Innovando la Enseñanza del método de bisección con ejercicios programables

Filánder **Sequeira** Chavarría
Escuela de Matemática, Universidad Nacional y Universidad de Costa Rica
Costa Rica
filander.sequeira.chavarria@una.cr / filander.sequeira@ucr.ac.cr
Helen **Guillén** Oviedo
Escuela de Matemática, Universidad Nacional
Costa Rica
hellen.guillen.oviedo@una.ac.cr

Resumen

Este artículo presenta una propuesta didáctica compuesta por tres ejercicios innovadores orientados a mejorar la comprensión del método de bisección mediante su implementación en MATLAB. La experiencia se realizó con nueve estudiantes del curso de Métodos Numéricos de la carrera de Bachillerato en la Enseñanza de la Matemática (BLEM) de la Universidad Nacional (UNA), Costa Rica, quienes resolvieron individualmente uno de los ejercicios asignados. Siete de ellos lograron modificar correctamente el algoritmo, evidenciando que la programación facilita la comprensión del proceso iterativo y la detección de errores comunes. Sin embargo, también se identificó como dificultad la traducción del lenguaje matemático al computacional, especialmente en estudiantes sin formación previa en programación. Los resultados destacan el potencial de las herramientas computacionales como apoyo didáctico en el aprendizaje de métodos numéricos.

Palabras clave: Método de bisección, Métodos Numéricos, Programación, Educación Matemática; MATLAB.

Introducción

El avance tecnológico en campos como la inteligencia artificial y el análisis de datos ha resaltado la relevancia de los algoritmos numéricos en la resolución de problemas computacionales, los cuales se abordan en los cursos universitarios de Métodos Numéricos,

disciplina clave para obtener soluciones aproximadas cuando los métodos analíticos no son viables (Flórez et al., 2019). Sin embargo, en carreras de la Universidad Nacional y la Universidad de Costa Rica, muchos estudiantes carecen de formación previa en programación, lo que dificulta su aprendizaje al enfrentarse simultáneamente a conceptos computacionales y matemáticos. Además, el enfoque predominantemente procedimental de estos cursos puede limitar el desarrollo de habilidades como el razonamiento lógico y el pensamiento crítico, reduciendo el aprendizaje a una ejecución mecánica (Montero et al., 2015).

El método de bisección, aunque útil y sencillo para encontrar raíces de ecuaciones, suele enseñarse desde un enfoque teórico que no siempre permite a los estudiantes apreciar su utilidad (Burden et al., 2015; Chapra, 2008; Süli, 2003). Integrar la programación como herramienta didáctica favorece la comprensión del proceso iterativo, el desarrollo de habilidades computacionales y la apropiación conceptual del método. En este marco, se propone una serie de ejercicios programables que buscan fortalecer la comprensión y aplicación del método de bisección desde un enfoque práctico.

Marco Teórico

Uno de los problemas más significativos a abordar en todo curso sobre Métodos Numéricos corresponde a la resolución de ecuaciones en una dimensión, o equivalentemente, a la búsqueda de ceros de una función de una variable. Más precisamente, es de gran interés hallar las posibles soluciones a ecuaciones de la forma: $f(x) = 0$ para $x \in [a, b]$ donde $f: [a, b] \to IR$ es al menos continua sobre $[a, b]$. Su importancia surge en la necesidad de dar solución a problemas en múltiples disciplinas como la modelización de fenómenos físicos (por ejemplo, en Análisis Estructural, Mecánica de Fluidos y Electromagnetismo), biológicos (por ejemplo, en Modelamiento de crecimiento poblacional) y la optimización de procesos financieros (por ejemplo, en Modelos de oferta y demanda), por mencionar algunas.

A primera impresión se puede considerar un problema simple, pero de hecho no es algo que se resuelva de forma inmediata. En efecto, a manera de ejemplo, considere la ecuación $e^x - 2x - 1 = 0$ sobre $[1,2]$ (Süli, 2003), no es difícil comprobar que no es posible despejar la variable x explícitamente debido a la ausencia de técnicas naturales en estos procesos (factorización, leyes de potencias, propiedades de exponenciales, entre otras), en virtud de que es una ecuación que combina distintas familias de funciones (la exponencial con la polinomial). En virtud de ello se descartan las técnicas deterministas y se consideran estrategias numéricas con el fin de hallar una aproximación suficientemente aceptable para esta ecuación. La más natural en todo texto sobre el tema corresponde al método de bisección, primero porque su deducción suele ser inmediata cuando se aborda como consecuencia del Teorema de Valores Intermedios (Burden et al., 2015; Chapra, 2008).

