



UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERÍA

2DO CUATRIMESTRE DE 2022

ANÁLISIS NUMÉRICO

---

## Búsqueda de raíces

---

**Curso:**

Sassano

**Integrantes:**

integrante 1	mail	padrón
integrante 2	mail	padrón
integrante 3	mail	padrón
integrante 4	mail	padrón

**Lenguaje Elegido:** lenguaje

## 1. Enunciado

### 1. No usar fracciones

Durante la clase y en instancias de evaluación se solicita al alumno que no utilice fracciones como expresión de un número.

Esta vez vamos a solicitarle al alumno que encuentre cuales son los valores que puede tomar al representarlo a través de la herramienta con la que estamos trabajando.

Teniendo en cuenta que  $total = \sum últimoNúmeroPadrón(Integrante_i)$ , se solicita lo siguiente:

- (a) Realizar la cuenta :  $\frac{1}{total}$  y almacenar el valor representando en formato Half (16 bits), Single (32 bits) y Double (64 bits) y expresar cual es el siguiente y el anterior valor posible.
- (b) Al momento de calcular la propagación de errores ¿Debemos tener en cuenta que cada fracción es una variable más?
- (c) Si se nos preguntara si una respuesta dada por una herramienta fue exacta para cualquier calculo ¿Qué deberíamos tener en cuenta?

## 2. Resolución polinomio de segundo orden

En el mundo de la matemática la resolución del polinomio de segundo orden esta definido de manera univoca, es decir, si realizamos las cuentas utilizando una expresión matemáticamente equivalente, obtenemos el mismo resultado.

Sin embargo en esta ocasión queremos conocer los límites de las herramientas que utilizamos día a día para hacer las cuentas.

Por ese motivo se pide realizar los siguientes programas, que reciban como parámetro los valores  $a$ ,  $b$  y  $c$  correspondientes a los terminos de un polinomio de orden 2 y además un parámetro adicional para saber de que tipo de representación de punto flotante se está utilizando (Half [16], Single [32] o Double [64]).

$$a \cdot x^2 + b \cdot x + c \quad (1)$$

- (a) Programar un algoritmo para hallar las raíces reales de un polinomio de orden dos de forma convencional (sin lógica adicional).
- (b) Programar un algoritmo para hallar las raíces reales de un polinomio de orden dos de forma dinámica en respuesta al valor  $b$ , teniendo en cuenta su parámetro adicional para definir el criterio por el cual considera que expresión matemática utilizar.
- (c) Realizar pruebas para los primeros dos items donde los coeficientes  $a$ ,  $b$  y  $c$  del polinomio tengan mucha diferencia en sus ordenes de magnitud.
- (d) Comparar los resultados obtenidos de las dos formas de programación.

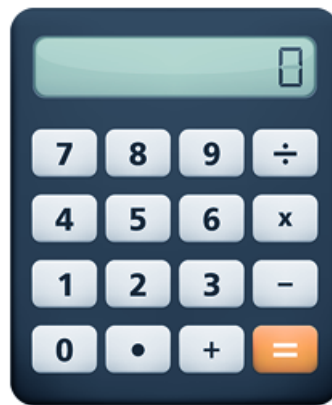


Figura 1:

### 3. Búsqueda de raíces

Para las siguientes funciones continuas y con raíz única en el intervalo  $[0, 2]$ :

$$f_1(x) = x^2 - 2 \quad (2)$$

$$f_2(x) = x^5 - 6.6x^4 + 5.12x^3 + 21.312x^2 - 38.016x + 17.28$$

$$f_3(x) = (x - 1.5)e^{-4(x-1.5)^2}$$

se pide:

- (a) Graficar las funciones en el intervalo de interés
- (b) Halle para cada una de ellas la raíz en el intervalo indicado mediante los métodos vistos en clase:
  - Bisección
  - Punto Fijo
  - Secante
  - Newton-Raphson
  - Newton-Raphson modificado

Use para todos los métodos como criterio de parada las siguientes diferencia entre dos iteraciones sucesivas:

- $1 \cdot 10^{-5}$
- $1 \cdot 10^{-13}$

para Newton-Raphson use semilla  $x_0 = 1$ .

Mostrar en una tabla por cada método los resultados obtenidos (en caso que se obtengan muchas iteraciones se pueden mostrar las primeras 5 y las últimas 5).

- (c) Halle la raíz mediante la función de búsqueda de raíces de un lenguaje o paquete orientado a cálculo numérico (e.g. Python+SciPy: `scipy.optimize.brentq`). Al resultado obtenido le vamos a asignar como " $x_{Real}$ "<sup>1</sup>.
- (d) Compare los resultados obtenidos para los distintos métodos y cotas de error, grafique para todos los casos:
  - Orden de convergencia  $P$  vs iteraciones
  - Constante asintótica  $\lambda$  vs iteraciones
  - $\log_{10}(/ \Delta x /)$  vs iteraciones
  - $\log_{10}(/ x_{Candidata} - x_{Real} /)$  vs iteraciones

Discuta ventajas y desventajas. ¿Son las que esperaba en base a la teoría?

### Ayuda

Si necesita derivadas:

$$f_1'(x) = 2x$$

$$f_1''(x) = 2$$

$$f_2'(x) = 5x^4 - 26.4x^3 + 15.36x^2 + 42.624x - 38.016$$

$$f_2''(x) = 20x^3 - 79.2x^2 + 30.72x + 42.624$$

$$f_3'(x) = (-8x + 12.0)(x - 1.5)e^{-4(x-1.5)^2} + e^{-4(x-1.5)^2}$$

$$f_3''(x) = (-24x + (x - 1.5)(8x - 12.0)^2 + 36.0)e^{-4(x-1.5)^2}$$

---

<sup>1</sup>si bien sabemos que no es cierto, simplemente lo tomamos para trabajar en los graficos posteriores

## 2. Especificación de formato de informe y entrega

- El informe técnico no debe exceder las 8 hojas.
- Debe seguir las especificaciones de informes del curso. Las mismas las encontrará en el campus, en la sección “Especificación de informes”. Antes de entregar el trabajo práctico, tenga a bien de verificar el cumplimiento de la guía y formato especificados en dicho documento.
- El día de entrega, debe adjuntar en la sección correspondiente del campus un archivo comprimido en formato ZIP conteniendo el informe en formato PDF y una carpeta con los scripts de código necesarios para la verificación por parte de los docentes de los resultados del trabajo. En el caso que el trabajo se haya realizado en Colab o una notebook Python, se deberá entregar el archivo en formato IPYNB.
- El nombre del archivo ZIP o IPYNB a cargar en el campus debe ser de la forma "TPx\_grupo.z", donde x es el número de TP y z es el número de grupo. Por favor siga este formato para facilitarle a los docentes la descarga de los trabajos.
- Respecto a los resultados obtenidos, el informe debe indicar al lector qué y cómo ejecutar los archivos de código fuente para reproducir los mismos resultados que se muestren en el documento.
- El no cumplimiento de lo especificado en esta sección puede ser razón de correcciones del informe.

## Referencias

- [1] Cheney, W.; Kincaid, D. *Numerical Mathematics and Computing*. 6ta ed. EE.UU.: Thomson Brooks/Cole, 2008.
- [2] Burden, R. L.; Faires, J.D. *Análisis Numérico*. 2da ed. México: Iberoamérica, 1996.