed in the following subsections and which can be geometrically explained instead of using the traditional algebraic-functional approach.

RECURSIVE METHODS: BISECTION, SECANT AND REGULA-FALSI

Bisection method

This method computes an estimation of the (not necessarily simple) unique solution of $f(x)=0$, where $f:[a,b]\rightarrow\Re$ is a continuous real function such that $\boldsymbol{f}(a)\cdot \boldsymbol{f}(b)<0$ (i.e. the interval extremes have opposite signs). This method uses an iteration formula computing the iteration x_{k+1} starting from x_k and x_{k-1}. Hence, two initial iterations $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are needed as inputs to initiate the method.

The recursive iteration formula is obtained as follows: the initial iterations $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are defined as the interval extremes; i.e. $x_0=a$ and $x_1=b$. Then, we compute the midpoint of $[a,b]$ as $x_2=\frac{x_0+x_1}{2}$. If $f(x_2)=0$, then the iteration x_2 is the exact solution of the equation and the method stops. Otherwise, we consider the two possible subintervals $[x_0,x_2]$ and $[x_2,x_1]$ by splitting the initial interval $[x_0,x_1]=[a,b]$. Due to the opposite signs in the extremes of $[a,b]$, opposite signs must also be found in either $[x_0,x_2]$ or $[x_2,x_1]$. Thus, if $f(x_2)\cdot f(x_0)<0$, the change of sign is located in $[x_0,x_2]$ and this subinterval contains the solution; otherwise, we consider the interval $[x_2,x_1]$. The next iteration x_3 is computed as the midpoint of the subinterval chosen with the previously explained criterion:

$$x_3=\begin{cases}\dfrac{x_0+x_2}{2}, & \text{if } f(x_0)\cdot f(x_2)<0\\ \dfrac{x_2+x_1}{2}, & \text{if } f(x_0)\cdot f(x_2)>0\end{cases}$$

Once computed the iteration x_3, the interval used in its computation is split into two new subintervals in which the sign of f is studied again to decide the subinterval containing the solution.

For each iteration, the length of the interval decrease in the half. Hence, the sequence $\{x_k\}_{k\in\aleph}$ converges to the solution in the interval $[a,b]$. Hence, the convergence of this method does not require additional hypotheses.

Secant method

This method computes an estimation of the (not necessarily simple) unique solution of $f(x)=0$, where $f:[a,b]\rightarrow\Re$ is a continuous real function. As the previous one, this also uses an iteration formula computing the iteration x_{k+1} from the iterations x_k and x_{k-1}. Hence, two initial iterations x_0 and x_1 are needed as inputs.

Once x_0 and x_1 are fixed as the interval extremes (as in bisection), the next iteration x_3 is computed as follows:

(1) Construct the line through points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ in the graph of f, using the symmetric equation $\dfrac{y-f(x_1)}{f(x_1)-f(x_0)}=\dfrac{x-x_1}{x_1-x_0}$.

(2) Compute the intersection point x_2 of this line and the x-axis (of implicit equation $y=0$). Therefore, x_2 is expressed as $x_2=\dfrac{x_1\cdot f(x_0)-x_0\cdot f(x_1)}{f(x_1)-f(x_0)}$ (Figure 1 is a geometrical representation of this method).This expression is trivially generalized to compute x_{k+1} from x_k and x_{k-1}.

The secant method is not always convergent to the solution. To assure the convergence, the function f must be two-times continuously differentiable in the interval $[a,b]$ and the solution must be simple. However, there are not criteria to decide the value of initial iterations assuring the convergence. This decision must be done by empirical testing.

Regula-falsi (or false-position) method

This method consists in combining and applying together the previous two methods to obtain a composite method as follows: bisection always assures the convergence to the solution, whereas secant is quicker if converges. Hence, we use

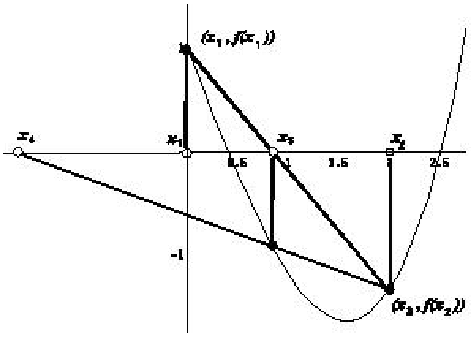


Figure 1. Geometrical representation of the secant method

the iteration formula of the secant method but choosing the interval with the criterion of bisection.

