

Taller 1 - Análisis Numérico

Juan Camilo Chafloque

Abel Santiago Cortes

Juan Sebastian Osorio

6 de agosto de 2019

1. Problema 1: Error de Redondeo

Los métodos numéricos operan con datos que pueden ser inexactos y con dispositivos para representar a los números reales. El error de redondeo se atribuye a la imposibilidad de almacenar todas las cifras de estos números y a la imprecisión de los instrumentos de medición con los cuales se obtienen los datos.

Las operaciones aritméticas pueden producir resultados que no se pueden representar exactamente en los dispositivos de almacenamiento. Si estos errores se producen en forma recurrente entonces el error propagado pudiera crecer de forma significativa dependiendo de la cantidad de operaciones requeridas. Esta cantidad de operaciones está determinada por la eficiencia del algoritmo.

En el siguiente problema se tiene un dispositivo que puede almacenar únicamente los cuatro primeros dígitos decimales de cada número real y trunca los restantes. Se requiere calcular el error de redondeo de un número real N , que para este problema es igual a 536.78

Lo primero que se tiene que hacer es expresar el número en forma normalizada, es decir sin enteros y ajustando su magnitud en potencias de 10. Esto se realizó utilizando el siguiente algoritmo:

```
1: while  $N > 1$  do
2:    $N = \frac{N}{10}$ 
3: end while
4: end
```

Luego de normalizar el número, este queda de la forma: $N = 0,53678 * 10^3$. Sin embargo, como el número solo puede almacenar los cuatro primeros números decimales, es necesario descomponer el número en dos partes: Un N real y un N aproximado, en donde la diferencia entre estos dos valores nos va a dar el sobrante del valor real.

El nuevo valor de N es: $N = 0,5367 * 10^3 + 0,00008 * 10^3$

En general, si n es la cantidad de enteros del número normalizado con potencias de 10, y m es la cantidad de cifras decimales que se pueden almacenar en el dispositivo, entonces si se truncan los decimales sin ajustar la cifra anterior, el error de redondeo absoluto está acotado por:

$$|E| < 10^{n-m}$$

El error de redondeo es equivalente a tener: $E = 0,08 * 10^{3-4} = 0,08 * 10^{-1}$

2. Problema 2: Algoritmo de Raiz Cuadrada

En este problema se implementa un algoritmo que permite calcular una aproximación a la raíz cuadrada de un número real N. El algoritmo se presenta a continuación:

```

1:  $y = (\frac{1}{2}(x + \frac{n}{x}))$ 
2: while  $|x - y| > E$  do
3:    $x = y$ 
4:    $y = (\frac{1}{2}(x + \frac{n}{x}))$ 
5: end while
6: end

```

Donde:

n = Dato de entrada

x = Valor Inicial

E = Error Permitido

y = Respuesta calculada con error E.

Luego de tener el algoritmo se procede a calcular la raíz cuadrada de un número real N para determinar si el algoritmo es preciso en sus calculos y converge. El valor de prueba que se utilizó fue la raíz cuadrada de 7. Los datos de entrada se muestran a continuación:

1. **Valor Inicial:** 0,1
2. **Tolerancia:** $10e - 8$
3. **Dato:** 7

Ademas de los datos de entrada que se mencionaron anteriormente, se utilizó un contador para determinar cuantas iteraciones iba a realizar el algoritmo antes de salir del ciclo y obtener una respuesta con un error menor al establecido.

La aproximacion de la raiz cuadrada del dato de prueba es: 2.6457513110645907

La cantidad de iteraciones que se tuvieron fue de: 8

3. Problema 3. Teorema de Taylor

El teorema de Taylor da una representación de una función usando un polinomio y una expresión para el error. Esto permite reemplazar una función por algo más sencillo, un poli-

nomio y una estimación del error.