Método de bisección

El método de bisección se define como sigue. Sea $f: [a, b] \to IR$ una función continua sobre $[a, b]$, tal que $f(a)f(b) \leq 0$. Se conoce como la iteración de bisección a la sucesión numérica $\{x_k\}_{k \in IN} \subseteq [a, b]$, definida recursivamente por:

$$x_k \coloneqq \frac{a_k + b_k}{2}, \qquad \text{para } k = 1,2,\ldots,$$

donde se cumple que:

$$[a_k, b_k] \coloneqq \begin{cases} [a, b] & \text{si } k = 0 \\ [a_{k-1}, x_{k-1}] & \text{si } k \geq 1 \wedge f(a_k)f(x_k) < 0 \\ [x_{k-1}, b_{k-1}] & \text{si } k \geq 1 \wedge f(a_k)f(x_k) > 0 \end{cases}$$

La sucesión generada por el método de bisección garantiza converger a un cero de la función f sobre el intervalo $[a, b]$ siempre que se cumpla que $f(a)f(b) \leq 0$. Es importante tener en cuenta que la solución dada por bisección es única, pero esta depende del intervalo inicial. Así, si se desean otras soluciones se debe elegir un intervalo inicial adecuado, donde preferiblemente exista un único cero de la función. (Burden et al., 2015; Chapra, 2008, Süli, 2003).

MATLAB

MATLAB es una herramienta ampliamente utilizada en la enseñanza y aplicación de métodos numéricos por su eficiencia, facilidad de uso y enfoque específico en cálculos matemáticos y científicos. Su entorno optimizado y su sintaxis intuitiva permiten implementar algoritmos sin preocuparse por aspectos técnicos complejos, además de ofrecer funciones visuales útiles para analizar la convergencia y estabilidad de los métodos. Estas características lo hacen superior a otros lenguajes de propósito general como Python o C++, especialmente en contextos académicos e industriales (Yang et al., 2005; Chapra, 2008).

Enfoques de la Educación Matemática

Una de las corrientes más relevantes en la Educación Matemática para este estudio es el enfoque basado en la resolución de problemas, el cual sostiene que los estudiantes construyen conocimiento matemático cuando se enfrentan a situaciones desafiantes que demandan exploración, reflexión y análisis (Schoenfeld, 1992). Esta perspectiva se complementa con el constructivismo, desde el cual autores como Piaget (1970) y Vygotsky (1978) afirman que el aprendizaje es más eficaz cuando los estudiantes participan activamente en la construcción del conocimiento mediante la interacción con el entorno y con otros. En este marco, la implementación del método de bisección mediante ejercicios programables en MATLAB permite estructurar actividades que no solo enseñan una técnica, sino que sitúan al estudiante en un proceso activo de resolución de problemas de tipo computacional. Esta forma de enseñanza responde a la necesidad de superar las limitaciones de la educación tradicional, la cual raramente promueve el desarrollo de habilidades, competencias y capacidades clave. Por tanto, se propone una transformación en la concepción del proceso de enseñanza-aprendizaje, que no excluye la clase expositiva, pero sí la complementa con estrategias orientadas al desarrollo integral del estudiante (Morales et al., 2004).

En esta línea, el Aprendizaje Basado en Problemas (ABP) se presenta como una metodología que favorece la participación activa y autónoma de los estudiantes, colocando la resolución de problemas como eje central del proceso formativo. Según Guamán y Espinoza (2022), el ABP se caracteriza por el protagonismo del estudiante, el rol del docente como

facilitador, y el trabajo colaborativo en equipos pequeños. Además, esta metodología desarrolla habilidades investigativas a través de la búsqueda de información, y fomenta la responsabilidad tanto individual como colectiva en el logro de objetivos. El ABP también tiene un enfoque interdisciplinar, ya que los problemas tratados suelen requerir la integración de conocimientos de diversas áreas. Aplicado al contexto de ejercicios innovadores en MATLAB, el ABP permite presentar estos desafíos como problemas abiertos, en los que los estudiantes deben comprender conceptos matemáticos, traducirlos a código, interpretar los resultados y corregir errores. De este modo, se promueve una comprensión más profunda del método de bisección, trascendiendo su simple aplicación mecánica y fortaleciendo la conexión entre la Matemática y su implementación computacional.

