Universidad Mayor de San Andrés
Facultad de Ciencias Puras y Naturales
Carrera de Informática

Proyecto Final
Simulador de Convergencia de Metodos Iterativos

Curso: Metodos Numericos I

Docente: Carlos Mullisaca Choque

Integrantes:

- Ian Ezequiel Salinas Condori
- Marco Gabriel Mamani Laura
- Cristhian Manuel Surco Quisbert

5 de diciembre de 2025

Índice

1. Introducción

El presente trabajo tiene como finalidad desarrollar una aplicación interactiva denominada *Los Convergentes*, cuyo propósito es comparar el comportamiento de distintos métodos iterativos para la búsqueda de raíces de ecuaciones no lineales.

Los métodos iterativos son fundamentales en el análisis numérico porque permiten aproximar soluciones cuando no existe una fórmula cerrada o cuando resolver una ecuación analíticamente resulta demasiado complejo. En numerosas aplicaciones de la física, la ingeniería, la economía y la informática, aparecen ecuaciones del tipo

$$f(x) = 0$$

para las cuales no es posible obtener la solución exacta de manera algebraica.

La aplicación desarrollada utiliza Python junto con las librerías *Streamlit*, *NumPy* y *Matplotlib*, y proporciona una interfaz gráfica que permite analizar:

- La rapidez de convergencia de cada método.
- El comportamiento del error en función del número de iteraciones.
- El orden de convergencia estimado.
- La eficiencia computacional.
- La robustez frente a diferentes condiciones iniciales.

En este informe se describen los fundamentos teóricos de los métodos utilizados, el diseño de la aplicación, la metodología de pruebas, los resultados obtenidos y las conclusiones principales del estudio.

2. Objetivos

2.1. Objetivo general

Desarrollar una herramienta interactiva que permita comparar la convergencia de cinco métodos iterativos clásicos para el cálculo de raíces de ecuaciones no lineales: Punto Fijo, Bisección, Regla Falsa, Newton–Raphson y Secante, mediante visualizaciones, tablas y análisis numéricos.

2.2. Objetivos específicos

- Implementar correctamente los algoritmos de punto fijo, bisección, regla falsa, Newton–Raphson y secante.
- Visualizar el comportamiento del error en función del número de iteraciones para cada método.
- Estimar y comparar el orden de convergencia aproximado de los métodos.
- Analizar la eficiencia computacional mediante la medición del tiempo de ejecución.
- Validar los resultados utilizando funciones de prueba con raíces conocidas.
- Diseñar una interfaz gráfica intuitiva y fácil de usar para fines académicos.

3. Marco teórico

3.1. Métodos iterativos

Los métodos iterativos para la resolución de ecuaciones no lineales consisten en la construcción de una sucesión

$$\{x_n\}_{n=0}^{\infty}, \quad x_0, x_1, x_2, \ldots$$

que, bajo ciertas condiciones, converge hacia una raíz r de la ecuación

$$f(x) = 0.$$

Es decir, se busca que

$$\lim_{n\to\infty} x_n = r, \quad \text{tal que } f(r) = 0.$$

Estos métodos permiten aproximar soluciones numéricas con una tolerancia previamente definida, y son especialmente útiles cuando la ecuación no puede resolverse de forma exacta o cuando la solución analítica es demasiado costosa de obtener.

3.2. Convergencia y error

Se dice que un método iterativo converge si la sucesión $\{x_n\}$ generada por el algoritmo se aproxima a la raíz r. El error absoluto en la iteración n puede medirse como

$$e_n = |r - x_n|$$

o, en la práctica, a través de

$$e_n = |f(x_n)|.$$

En la implementación presentada se emplea principalmente el error $|f(x_n)|$ como criterio para decidir si la aproximación es suficientemente buena, es decir, si

$$|f(x_n)| < \text{tol},$$

donde *tol* es la tolerancia especificada por el usuario.

3.3. Orden de convergencia

El orden de convergencia describe la rapidez con la que disminuye el error en cada iteración. Formalmente, un método tiene orden de convergencia p si se cumple que

$$e_{n+1} \approx C e_n^p,$$

donde C es una constante positiva y e_n es el error en la iteración n.