Teorema de Taylor:

Sea f una función con sus primeras $n + 1$ derivadas continuas en $[a, b]$ y sean $x, x_0 \in [a, b]$. Entonces,

$$f(x) = P_n(x) + R_n(x) \quad (1)$$

Donde:

$$P_n(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} * (x - x_0)^k \quad (2)$$

Una vez se conoce la expresión del polinomio de Taylor se procede a implementar dicha expresión en un algoritmo para poder calcular un valor aproximado de una función $f(x)$ que en este caso es $f(x) = e^x$, con un $x = 0,5$. Para este caso en específico, no se tendrían problemas a la hora de obtener las derivadas, ya que a la función propuesta se le pueden sacar infinitas derivadas. La salida del algoritmo es el valor aproximado de la función con cinco cifras significativas. Los datos de entrada al algoritmo son:

1. **Valor Inicial x_0 :** 0
2. **Grado del Polinomio de Taylor:** 5

El valor real de la función es: 1.6487
El valor aproximado de la función con un grado de 5 es: 1.6484

Se puede ver que la aproximación con el polinomio de Taylor es muy cercana al valor real de la función y si se utiliza un grado mayor al utilizado en las pruebas, se va a poder obtener un resultado aún mejor.

4. Problema 4. Error absoluto y relativo de una operación

En los métodos directos debe considerarse el error que se propaga en las operaciones aritméticas, el cual puede ser significativo cuando la cantidad de cálculos requeridos es grande. Para el siguiente problema se tiene que calcular la distancia que viaja una partícula dada su velocidad y el tiempo que dura viajando. Sin embargo también es necesario calcular los errores relativos y absolutos que se tienen al realizar dicha multiplicación.

Para una multiplicación de dos valores X y Y , el error de redondeo viene dado por:

$$P = XY + XE_y + YE_x \quad (3)$$

En donde el error de redondeo absoluto de la multiplicación es:

$$E_{xy} = XE_y + YE_x \quad (4)$$

Y el error de redondeo relativo de la multiplicación es:

$$e_{xy} = \frac{E_y}{Y} + \frac{E_x}{X} \quad (5)$$

Ya teniendo las ecuaciones para calcular los errores de redondeo de una multiplicación, se procedió a implementar un algoritmo que permitiera calcular dichos valores. Los datos de entrada se muestran a continuación:

1. **Velocidad m/s:** 4
2. **Tiempo s:** 5
3. **Error de la velocidad:** 0,1
4. **Error del tiempo:** 0,1

Distancia de la partícula con el error absoluto: 20 ± 0.9
El error relativo fue de: 0.045

5. Problema 5. Método de Horner

El método de Horner es un algoritmo el cual sirve para calcular con n multiplicaciones un valor aproximado de un polinomio $P(x)$ dado un x_0 inicial. Se hicieron pruebas del algoritmo con el siguiente polinomio:

$$P(x) = 2x^4 - 3x^2 + 3x - 4 \quad (6)$$

Se implementó un algoritmo para calcular el valor aproximando de un polinomio en el menor número de multiplicaciones. El dato de entrada fue:

1. **Valor de x_0 :** -2

El resultado del polinomio evaluado en: $x_0 = -2$ es igual a: 10
Con un total de 4 multiplicaciones.

6. Problema 6. Formula Cuadrática Mejorada

Cuando se quiere calcular las raíces de una ecuación, se presenta un problema al utilizar la formula cuadratica general, debido a que si la ecuación presenta un valor muy grande en b este puede afectar la precisión en la formula y de las raíces ya que se obtendrán valores de 0 cuando en verdad son valores cercanos pero no exactamente 0.

Una solución es utilizar la formula citardauq, que es una variación de la formula cuadratica original pero permite tener resultados mas precisos de las raíces, ya que intercambia una resta por una suma, evitando que hayan cancelaciones de terminos. La formula es la siguiente:

$$x = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}} \quad (7)$$

Los datos de entrada que se utilizaron fueron:

1. **Valor de a:** 3
2. **Valor de b:** 9^{12}
3. **Valor de c:** -3

Los valores de las raíces con la formula original son: $x_1 = -94143178827.0$ y $x_2 = 0$

Los valores de las raíces con la formula citardauq son:

$$x_1 = 1.062211848e-11 \text{ y } x_2 = -9.414317883e10$$

Se puede ver de los resultados que la formula citardauq es más precisa a la hora de generar las raíces de una función.

7. Ejercicio 8. Intersección de dos funciones

Para este problema se requiere determinar la raíz real de la ecuación $\sin(x) = \ln(x)$. Al graficar ambas funciones se va a poder ver en donde se interseccionan las curvas y dicho valor será la raíz que se quiere encontrar.