Starting from two initial iterations x_0 and x_1 such that $f(x_0)\cdot f(x_1)<0$, the regula-falsi method computes the iteration using the iteration formula of the secant method: $x_2 = \dfrac{x_1\cdot f(x_0)-x_0\cdot f(x_1)}{f(x_1)-f(x_0)}$.

To compute the iteration x_3, we use the test in the bisection method to detect if the solution of belongs to the interval $[x_0,x_2]$ or the interval $[x_2,x_1]$:

(1) If $f(x_0)\cdot f(x_2)<0$, then choose the interval $[x_0,x_2]$ and

$$x_3 = \frac{x_2\cdot f(x_0)-x_0\cdot f(x_2)}{f(x_2)-f(x_0)}.$$

If $f(x_0)\cdot f(x_2)>0$, then choose the interval $[x_2,x_1]$ and

$$x_3 = \frac{x_2\cdot f(x_1)-x_1\cdot f(x_2)}{f(x_2)-f(x_1)}.$$

We show a geometrical representation of the regula-falsi method in Figure 2.

Iterative or fixed-point methods: tangent or Newton-Raphson

The tangent (or Newton-Raphson) method estimates the solution (which must be simple) of $f(x)=0$ by only using the previous iteration to the one to be computed with the iteration formula. Hence, a unique initial iteration x_0 is necessary to initialize. Its iteration formula is iterative and consists in computing the iteration as the intersection point of the x-axis and the tangent line $y=f(x_0)+f'(x_0)\cdot(x-x_0)$ to the function f in the point $(x_0, f(x_0))$. Thus, the iteration x_1 is computed as $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$, and, in general, the iteration formula is $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$.

By construction, the method involves that f must be continuously differentiable in an interval containing the solution. To assure the convergence, f must be two-times continuously differentiable in such an interval. Furthermore, Fourier's Rule allows us to determine the value of the initial iteration x_0 only rewriting the hypotheses implying the convergence. We show in Figure 3 a geometrical representation of the method of the tangent.

To conclude, we want to indicate that there exist variants of this method to be applied for multiple solutions, being the most usual given by the iteration formula

