Pontificia Universidad Javeriana

Departamento de Matemática
Análisis Numérico

Taller 1

Parte 1: Problemas

por

Monica Alejandra Alvarez Carrillo
monica_alvarez@javeriana.edu.co

Santiago Palacios Loaiza
palacios-santiago@javeriana.edu.co

Profesora: Eddy Herrera Daza

Bogotá D.C., 10 de Enero del 2020

1. Problema 1: Error de truncamiento

1.1. Enunciado

Suponga que un dispositivo solo puede almacenar únicamente los cuatro primeros dígitos decimales de cada número real, y trunca los restantes (esto es redondeo inferior). Calcule el error de redondeo si se quiere almacenar el número 536.78.

1.2. Teoría

Existen operaciones aritméticas que producen resultados que no se pueden representar de manera exacta en las computadoras, éstas últimas usan métodos como el redondeo y el truncamiento para tratar con este inconveniente. Sin embargo, esta falta de exactitud deriva en una propagación de error bastante pequeño, pero que puede aumentar de acuerdo al número de operaciones que se realice.

Dicho esto, la formula que permite calcular el error absoluto de truncamiento es:

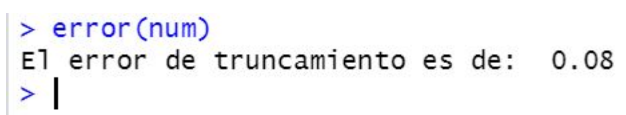
$$|E| < 1 * 10^{n-m}$$

Y del error relativo:

$$|e| < \frac{max(|E|)}{min(|X|)} = \frac{1*10^{n-m}}{0,1*10^{n}} = 10 * 10^{-m}$$

1.3. Solución

Se implementó el algoritmo que permite calcular dicho error de truncamiento, éste recibe como entrada el número. El número máximo de dígitos que se puede almacenar es 4 y está por defecto en una variable. Al realizar la prueba se envío el número 536,78 y se obtuvo como resultado 0,08 de error.



Error calculado por el algoritmo

El código realizado en R se encuentra en el mismo repositorio que este documento, bajo el nombre de ErrorTruncamiento.R

2. Problema 2: Cálculo de la raíz cuadrada

2.1. Enunciado

Implemente en cualquier lenguaje el siguiente algoritmo que sirve para calcular la raíz cuadrada. Aplíquelo para evaluar la raíz cuadrada d<e 7, analice su precisión, como podría evaluar la convergencia y validez del algoritmo.

2.2. Teoría

Los métodos numéricos iterativos permiten producir una respuesta más aproximada a la real mediante el uso de aproximaciones sucesivas, esto permite que el valor estimado sea mucho mejor tras cada iteración y la convergencia del método está dada por la reducción del error absoluto. La teoría de los métodos iterativos propone una fórmula que permite hallar la raíz cuadrada de un número real positivo mediante el uso de operaciones básicas.

- Fórmula:

$$y = \frac{1}{2}(x + \frac{n}{x})$$

  Dónde x representa el valor aproximado de la raíz.

A continuación se muestra el algoritmo propuesto por el enunciado.

```
Algoritmo: Raíz cuadrada
Entra:      n     Dato
            E     Error permitido
            x     Valor inicial
Sale:       y     Respuesta calculada con error E
y ← 1/2(x + n/x)
Repetir mientras  |x - y|  >  E
      x ← y
      y ← 1/2(x + n/x)
Fin
```

Algoritmo propuesto para el calculo de la raíz

El algoritmo funciona haciendo un par de operaciones simples que aproxima un valor inicial al resultado real de la raíz, luego de cada iteración se comprueba el ultimo valor dado con el anterior y con la tolerancia dada.