En la práctica, el orden de convergencia se puede estimar a partir de los errores sucesivos mediante relaciones logarítmicas. Valores típicos son:

- Bisección: $p \approx 1$ (convergencia lineal).
- Regla Falsa: $p \approx 1$ (también lineal).
- Secante: $p \approx 1{,}618$ (convergencia superlineal).
- Newton–Raphson: $p \approx 2$ (convergencia cuadrática).

3.4. Descripción matemática de los métodos

A continuación se describen brevemente los métodos iterativos implementados.

3.4.1. Método de Punto Fijo

El método de punto fijo reescribe la ecuación $f(x) = 0$ en la forma

$$x = g(x),$$

y genera la sucesión

$$x_{n+1} = g(x_n).$$

Bajo condiciones adecuadas sobre g (por ejemplo, que sea contractiva en un intervalo), la sucesión converge hacia un punto fijo r tal que $r = g(r)$, que es una raíz de f.

3.4.2. Método de Bisección

El método de bisección parte de un intervalo $[a, b]$ tal que $f(a)f(b) < 0$, garantizando la existencia de al menos una raíz en dicho intervalo (por el teorema del valor intermedio). En cada iteración se calcula el punto medio

$$m = \frac{a + b}{2}$$

y se evalúa $f(m)$, seleccionando el subintervalo donde cambia de signo la función. El proceso se repite hasta que la longitud del intervalo o el valor de $|f(m)|$ sea menor que la tolerancia.

3.4.3. Método de Regla Falsa

La regla falsa (false position) también utiliza un intervalo $[a, b]$ con $f(a)f(b) < 0$, pero en lugar de usar el punto medio, emplea la intersección de la recta secante con el eje x:

$$x = b - f(b)\frac{b - a}{f(b) - f(a)}.$$

Luego se actualiza el intervalo de forma similar a la bisección, manteniendo siempre la condición de cambio de signo.

3.4.4. Método de Newton–Raphson

El método de Newton–Raphson utiliza la información de la derivada de f. A partir de x_n, se aproxima la función localmente por la recta tangente y se calcula:

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

Este método suele presentar convergencia cuadrática cerca de la raíz, pero requiere que la derivada no se anule y que la aproximación inicial sea adecuada.

3.4.5. Método de la Secante

El método de la secante es similar a Newton, pero no requiere derivada explícita. Utiliza dos aproximaciones anteriores x_{n-1} y x_n y define:

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

La convergencia suele ser superlineal, con un orden aproximado de $p \approx 1{,}618$.

4. Descripción del sistema desarrollado

4.1. Tecnologías utilizadas

La aplicación *Los Convergentes* fue desarrollada en el lenguaje Python, utilizando las siguientes librerías:

- **Streamlit**: para la construcción de la interfaz gráfica web.
- **NumPy**: para operaciones numéricas.
- **Matplotlib**: para la generación de gráficos.
- **Pandas**: para la organización tabular de resultados.

Estas herramientas son recomendadas para el desarrollo de aplicaciones numéricas interactivas debido a su facilidad de uso y amplia documentación.

4.2. Estructura general de la aplicación

El código principal se encuentra en el archivo `app.py`. En él se definen:

- Las funciones correspondientes a cada método iterativo.
- Una estructura de datos para almacenar las iteraciones, errores, tiempo de ejecución y orden de convergencia.
- La interfaz gráfica con Streamlit, que incluye:
  - Un panel lateral para la configuración de parámetros.
  - Un panel central con tablas y gráficos.

4.3. Interfaz gráfica

La interfaz permite al usuario:

- Seleccionar la función de prueba.
- Definir la tolerancia y el máximo de iteraciones.
- Ingresar intervalos iniciales y valores iniciales x_0, x_1.
- Activar o desactivar los métodos a comparar.
- Visualizar tablas de resultados y gráficos de convergencia.

5. Metodología y casos de estudio

5.1. Funciones de prueba

Para evaluar el comportamiento de los métodos, se consideraron funciones de prueba clásicas, tales como:

- $f(x) = x^2 - 2$
- $f(x) = x^3 - x - 1$
- $f(x) = \cos(x) - x$

Estas funciones tienen raíces conocidas o ampliamente estudiadas, lo que permite validar las aproximaciones obtenidas por los métodos iterativos.