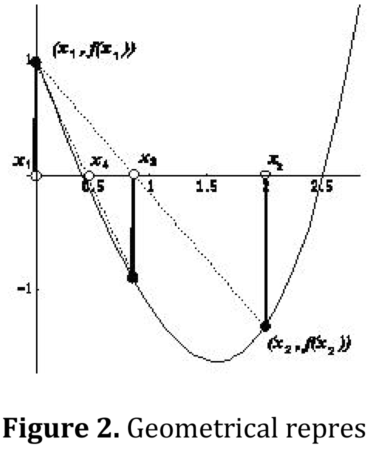


Figure 2. Geometrical representation of the regula-falsi method

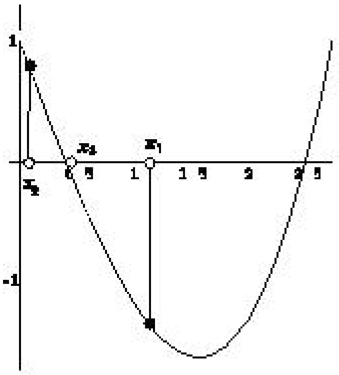


Figure 3. Geometrical representation of the method of the tangent

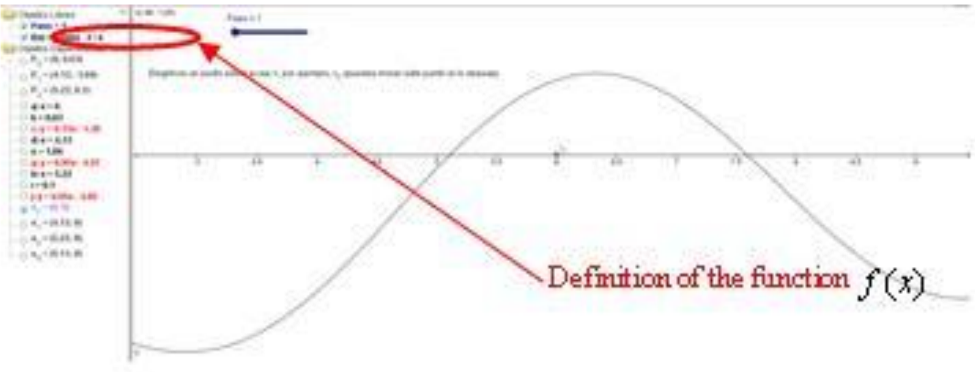


Figure 4. Defining the function in the applet

$x_{n+1} = x_n - m \cdot \dfrac{f(x_n)}{f'(x_n)}$, where m is the multiplicity of the solution. Results about the convergence of the method can be adjusted to this variant, generalizing the case of simple solution (i.e. $m = 1$).

Using GeoGebra to teach numerical methods

Teachers have deemed convenient that several tools should be made available to our students for visualization of how numerical methods work making easier its assimilation by manipulation.

GeoGebra is very helpful and useful to geometrically explain numerical methods solving equations in the classroom. Due to their strong geometric meaning, interactive dynamic geometry software allows students to simulate and manipulate iterative or recursive methods, being of great pedagogical use. Bearing this in mind, we have developed GeoGebra applets to work geometrically on these numerical methods. Next, we explain concisely these applets and how they work and are used by our students.

Iterative or Fixed-Point Methods: tangent or Newton-Raphson

For the tangent method, our applet only requires inputting the function f as one of the free objects (not depending on others), as showed in Figure 4.

Next, students must graphically place the value of the initial iteration x_0 in the x-axis (we do it in Figure 5). Obviously, this value is also a free object, being updatable by scrolling the point x_0 along the x-axis. This fact allows our student to empirically check the convergence of the method by using the applet.

After inserting the value of x_0, the following step to do with the applet consists in graphically representing the point P_0 associated with the value of $f(x_0)$ in the graph of f, (See Figure 6). The coordinates of P_0 are the first dependent objects (i.e. objects defined in terms of others previously computed or inserted).

Then, Figure 7 shows how the applet draws the tangent to the graph of f in the point P_0, as well as returning its analytic expression. Additionally, the intersection point x_1 of this tangent and the x-axis is also returned both graphically and analytically. Precisely, the analytic expression of x_1 is given as a dependent object.

Once the iteration x_1 is computed, three objects are generated: a) the point P_1 of the graph for x_1; b) the tangent to the graph in the point P_1; and c) the intersection point x_2 of this tangent and the x-axis. We show it in Figure 8.

The applet includes the computations to obtain a third iteration x_3 (see Figure 9).

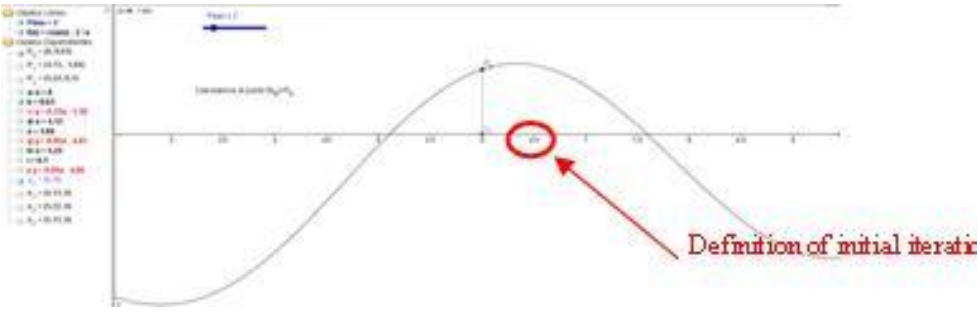
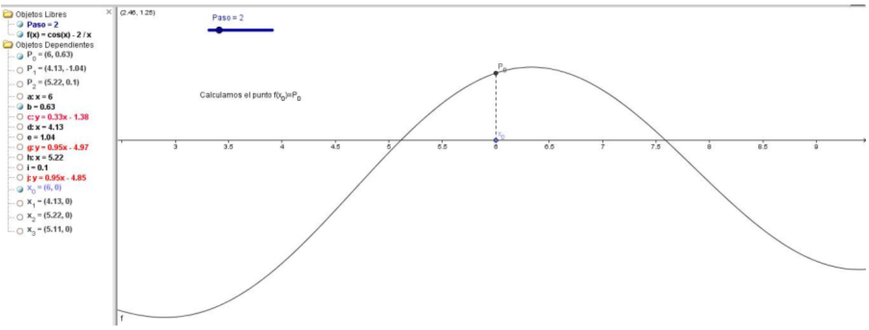