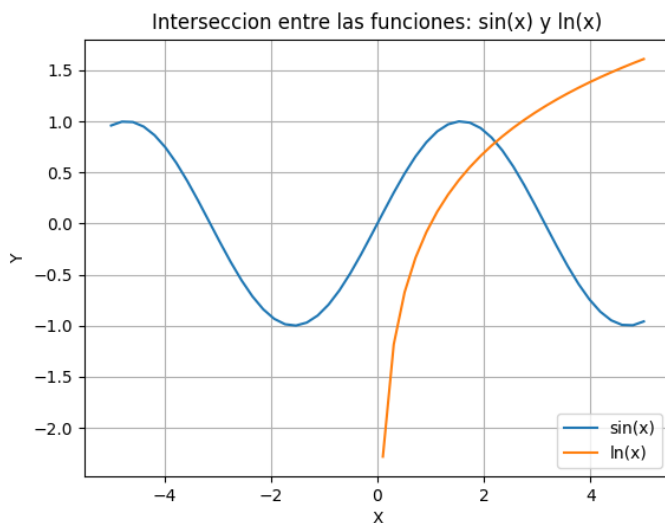


Figura 1: Gráfica Intersección de la curvas $\sin(x)$ y $\ln(x)$

De la grafica anterior se puede concluir que un intervalo apropiado para poder encontrar la raíz o intersección de las dos funciones puede ser el intervalo $[2,3]$.

Se utilizó el método de bisección, que es un método para calcular las raíces de una función. La función a evaluar en este problema es:

$$f(x) = \sin(x) - \ln(x) \quad (8)$$

Para poder utilizar el método de bisección se deben cumplir dos condiciones.

Método de Bisección Si una función f es continua en un intervalo $[a, b]$ y $f(a)$ y $f(b)$ son diferentes en signo, existe por lo menos un punto r en (a, b) tal que $f(r) = 0$. Los valores de entrada al algoritmo de bisección fueron:

1. **Valor de a:** 2
2. **Valor de b:** 3
3. **Tolerancia:** $10e - 8$

La aproximación de la raíz de la función es: 2.2191070914268494
El número de iteraciones que se tuvieron fue de: 24

En la siguiente grafica se puede ver que el error tiene un comportamiento lineal, lo cual asegura el orden de complejidad del algoritmo de bisección.

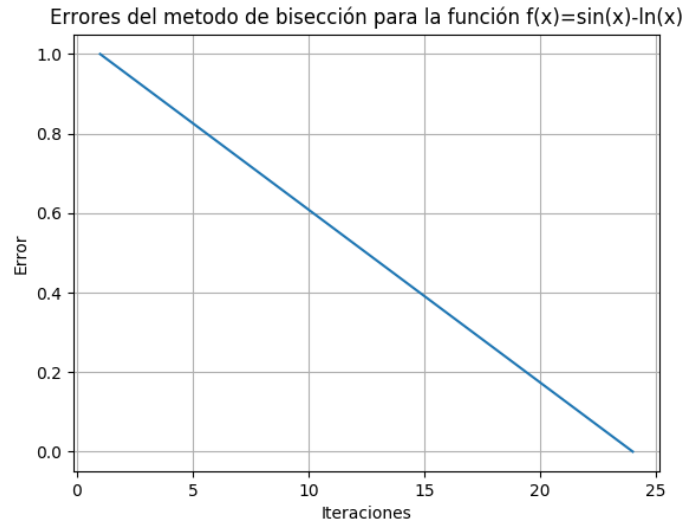


Figura 2: Grafica Iteraciones vs. Errores

8. Ejercicio 13. Algoritmo para calcular la raíz n-ésima

Para este problema se necesitaba encontrar una fórmula iterativa para calcular la raíz real n-ésima de un número real N , y el algoritmo solo debía incluir operaciones aritméticas elementales. Se tiene una función de la forma $x = \sqrt[n]{N}$ que se puede reformular de la siguiente manera: $x^n - N = 0$. De la última ecuación se puede utilizar el método de Newton para evaluar las raíces de una función. Por lo tanto, para calcular la raíz n-ésima de un número real N es necesario tener una función:

$$f(x) = x^n - N \quad (9)$$

y su derivada:

$$f'(x) = nx^{n-1} \quad (10)$$

Se implementó un algoritmo con una variación del método de Newton:

- 1: **while** $|x_{n+1}| > E$ **do**
- 2: $x_{n+1} = \left(\frac{\frac{N}{x_n^{n-1}}}{n} \right) - x_n$
- 3: $x_n = x_n + x_{n+1}$
- 4: **end while**
- 5: **end**

Teniendo el algoritmo, se proceden a determinar los valores de entrada:

1. **Valor de n:** 3

2. **Valor de N:** 120

3. **Valor de x_0 :** 5

El valor inicial es un valor cercano al valor real de la raíz, para así poder tener una convergencia más rapido.