Marco Metodológico

Esta investigación se enmarca en un enfoque cualitativo de tipo investigación acción educativa, en tanto busca mejorar la práctica docente a través de una intervención pedagógica diseñada, aplicada y analizada en el contexto real de enseñanza del curso de Métodos Numéricos. Se trata de una experiencia didáctica implementada por un docente con amplia trayectoria en la enseñanza de esta asignatura, cuyo objetivo fue valorar el impacto de actividades basadas en programación sobre la comprensión del método de bisección por parte del estudiantado. La experiencia se desarrolló en el curso de Métodos Numéricos del BLEM en la UNA, durante el primer semestre del año 2024. Participaron nueve estudiantes del curso, quienes trabajaron individualmente durante una clase presencial de 90 minutos. Cada estudiante fue asignado a una de las tres actividades propuestas, de modo que tres personas realizaron la Actividad 1, tres la Actividad 2 y tres la Actividad 3.

Previo a la aplicación de las actividades, se desarrolló en clase una introducción teórico-práctica del método de bisección, en la que el docente presentó la definición formal del método, sus condiciones de aplicabilidad, y su fundamento matemático como proceso iterativo. A continuación, y mediante trabajo conjunto con el estudiantado, se llevó a cabo la programación paso a paso del método en MATLAB, de manera que se discutiera el papel de cada línea de código en relación con el procedimiento numérico. Esta fase tuvo como propósito promover una comprensión conceptual del algoritmo más allá de su simple ejecución, articulando el razonamiento matemático con la lógica computacional. Luego de este trabajo introductorio, se procedió a la aplicación de tres actividades originales diseñadas para profundizar en la comprensión del método de bisección. Estas actividades requerían que el estudiante modificara o adaptara la programación previamente vista, para resolver situaciones específicas que ponían a prueba su dominio tanto del funcionamiento del método como de su implementación computacional. Los problemas propuestos incluían análisis de convergencia, identificación de errores comunes y exploración de condiciones límite del algoritmo.

La propuesta metodológica combina componentes didácticos clave para fomentar un aprendizaje activo y significativo del método de bisección, utilizando la programación como herramienta para visualizar y experimentar con el algoritmo, e integrando representaciones múltiples para enriquecer la comprensión. Las actividades, diseñadas como situaciones abiertas que promueven la interpretación, el análisis crítico y la conexión entre teoría y práctica, también buscan desarrollar pensamiento computacional y matemático, incluso en estudiantes sin

experiencia previa en programación, promoviendo además la metacognición y el aprendizaje autorregulado.

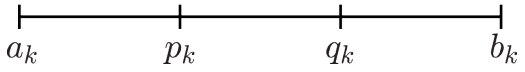
Actividad 1

Recuerde que el método de bisección para aproximar una solución de $f(x) = 0$ corresponde a: dados a_0 y b_0 tales que $f(a_0)f(b_0) \leq 0$, se tiene que:

$$x_k \coloneqq \frac{a_k + b_k}{2} \quad y \quad [a_{k+1}, b_{k+1}] \coloneqq \begin{cases} [a_k, x_k] & \text{si } f(a_k)f(x_k) \leq 0 \\ [x_k, b_k] & \text{si } f(a_k)f(x_k) > 0 \end{cases}$$

para todo $k \in IN$.

Ahora, considere la siguiente variante, la cual conoceremos como el **método de tribisección**: Sean a_0 y b_0 tales que $f(a_0)f(b_0) \leq 0$. Dividimos, en cada iteración, el intervalo $[a_k, b_k]$ en tres partes iguales:



donde:

$$p_k \coloneqq \frac{2a_k + b_k}{3}, \quad q_k \coloneqq \frac{a_k + 2b_k}{3}, \quad [a_{k+1}, b_{k+1}] \coloneqq \begin{cases} [a_k, p_k] & \text{si } f(a_k)f(p_k) \leq 0 \\ [p_k, q_k] & \text{si } f(p_k)f(q_k) \leq 0 \\ [q_k, b_k] & \text{en otro caso} \end{cases}$$

y así la nueva aproximación x_k está dada por:

$$x_k \coloneqq \frac{a_{k+1} + b_{k+1}}{2} \quad \forall\, k \in IN.$$

a) Realice 4 iteraciones del método de tribisección para aproximar una solución de $2x - \cos(x) - \operatorname{sen}(x) = 0$ en $[0, 2]$. $(\star)$
b) Anote las semejanzas y diferencias que puede notar entre este nuevo método con la iteración de bisección tradicional.
c) Describa cuál o cuáles líneas de la implementación de bisección realizada en clase, deben modificarse para implementar en MATLAB la nueva variante del método de tribisección.
d) Implemente en MATLAB una función de nombre:

[x, k] = tribiseccion(f, a, b, tol, iterMax)

la cual lleve a cabo la iteración de tribisección para aproximar un cero de f en $[a, b]$.
e) Utilice su implementación para aproximar una solución de $(\star)$ con una tolerancia de 10^{-8} y un número máximo de 100 iteraciones. Responda: Aproximación obtenida: _____. Iteraciones usadas: _____.