Definition of initial iterati

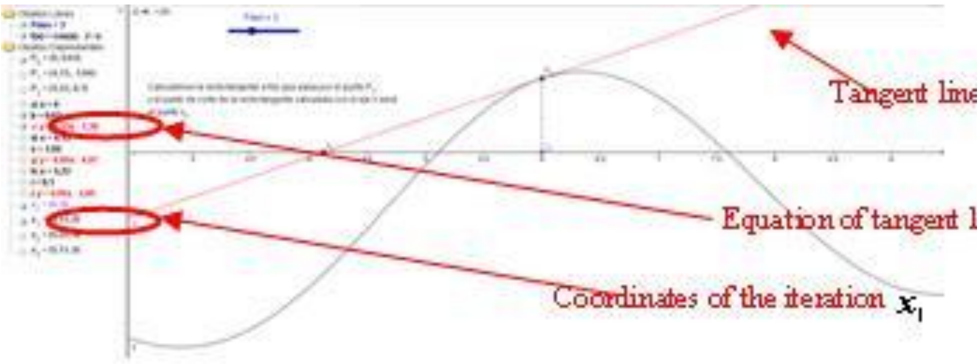
Figure 5. Defining the initial iteration in the applet


Objetos Libres
- Paso = 2
- f(x) = cos(x) - 2 / x

Objetos Dependientes
- P_0 = (6, 0.63)
- P_1 = (4.13, -1.04)
- P_2 = (5.22, 0.1)
- a: x = 6
- b = 0.63
- c: y = 0.33x - 1.38
- d: x = 4.13
- e = 1.04
- g: y = 0.95x - 4.97
- h: x = 5.22
- i = 0.1
- j: y = 0.95x - 4.85
- x_0 = (6, 0)
- x_1 = (4.13, 0)
- x_2 = (5.22, 0)
- x_3 = (5.11, 0)

Calculamos el punto $f(x_0)=P_0$

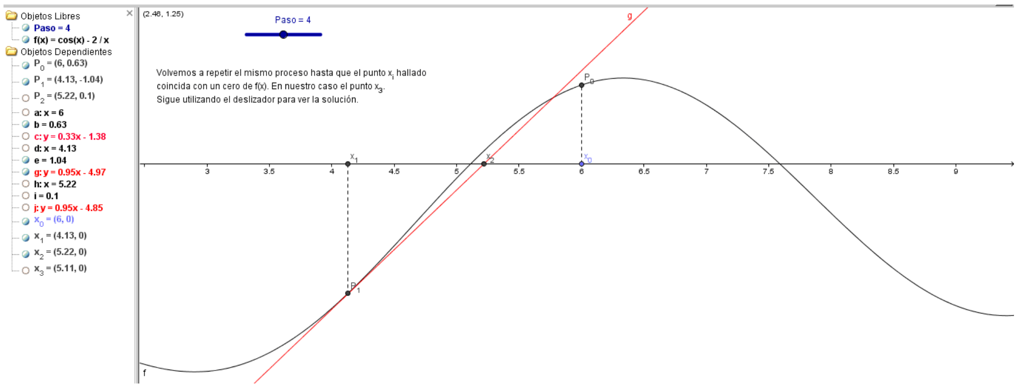
Figure 6. Computing the point $P_0 = (x_0, f(x_0))$