La aproximación de la raíz 3-ésima del número 120 es: 4.932424148660941
El número de iteraciones que se tuvieron fue de: 4

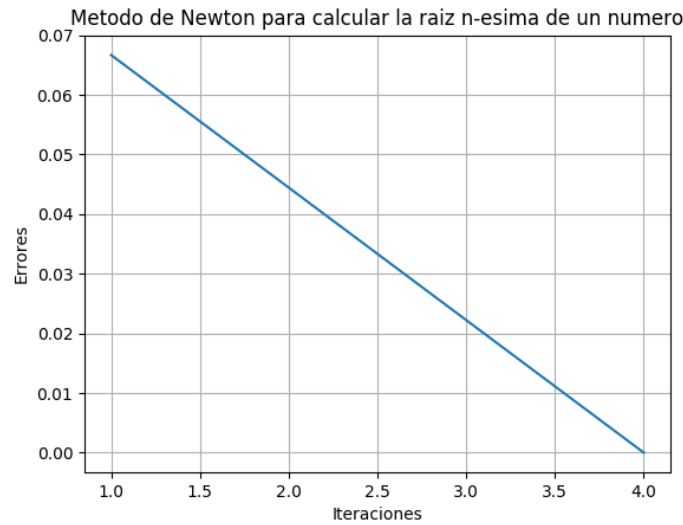


Figura 3: Grafica Iteraciones vs. Errores

El comportamiento de la anterior grafica en relación a los errores tiende a ser lineal. Por lo tanto el orden de convergencia del método tiene una convergencia lineal.

9. Ejercicio 15. Método de Punto Fijo en una Integral

Se tiene la siguiente ecuación que se pretende resolver con el método del punto fijo:

$$\int_0^X (5 - e^u) du = 2 \quad (11)$$

El primer paso para solucionar el problema es obtener la ecuación $f(x) = 0$ después de resolver la integral.

$$f(x) = -e^x + 5x - 1 \quad (12)$$

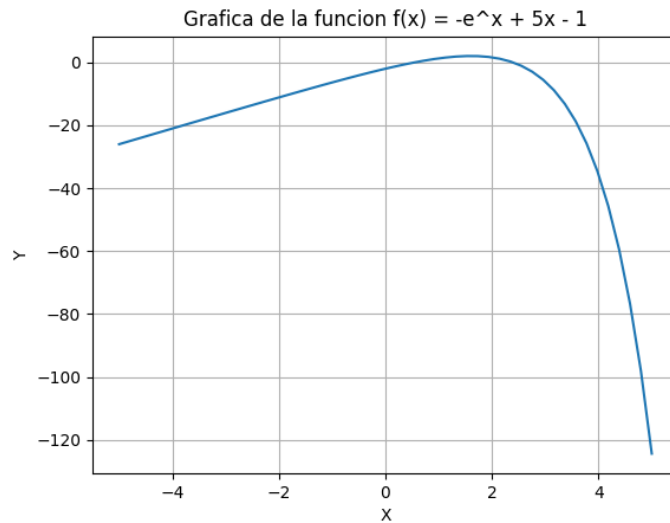


Figura 4: Grafica de la función $f(x) = -e^x + 5x - 1$

Una vez se tiene la función $f(x)$ se grafica para determinar las raíces de la función:

De la anterior grafica se puede ver que la función tiene dos raíces reales entre los intervalos $[0,1]$ y $[2,3]$ aproximadamente. Ya teniendo una idea de donde podrían estar las raíces, se procede a calcular las raíces utilizando el método del punto fijo, por lo que es necesario proponer una ecuación nueva de la forma $x = g(x)$.

$$x = \frac{e^x + 1}{5} \quad (13)$$

Una vez se tiene una ecuación $x = g(x)$, se determinan los valores de entrada para el método:

1. **Valor de a:** 0
2. **Valor de b:** 1
3. **Iteraciones máximas:** 5
4. **Tolerancia:** $10e - 8$

La aproximación de la raíz de la función es: 0.5446687194804329
El número de iteraciones que se tuvieron fue de: 5

En la siguiente grafica se puede ver que el comportamiento de los errores conforme se va iterando en el método de punto fijo tiene un comportamiento lineal, por lo que se puede decir que el método de punto fijo tiene una convergencia lineal para este caso.