Actividad 2

Esta es similar a la Actividad 1, denominado el **método de tribisecciónPF**, pero ahora las definiciones de p_k, q_k y del nuevo intervalo $[a_{k+1}, b_{k+1}]$ son:

$$p_k \coloneqq a_k + \lambda(b_k - a_k), \quad q_k \coloneqq b_k - \lambda(b_k - a_k), \quad [a_{k+1}, b_{k+1}]$$
$$\coloneqq \begin{cases} [a_k, p_k] & \text{si } f(a_k)f(p_k) \leq 0 \\ [p_k, q_k] & \text{si } f(p_k)f(q_k) \leq 0 \\ [q_k, b_k] & \text{en otro caso} \end{cases}$$

para $\lambda \in \left]0, \frac{1}{2}\right[$. Luego, las cinco preguntas a responder son las mimas de la Actividad 1.

Actividad 3

La tercera actividad es similar a la Actividad 1, pero donde la nueva aproximación x_k se calcula de la forma:

$$x_k := \begin{cases} p_k & \text{si el cambio de signo ocurre en } [a_k, p_k]\,, \\ p_k & \text{si el cambio de signo ocurre en } [p_k, q_k] \quad \text{y} \quad |f(p_k)| \leq |f(q_k)| \\ q_k & \text{si el cambio de signo ocurre en } [p_k, q_k] \quad \text{y} \quad |f(q_k)| \leq |f(p_k)| \\ q_k & \text{si el cambio de signo ocurre en } [q_k, b_k]\,, \end{cases}$$

para todo $k \in IN$, donde es claro que el nuevo intervalo $[a_{k+1}, b_{k+1}]$ es aquel que preserva el cambio de signo. A esta variante se le conocerá como el **método de trisección**.

Luego, las cinco preguntas a responder son las mimas de la Actividad 1, excepto en el punto a), donde la ecuación ahora es: $e^x + \csc(x) - 7 = 0$ en $[1, 2]$.

Resultados

La implementación de las actividades con los nueve participantes del curso permitió identificar resultados relevantes en términos de comprensión conceptual y desempeño en programación del método de bisección. De los nueve estudiantes, siete lograron completar exitosamente la actividad asignada, mientras que dos encontraron dificultades relacionadas con la interpretación de instrucciones matemáticas o con la gestión del tiempo para completar la codificación. En particular, uno de los casos correspondió a un estudiante que no logró comprender la notación matemática en la Actividad 2, y el otro, asignado a la Actividad 3, aunque logró entender el procedimiento, no alcanzó a finalizar la implementación por falta de tiempo.

Entre los estudiantes que finalizaron exitosamente, se observó un proceso consistente: dedicaron la mayor parte del tiempo a comprender el método de forma analítica antes de proceder a la codificación. El trabajo previo con papel y lápiz, así como el uso constante del dibujo del intervalo trisecado, funcionó como un apoyo visual importante para facilitar la lógica del algoritmo. Esta estrategia evidencia una apropiación del enfoque de resolución de problemas como medio para construir conocimiento matemático (Schoenfeld, 1992) y confirma la importancia de representaciones múltiples en la comprensión de procesos computacionales complejos. A continuación, se presenta en la Figura 1 la solución de un estudiante que resolvió la Actividad 1, ilustrando este enfoque.