Tangent line

Equation of tangent l

Coordinates of the iteration x_1

Figure 7. Computing and drawing the tangent and the iteration x_1



Objetos Libres
- Paso = 4
- f(x) = cos(x) - 2 / x

Objetos Dependientes
- P_0 = (6, 0.63)
- P_1 = (4.13, -1.04)
- P_2 = (5.22, 0.1)
- a: x = 6
- b = 0.63
- c: y = 0.33x - 1.38
- d: x = 4.13
- e = 1.04
- g: y = 0.95x - 4.97
- h: x = 5.22
- i = 0.1
- j: y = 0.95x - 4.85
- x_0 = (6, 0)
- x_1 = (4.13, 0)
- x_2 = (5.22, 0)
- x_3 = (5.11, 0)

Volvemos a repetir el mismo proceso hasta que el punto x_i hallado coincida con un cero de f(x). En nuestro caso el punto x_3. Sigue utilizando el deslizador para ver la solución.

Figure 8. Computing and drawing the points P_1 and x_2 and the tangent in P_1

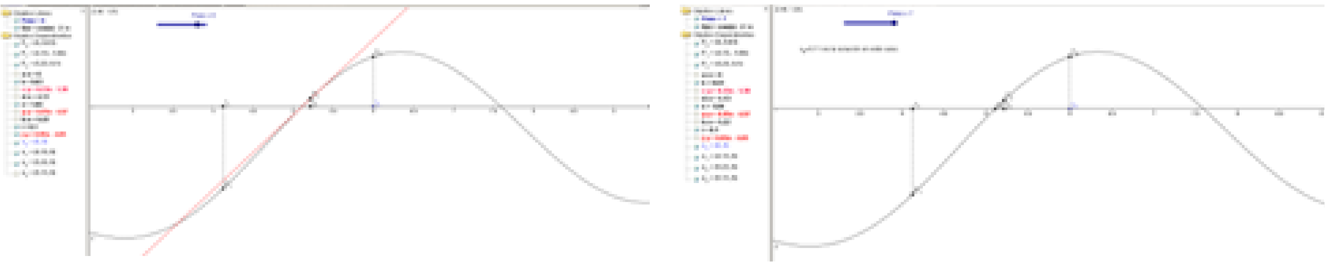
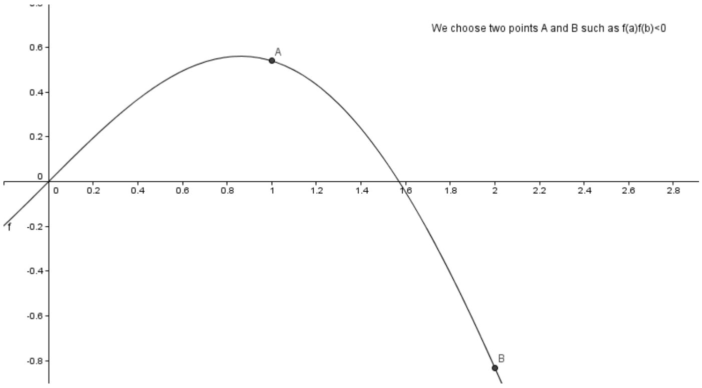


Figure 9. Screenshots for the computations involved in iteration x_3



We choose two points A and B such as f(a)f(b)<0

Figure 10. Defining the initial iteration in the applet

Recursive methods

Although there exist several interesting recursive method to numerically solve nonlinear equations, only three of them are worked on in our lectures, namely; bisection, secant and regula-falsi. To shorten this article, we only show applets for the first and the last.

Regula-falsi (or false-position) method

For the regula-falsi method, our applet only requires to insert the following three free objects: the function f and two points a and b such that $f(a)\cdot f(b)<0$, as showed in Figure 10.

Now in Figure 11 is showed how we calculate the intersection point x_1 of the secant line AB and the line $y=0$.