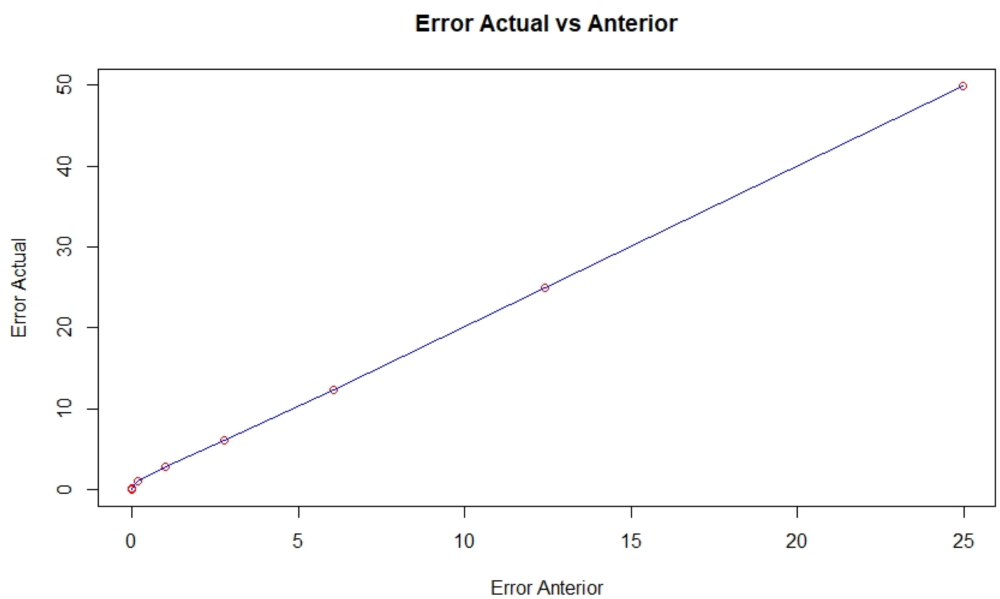
2.3. Solución

Para la solución del problema se aplico el algoritmo propuesto al pie de la letra. Siendo la operación principal y=0.5 * (x + (n / x)). Donde 'y' corresponde al resultado final, 'n' al numero sobre que se le quiere calcular la raíz y 'x' es el valor desde el cual se hace la aproximación a la raíz. También se usa un parámetro 'E' que representa el valor de la tolerancia del error.

Para probar el algoritmo se utilizaron los siguientes valores: y=7; E=0.00000001; x=100. El resultado obtenido fue: 2.645751. El anterior es un valor correcto que cumple con las características del enunciado, a continuación se adjuntan las tabla de valores y la gráfica respectiva.

Error Actual	Error Anterior
24.94755	49.96500
12.40421	24.94755
6.06566	12.40421
2.77989	6.06566
1.00683	2.77989
0.17905	1.00683
0.00604	0.17905
0.00001	0.00604
0.00000	0.00001

Tabla con los resultados de los errores



Gráfica que muestra la convergencia de los errores y la aproximación al valor final.

El código realizado en R se encuentra en el mismo repositorio que este documento, bajo el nombre de Raiz.R"

3. Problema 3: Aproximación polinómica con Teorema de Taylor

3.1. Enunciado

Utilizando el teorema de Taylor hallar la aproximación de $e^{0,5}$ con cinco cifras significativas.

3.2. Teoría

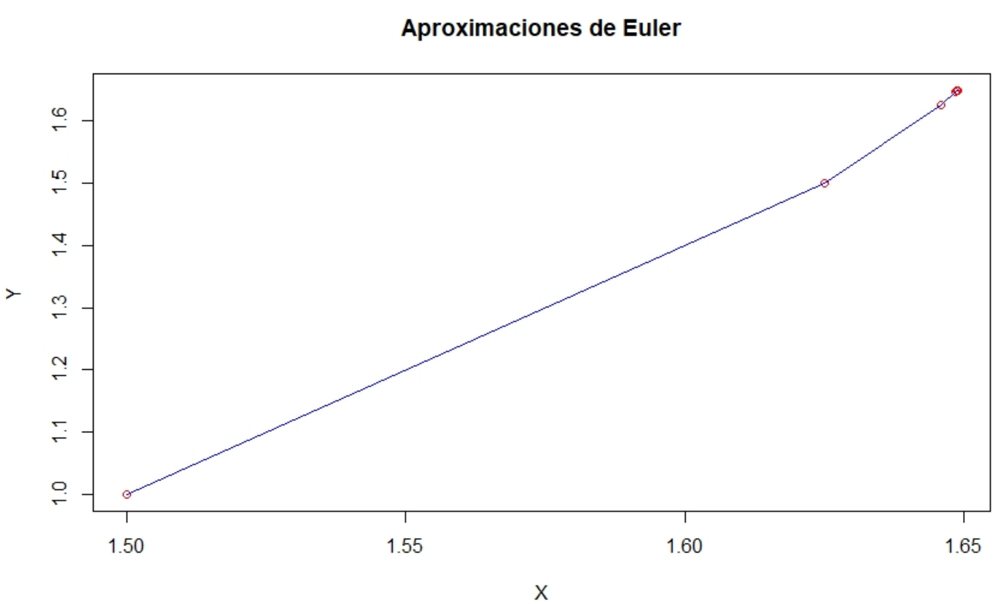
El teorema de Taylor fue enunciado por Brook Taylor en 1712. Es un teorema matemático que esta creada para conseguir aproximaciones polinómicas de una función, mientras que la función misma sea diferenciable. Este teorema también puede ser usado para acotar el error de su propia estimación. El teorema de Taylor esta fuertemente asociado con la función e^x pues es un perfecto ejemplo de como puede funcionar el teorema con una función diferenciable.