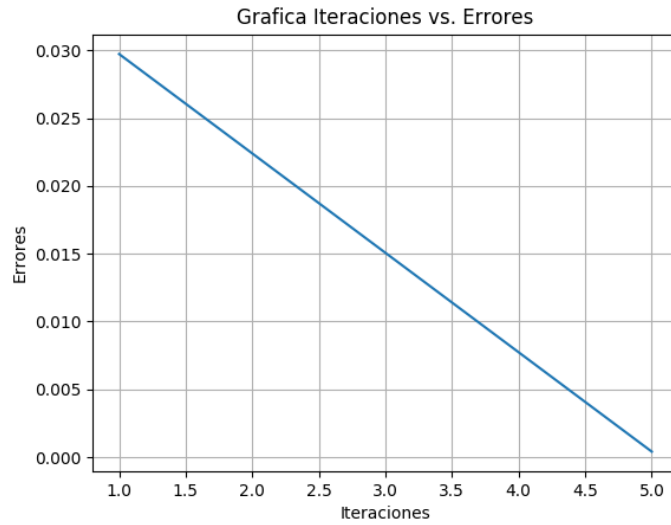


Figura 5: Grafica Iteraciones vs. Errores

Tambien se hizo un seguimiento del error en cada iteración del algoritmo y como se puede ver en los valores de la tabla, el error nunca llegó al valor de la tolerancia debido a que el ciclo terminó por otro factor que fue el de máximas iteraciones permitidas. Igualmente los errores mantuvieron una convergencia relativamente lineal como se vio anteriormente en la grafica.

Iteración	Error
1	0.029744254140025683
2	0.009955319713903399
3	0.003398707403813317
4	0.001168074516109896
5	0.000402363706580644

Cuadro 1: Tabla de Iteraciones vs. Errores.

10. Ejercicio 27. Raices en Funciones Polares

El movimiento de una partícula en el plano, se encuentra representado por las ecuaciones paramétricas:

$$x(t) = 3\sin^3(t) - 1 \quad (14)$$

$$y(t) = 4\sin(t)\cos(t) \quad (15)$$

Donde x , y son las coordenadas de la posición expresadas en cm, t se expresa en segundos. La primera parte del problema consistió en demostrar que existe un instante en $t \in [0, \pi/2]$ tal que sus coordenadas x e y coincidan. Para esto se utilizó la librería de Python para realizar graficas y además adaptar las graficas para que grafiquen funciones polares.

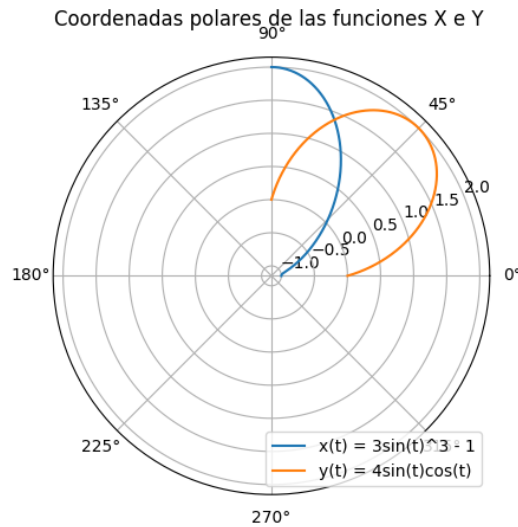


Figura 6: Coordenadas polares de las ecuaciones X e Y

De la anterior grafica, se puede ver en algún momento t en el intervalo $[0, \pi/2]$ las ecuaciones X e Y se van a intersectar, por lo que se puede tener certeza que el problema va a tener solución.

Una vez se tenía seguridad del intervalo, se procedió a calcular el instante en donde ambas ecuaciones se iban a intersectar utilizando el método de bisección para encontrar raíces, generando una ecuación única que incluyera ambas ecuaciones:

$$h(t) = 3\sin^3(t) - 1 - 4\sin(t)\cos(t) \quad (16)$$

Los datos de entrada que se consideraron fueron los siguientes:

1. **Valor de a:** 0
2. **Valor de b:** $\pi/2$
3. **Tolerancia:** $10e - 5$

El instante t en donde las ecuaciones X e Y se intersectaran es: 1.1864382656303005
El número de iteraciones que se tuvieron fue de: 14