Once x_1 is computed, three dependent objects are generated: a) the point $C=(x_1, f(x_1))$ representing the image of x_1 under f on its graph; b) the secant line CB ; and c) the intersection point x_2 of the segment CB and the line and $y=0$

At this point, our students might ask us why we are considering the secant line CB and not the secant line AC . If we recall the construction of iterations reviewed in Section 3, the method need opposite signs in the extreme of the interval to generate the following iteration. Students must decide which of the intervals $[a, x_1]$ or $[x_1, b]$ is appropriate to construct the iteration x_2 by studying the existence of opposite signs in the extreme of these intervals under f . If the signs were different in extremes of $[a, x_1]$, the secant line would be AC ; whereas the secant line CB would be chosen otherwise. This question is posed to our students, who must take the decision based on theoretical arguments. In this case, the correct decision is the interval $[x_1, b]$ and, hence, the secant line CB .

We could implement an "If" sentence to take this decision. However, this latter would involve that our students might use the sentence and apply the method automatically; although they might not be aware of the need to test the existence of opposite signs in extremes of the interval where the construction is carried out.

After this digression, we picked up the application of the method after computing x_2 . Thus, we obtain two intervals $[x_1, x_2]$ and $[x_2, b]$. Once more, we test which interval shows opposite sign in their extremes to choose the interval in which the iteration x_3 will be computed. This procedure is repeated as many times as necessary. Figure 12 shows the construction of iterations x_2 , x_3 and x_4 .

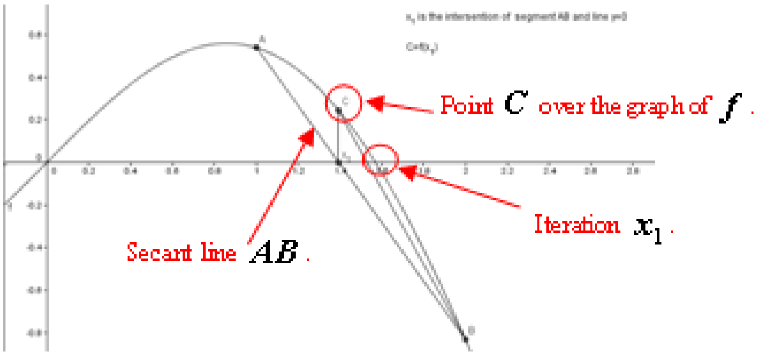

x1 is the intersection of segment AB and line y=0
C=f(x1)
A
C
Point C over the graph of f .
Iteration x1 .
Secant line AB .
B

Figure 11. Computing the intersection point x_1 and its representation $C=(x_1, f(x_1))$ on the graph

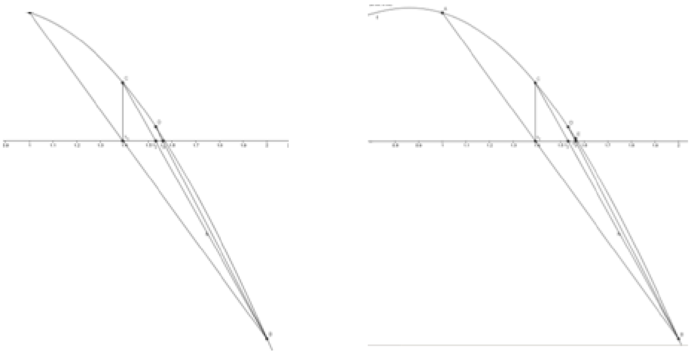


Figure 12. Screenshots with the computations related to the iterations x_2 , x_3 and x_4

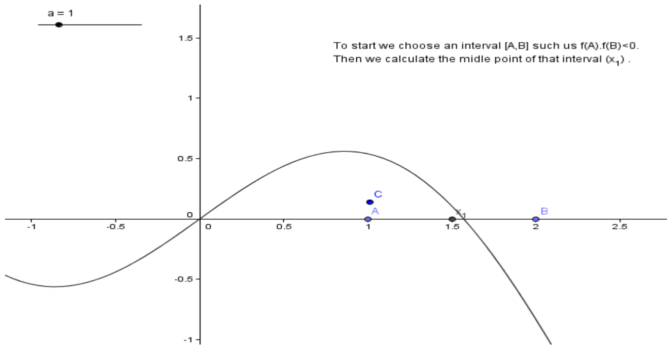
Bisection method

For this method, our applet requires the same three free objects as in regula-falsi, since the difference between both methods concerns the computation of the iteration, but not the decision of the interval to carry out this computation. Hence, we need to insert f and two points a and b such that $f(a)\cdot f(b)<0$.