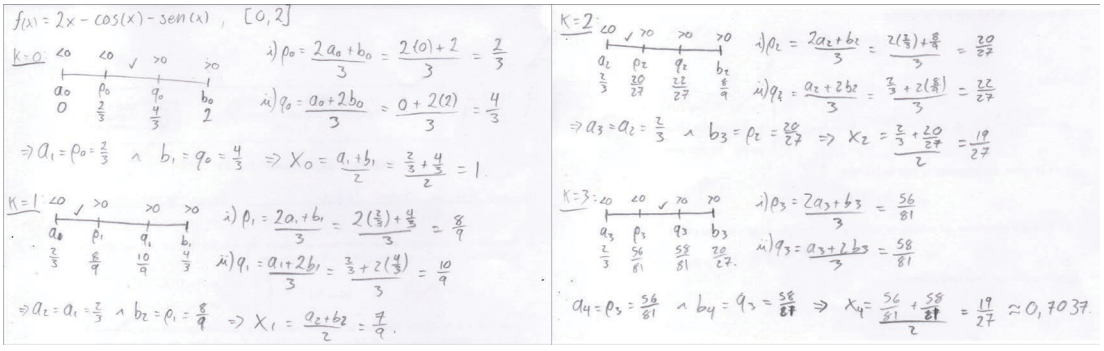
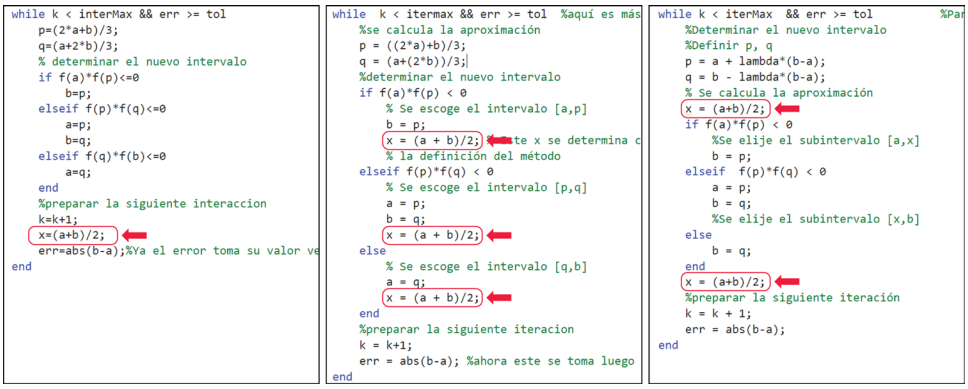


Figura 1. Solución propuesta por un estudiante de la actividad 1.

En la programación con MATLAB, los estudiantes mostraron habilidad para modificar el código base del método de bisección, especialmente en la definición de subintervalos y el cálculo de la nueva aproximación x_k. Sin embargo, se observó variabilidad en la ubicación de este cálculo dentro del código, en particular en las Actividades 1 y 2, lo cual se atribuye a un cambio conceptual importante: a diferencia del algoritmo tradicional, las variantes empleadas requerían seleccionar primero el subintervalo y luego calcular la nueva aproximación, lo que resultó más complejo de asimilar para algunos estudiantes.

Por ejemplo, en la siguiente Figura 2 se muestra parte de la programación realizada por tres estudiantes: las dos primeras corresponden a la Actividad 1 y la última a la Actividad 2. Todas son correctas, pero resulta innecesario contar con más de un cálculo para x_k, como ocurre en los dos últimos ejemplos de la figura.



```
while k < interMax && err >= tol
    p=(2*a+b)/3;
    q=(a+2*b)/3;
    % determinar el nuevo intervalo
    if f(a)*f(p)<=0
        b=p;
    elseif f(p)*f(q)<=0
        a=p;
        b=q;
    elseif f(q)*f(b)<=0
        a=q;
    end
    %preparar la siguiente interaccion
    k=k+1;
    x=(a+b)/2;
    err=abs(b-a);%Ya el error toma su valor ve
end
```

```
while  k < itermax && err >= tol  %aquí es más
    %se calcula la aproximación
    p = ((2*a)+b)/3;
    q = (a+(2*b))/3;
    %determinar el nuevo intervalo
    if f(a)*f(p) < 0
        % Se escoge el intervalo [a,p]
        b = p;
        x = (a + b)/2;  %este x se determina c
        % la definición del método
    elseif f(p)*f(q) < 0
        % Se escoge el intervalo [p,q]
        a = p;
        b = q;
        x = (a + b)/2;
    else
        % Se escoge el intervalo [q,b]
        a = q;
        x = (a + b)/2;
    end
    %preparar la siguiente iteracion
    k = k+1;
    err = abs(b-a); %ahora este se toma luego
end
```

```
while k < iterMax  && err >= tol          %Par
    %Determinar el nuevo intervalo
    %Definir p, q
    p = a + lambda*(b-a);
    q = b - lambda*(b-a);
    % Se calcula la aproximación
    x = (a+b)/2;
    if f(a)*f(p) < 0
        %Se elije el subintervalo [a,x]
        b = p;
    elseif  f(p)*f(q) < 0
        a = p;
        b = q;
        %Se elije el subintervalo [x,b]
    else
        b = q;
    end
    x = (a+b)/2;
    %preparar la siguiente iteración
    k = k + 1;
    err = abs(b-a);
end
```

Figura 2. Solución propuesta por dos estudiantes de la actividad 1 y otro estudiante de la actividad 2, respectivamente.