3.3. Solución

Para la solución se siguió la siguiente formula propuesta que representa el teorema de los infinitos polinomios de Taylor en la función e^x

$$f(x) = e^x = +1x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots + \frac{x^n}{n!} + \ldots$$

El polinomio anterior fue trasladado a código de R y se itero hasta que el error normalizado porcentual fue menor a 0.00005 para obtener las 5 cifras requeridas en el enunciado. A continuación se muestra la gráfica del avance del valor de cada iteración con respecto a la iteración anterior.



Gráfica que muestra la aproximación al valor final de e^x

El código realizado en R se encuentra en el mismo repositorio que este documento, bajo el nombre de "TaylorEuler.R"

4. Problema 4: Error en el cálculo de la distancia

4.1. Enunciado

Calcule el tamaño del error dado por las operaciones aritméticas, para la solución del siguiente problema

La velocidad de una partícula es constante e igual a **4 m/s**, medida con un error de **0.1 m/s** durante un tiempo de recorrido de **5 seg**. medido con error de **0.1 seg**. Determine el error absoluto y el error relativo en el valor de la distancia recorrida.

$v = 4$, $E_v = 0.1$	(velocidad)
$t = 5$, $E_t = 0.1$	(tiempo)
$d = vt$	(distancia recorrida)

4.2. Teoría

Los errores numéricos se generan con el uso de aproximaciones para representar operaciones y cantidades matemáticas, en la generación de cálculos dados por algoritmos. El error de truncamiento es aquel que resulta al usar una aproximación en lugar de un procedimiento matemático exacto y representa la diferencia entre estos dos resultados. Este error de truncamiento esta fuertemente relacionado con el teorema e Taylor por su utilidad cuando se reemplaza una expresión matemática compleja por una formula mas simple.

4.3. Solución

Para la solución de este ejercicio solo fue necesario implementar las formulas simples de los errores y aplicarlas con los datos dados en el enunciado. Ambas medidas, tanto la velocidad como el tiempo contaban con un error de 0,1 en su medición. A continuación se adjunta el

código implementado que calcula los errores pedidos en el enunciado

```

ea<-(v*ev)+(t*et) #calculo del error absoluto

erel<-(ev/v)+(et/t) #calculo del error relativo
```

El código completo realizado en R se encuentra en el mismo repositorio que este documento, bajo el nombre de ”ErrorDistancia.R”

5. Problema 4:

5.1. Enunciado

Evaluar el valor de un polinomio es una tarea que involucra para la máquina realizar un número de operaciones la cual debe ser mínimas. Como se puede evaluar el siguiente polinomio con el número mínimo de multiplicaciones $P(x) = 2x^4 - 3x^2 + 3x + 4, x_0 = -2$

5.2. Teoría

El metodo de Horner es un teorema usado en el análisis numérico creado por el matemático ingles William George Horner en el siglo XVII que consiste en calcular el valor de un polinomio de grado n utilizando el menor numero de multiplicaciones. Este teorema es especialmente útil para la solución de este ejercicio pues el mismo exige la evaluación de un polinomio con el numero mínimo de multiplicaciones.

5.3. Solución

Para la solución del problema se implemento el método del teorema de Horner en R, como parámetros: coeficientes=[2, 0, -3, 3, 4] y $x_0 = 2$ y dio como resultado un valor para el polinomio de 18, donde se realizaron 4 sumas y 4 multiplicaciones, para un total de 8 operaciones mínimas.
$18 = 2x^4 - 3x^2 + 3x + 4; x_0 = -2$

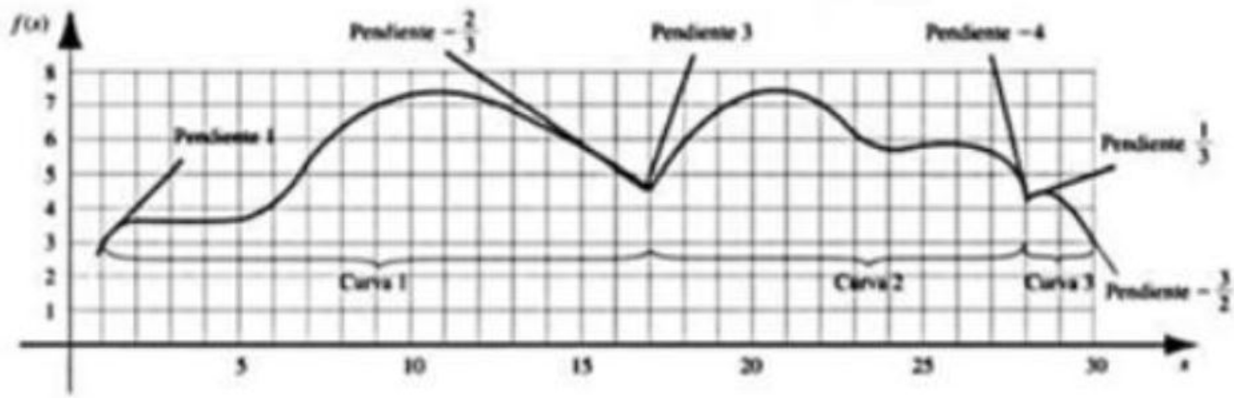
El código realizado en R se encuentra en el mismo repositorio que este documento, bajo el nombre de "Horner_Ejercicio.R"