Then, we compute the midpoint of the interval $[a,b]$, which splits the interval in subintervals $[a,x_1]$ and $[x_1,b]$ as shown in Figure 13.

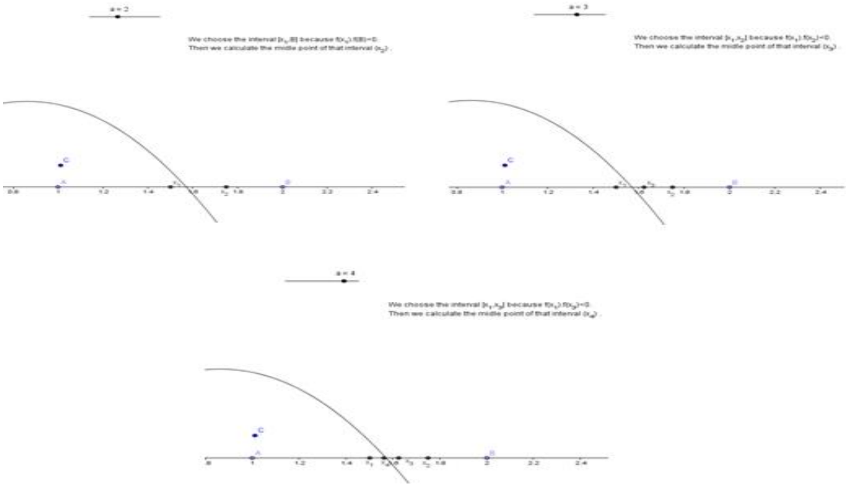
Next, as in regula-falsi, our students must decide which interval is appropriate to continue. The decision is based on testing which of them shows opposite signs in the extremes under f. The same digression as in the abovementioned method can be done in relation with not inserting "If" sentences.

Taking into account that the change of sign for f happens in the extremes of the interval $[x_1,b]$, this is the interval which we must use to construct the iteration x_2 as its midpoint. This is the procedure to carry out per iteration. The computations using GeoGebra for this method is exemplified in Figure 14 for the first three iterations. In fact, after these four iterations, the method returns an iteration which approximates the solution of the equation $f(x)=0$ with a good accuracy.



To start we choose an interval [A,B] such us f(A).f(B)<0.
Then we calculate the midle point of that interval (x_1) .

Figure 13. Defining the interval $[a,b]$ and the initial iteration x_1



We choose the interval [x_1,B] because f(x_1).f(B)<0
Then we calculate the midle point of that interval (x_2) .

We choose the interval [x_1,x_2] because f(x_1).f(x_2)<0
Then we calculate the midle point of that interval (x_3)

We choose the interval [x_1,x_3] because f(x_1).f(x_3)<0
Then we calculate the midle point of that interval (x_4) .

Figure 14. Screenshots with the computations involved in the iterations x_2, x_3, and x_4

CONCLUSIONS

We have explained how to use GeoGebra in the teaching and learning of numerical methods of non-linear equations. Using these applets, we show to our students the geometrical behaviour of both iterative and recursive numerical methods to solve a non-linear equation $f(x)=0$. Furthermore, GeoGebra allows our students to experiment with these methods by simulating different situations when modifying f and/or the initial iteration x_0. Precisely, experimenting and manipulating with these methods is one of the best didactic techniques to learn these methods significantly, since students observe differences appearing for iterations after modifications in the initial data of each method. Moreover, they can test the convergence of the methods for the inputs f and x_0. Obviously, every asseveration given by students must be based on theoretical reasoning, but this latter is not possible until students correctly understand the method and procedure involved in its application. These are the advantages and benefits to use GeoGebra when explaining numerical method.