Estas observaciones revelan que, aunque la lógica subyacente al algoritmo fue comprendida, la adaptación estructural del código presentó cierto grado de dificultad, lo que valida la premisa de que traducir el lenguaje matemático al lenguaje computacional implica un reto significativo para estudiantes sin experiencia previa en programación (Londoño, 2019; Morales & Landa, 2004). Aun así, las producciones estudiantiles evidencian avances importantes en la comprensión del método, el razonamiento computacional y la capacidad de análisis crítico frente a errores.

Conclusiones

La experiencia descrita en este estudio pone en evidencia que, aunque los textos clásicos sobre métodos numéricos explican el funcionamiento del método de bisección (Burden et al., 2015; Chapra, 2008), rara vez ofrecen actividades prácticas de programación que permitan a los estudiantes aplicar y profundizar en el contenido. En este sentido, las tres actividades propuestas se consolidan como una herramienta didáctica eficaz, ya que lograron comprometer a los estudiantes con el proceso de implementación algorítmica, incluso en un contexto donde no se dispone de formación previa en programación.

Los hallazgos confirman que la programación en MATLAB, utilizada como medio de aprendizaje activo y no como fin, permite a los estudiantes visualizar, manipular y experimentar

con el algoritmo, favoreciendo una comprensión más profunda del método. Además, el trabajo con representaciones múltiples y la resolución de problemas abiertos fomentaron el pensamiento computacional, el análisis crítico y la reflexión sobre el error, todos elementos que fortalecen la metacognición y el aprendizaje autorregulado (Guamán & Espinoza, 2022). Aunque se reconocen las dificultades iniciales al traducir conceptos matemáticos a código, los resultados muestran que, una vez superado ese umbral, la programación se convierte en una herramienta accesible y poderosa para el aprendizaje. En consecuencia, se evidencia la necesidad de integrar este tipo de propuestas en la enseñanza de los métodos numéricos, no solo para mejorar el dominio técnico, sino también para desarrollar habilidades clave en el contexto profesional actual, alineadas con enfoques como el Aprendizaje Basado en Problemas (ABP) y las teorías constructivistas del aprendizaje (Piaget, 1970; Vygotsky, 1978).

Referencias y bibliografía

Burden, R. L., Faires, J. D., Burden, A. M. (2015). *Numerical Analysis*, 10th. United States: Cengage Learning.

Chapra, S. C. (2008). *Applied Numerical Methods with MATLAB for Engineers and Scientists*. Spain: McGraw-Hill Companies,Incorporated.

Guamán Gómez, V. J., & Espinoza Freire, E. E. (2022). Aprendizaje basado en problemas para el proceso de enseñanzaaprendizaje. Revista Universidad y Sociedad, 14(2), 124-131.

Londoño, D. A. F. (2019). Programación Científica: Una Propuesta Didáctica para la Enseñanza de Métodos Numéricos y Programación. *Encuentro Internacional de Educación en Ingeniería*.

Morales, P., & Landa, V. (2004). Aprendizaje basado en problemas. *Theoria*, *13*(1), 145-157.

Montero, Y. H., Pedroza, M. E., Astiz, M. S. y Vilanova, S. L. (2015). Caracterización de las actitudes de estudiantes universitarios de Matemática hacia los métodos numéricos. *Revista Electrónica de Investigación Educativa*, 17(1), 88-99. Recuperado de http://redie.uabc.mx/vol17no1/contenido-montero-etal.html

Schoenfeld, A. H. (1992). On paradigms and methods: What do you do when the ones you know don't do what you want them to? Issues in the analysis of data in the form of videotapes. *The Journal of the Learning Sciences*, *2*(2), 179-214.

Süli, E., Mayers, D. F. (2003). *An Introduction to Numerical Analysis*. United Kingdom: Cambridge University Press.

Piaget, J. (1970). *The Science of Education and the Psychology of the Child*. Viking Press.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press.

Yang, W. Y., Cao, W., Chung, T., Morris, J. (2005). *Applied Numerical Methods Using MATLAB*. Germany: Wiley.