

Taller: Comparación experimental de métodos de búsqueda de raíces (1D) con GSL

Alexander Pérez Ruiz

23 de septiembre de 2025

1. Objetivos

- Comparar en la práctica los métodos de búsqueda de raíces más usados (bisección, falsa posición, secante, Brent, Newton) en términos de:
 - convergencia (velocidad y órdenes)
 - robustez (sensibilidad a la elección inicial / bracketing)
 - número de evaluaciones de f y f'
 - estabilidad numérica
 - tiempo de cómputo en condiciones realistas.
- Desarrollar habilidad para instrumentar experimentos numéricos (medir, graficar, interpretar).
- Formular conclusiones cualitativas y cuantitativas que no puedan obtenerse únicamente pidiendo respuestas a una IA (por ejemplo: comportamiento en hardware limitado, variabilidad estadística entre ejecuciones con ruido en parámetros, intuiciones sobre cuándo preferir un método en ingeniería real).

2. Funciones a evaluar

1. $f_1(x) = x^3 - 6x^2 + 11x - 6$
2. $f_2(x) = (x - 1)^2(x - 2)$
3. $f_3(x) = \tanh(10(x - 0,1)) - 0,5$
4. $f_4(x) = \sin(50x)/x$
5. $f_5(x) = |x - 0,5| - 0,1$

3. Experimentos

Para cada función y cada método:

- Preparar: elegir intervalos/brackets o puntos iniciales razonables (documentar elección).
- Criterios de parada: tolerancia en $|f(x)| < \text{tol}$ y $|\Delta x| < \text{tol}$. Registrar ambos.
- Medir:
 - número de iteraciones,
 - número de evaluaciones de f y de f' ,
 - tiempo de CPU,
 - si converge o no (y en cuántas iteraciones máximas — por ejemplo 200).
- Variación de inicialización: repetir con varias semillas de inicialización.

4. Preguntas orientadoras

- Para cada función, ¿qué método recomiendan en producción (por fiabilidad y coste) y por qué? (fundamentar con datos experimentales).
- ¿Dónde observan discrepancias entre la teoría (orden de convergencia) y los resultados prácticos? Expliquen las razones (redondeo, límites de representación, evaluaciones finitas, mala inicialización).
- ¿Qué fallos o inconsistencias inesperadas encontraron (p. ej. saltos, NaNs, convergencias a raíces falsas) y cómo los diagnosticarías y arreglarías en código de producción?
- ¿Cómo afecta el costo de evaluar f (p. ej. si f es una simulación costosa) la elección del método? Diseñen una regla práctica para elegir el método en función del costo de evaluación relativa de f y f' .
- ¿Qué precauciones adicionales implementarían en una librería real (logging, límites de iteración, backoffs, switches entre métodos)? Proponer un pseudocódigo para un root-finder robusto".
- Si el hardware es limitado (p. ej. microcontrolador con punto flotante simple), ¿qué métodos preferirían y por qué? Comparar con resultados en PC.
- Describe un experimento de "falsificación" que mostraría que un resultado empírico tuyo no es un artefacto de una mala instrumentación o azar.