REFERENCES

ANECA (2005, March). *Libro Blanco del Título de Grado en Ingeniería Informática*. Agencia Nacional de Evaluación de la Calidad y Acreditación. Retrieved from http://www.aneca.es/var/media/150388/libroblanco_jun05_informatica.pdf.

Bermudo, S., Moreno, P., & Tenorio, A.F. (2006). Una experiencia piloto en la Universidad Pablo de Olavide: "Fundamentos Matemáticos de la Informática I y II" y "Estadística" de la Ingeniería Técnica de Informática de Gestión. In *Actas I Jornadas Nacionales de Intercambio de Experiencias Piloto de Implantación de Metodologías ECTS (CD)* (8pp.). Badajoz: Universidad de Extremadura.

Burden, R.L., & Faires, J.D. (2001). *Numerical Analysis*. Brooks/Cole: Pacific Grove.

Hernández-Jiménez, B., Moreno-Navarro, P., & Tenorio-Villalón, A.F. (2008a). La tutoría: un apoyo sustancial de la Estadística–Matemáticas en la Ingeniería Técnica en Informática de Gestión de la Universidad Pablo de Olavide. In M. Zapata-Ros (Ed.), *Seminario Internacional RED-U 2-08: "La acción tutorial en la Universidad del siglo XXI"* (8 pp.). Murcia: RED-U.

Hernández-Jiménez, B., Moreno-Navarro, P., & Tenorio-Villalón, A.F. (2008b). ¿Cómo se enfoca la metodología ECTS y la virtualización en las asignaturas de contenido estadístico-matemático de la Ingeniería Técnica en Informática de Gestión de la Universidad Pablo de Olavide? In R. Peña Ros, P.A. Castillo Valdivieso & M. Anguita López (Eds.), *Actas XIV Jornadas de Enseñanza Universitaria de la Informática* (pp. 377-384). Madrid: Librotex.

MECD (2003,.Febrero). *La integración del sistema universitario español en el Espacio Europeo de Enseñanza Superior*. Ministerio de Educación, Cultura y Deporte. Retrieved from http://tecnologiaedu.us.es/mec2011/htm/mas/2/21/7.pdf.

Montero, L.J. (2006). Enseñanza de la Matemática dominada por algoritmos versus una enseñanza más conceptual. En *Actas digitales del I Encuentro de Enseñanza de la Matemática*. Costa Rica: Universidad Estatal a Distancia.

Pérez-Jiménez, A.J. (2005). Algoritmos en la enseñanza y el aprendizaje de las Matemáticas. *Unión: Revista Iberoamericana de Educación Matemática*, 1, 37-44.

Sáenz Castro, C. (2001). Una nueva función formativa: la tutoría telemática. *Tarbiya: Revista de Investigación e Innovación Educativa*, 29, 119-133.

Tenorio-Villalón, A.F. (2008). Propuesta de actividades con calculadora gráfica para el tratamiento de operaciones matriciales en el aula. *Unión: Revista Iberoamericana de Educación Matemática*, 15, 171-190.

Tenorio-Villalón, A.F. (2010). Resolución de problemas asistida por software matemático: evaluando conocimientos y procedimientos en el alumnado. In *Actas del VI Congreso Internacional de Docencia e Innovación Universitaria* (23pp.). Barcelona: Universitat Politecnica de Catalunya.

Tenorio-Villalón, A.F., Paralera-Morales, C., & Martín-Caraballo, A.M. (2010). Evaluación mediante competencias digitales: una experiencia con Mathematica. *Epsilón: Revista de la SAEM THALES*, 75, 123-136.

Tenorio-Villalón, A.F., & Oliver-García, E. (2011). Una experiencia docente sobre evaluación continua y seguimiento personalizado del alumnado de ingeniería en asignaturas de Matemáticas. In *Actas de las VIII Jornadas Internacionales de Innovación Universitaria* (8pp.). Madrid: Universidad Europea de Madrid.

◈◇◈