5.2. Parámetros iniciales

Para cada función se definieron:

- Intervalos $[a, b]$ adecuados para bisección y regla falsa.
- Valores iniciales x_0 y x_1 para Newton y secante.
- Tolerancias típicas de 10^{-6} y 10^{-8}.

6. Resultados y análisis

6.1. Error vs número de iteraciones

Los gráficos generados por la aplicación muestran el error $|f(x_n)|$ en escala logarítmica frente al número de iteraciones. Se observa que:

- El método de Newton–Raphson converge en pocas iteraciones.
- El método de la secante presenta un comportamiento intermedio en cuanto a rapidez.
- Los métodos de bisección y regla falsa requieren más iteraciones, pero son más estables.

6.2. Órdenes de convergencia estimados

A partir de los errores se calculó un orden de convergencia aproximado para cada método. Los valores obtenidos coinciden con la teoría:

- Bisección y Regla Falsa con orden cercano a 1.
- Secante con orden aproximado $p \approx 1{,}6$.
- Newton–Raphson con orden cercano a 2.

6.3. Eficiencia computacional

La aplicación mide el tiempo de ejecución de cada método en milisegundos. En general, se comprobó que:

- Newton–Raphson y Secante son más eficientes en términos de tiempo y número de iteraciones.
- Bisección y Regla Falsa consumen más iteraciones, pero mantienen una robustez alta.

6.4. Comparación global

Cada método presenta ventajas y desventajas:

- **Bisección**: siempre converge si $f(a)f(b) < 0$, pero es lento.
- **Regla Falsa**: mejora parcialmente la rapidez de bisección, manteniendo el cambio de signo.
- **Newton–Raphson**: muy rápido, pero requiere derivada y una buena aproximación inicial.
- **Secante**: no requiere derivada y mantiene buena rapidez de convergencia.
- **Punto Fijo**: depende fuertemente de la elección de $g(x)$ y puede divergir.

7. Validación

7.1. Comparación con valores exactos

En el caso de la función $f(x) = x^2 - 2$, la raíz exacta es $\sqrt{2} \approx 1{,}414213562$. Las aproximaciones obtenidas por los distintos métodos coinciden con este valor dentro de la tolerancia definida, confirmando la correcta implementación.

7.2. Discusión sobre estabilidad y robustez

Los resultados muestran que:

- Bisección y Regla Falsa son muy robustos, pero relativamente lentos.
- Newton–Raphson y Secante son métodos rápidos, pero pueden fallar si las condiciones iniciales no son adecuadas.
- El método de Punto Fijo es útil para fines didácticos, pero su comportamiento depende fuertemente de la función $g(x)$.

8. Conclusiones

En este trabajo se desarrolló la aplicación *Los Convergentes*, que permite comparar de forma interactiva distintos métodos iterativos para el cálculo de raíces de ecuaciones no lineales. La herramienta facilita la visualización del error, del orden de convergencia y de la eficiencia computacional de cada método.

Los resultados obtenidos son coherentes con la teoría del análisis numérico: Newton–Raphson y Secante presentan las tasas de convergencia más altas, mientras que Bisección y Regla Falsa ofrecen mayor estabilidad ante diferentes condiciones iniciales. El método de Punto Fijo, por su parte, ilustra la importancia de la elección adecuada de la función de iteración $g(x)$.

La aplicación constituye un recurso didáctico útil para comprender el comportamiento de los métodos iterativos y puede ampliarse en trabajos futuros incorporando nuevos métodos, permitiendo la definición libre de funciones por parte del usuario, o extendiéndose a sistemas de ecuaciones no lineales.

9. Bibliografía

- Burden, R. L., & Faires, J. D. (2011). *Análisis Numérico*. Cengage Learning.
- Chapra, S. C., & Canale, R. P. (2015). *Métodos Numéricos para Ingenieros*. McGraw-Hill.
- Sauer, T. (2012). *Numerical Analysis*. Pearson.

Anexos

En esta sección pueden incluirse capturas de pantalla de la aplicación, tablas detalladas de iteraciones y configuraciones específicas utilizadas durante las pruebas.