6. Problema 5: Reconstrucción del perrito

6.1. Enunciado

Reconstruir la silueta del perrito utilizando la menor cantidad de puntos para reproducir el dibujo del contorno completo del perrito sin bigotes, con la información dada

Coordenadas:
y=c(3, 3.7, 3.9, 4.5, 5.7, 6.69, 7.12, 6.7, 4.45, 7, 6.1, 5.6, 5.87, 5.15, 4.1, 4.3, 4.1, 3)
x=c(1 , 2, 5, 6, 7.5, 8.1, 10, 13, 17.6, 20, 23.5, 24.5, 25, 26.5, 27.5, 28, 29, 30)



6.2. Teoría

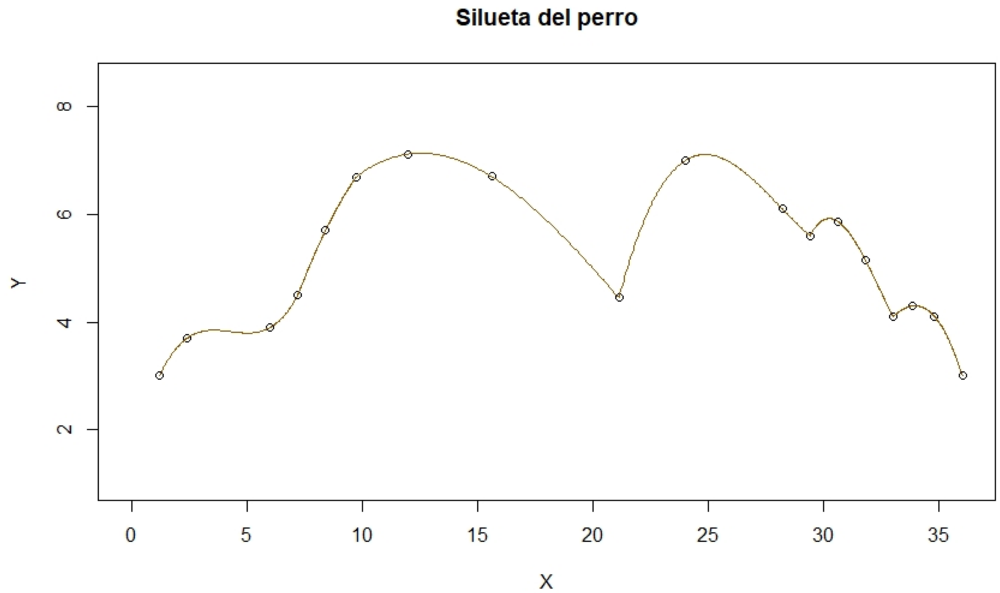
La Interpolación es un campo del Análisis Numérico, que se encarga de la obtención de puntos a partir de otros conjuntos de puntos y de la aproximación de funciones complicadas mediante funciones mas simples. Es esta ultima función de la interpolación que resulta mas útil para la solución del problema. Una característica importante de la interpolación que la diferencia de las regresiones es que todo los puntos son interceptados por las funciones que se crean.

6.3. Solución

Para la solución del problema se le hicieron pequeños ajustes a la serie de puntos propuestos, los nuevos vectores se muestran a continuación: y=c(3, 3.7, 3.9, 4.5, 5.7, 6.69, 7.12, 6.7, 4.45, 7, 6.1, 5.6, 5.87, 5.15, 4.1, 4.3, 4.1, 3)

x=c(1 , 2, 5, 6, 7, 8.1, 10, 13, 17.6, 20, 23.5, 24.5, 25.5, 26.5, 27.5, 28.2, 29, 30)

Tras arreglar las coordenadas se dividieron los puntos en otros vectores para facilitar la tarea de la función "spline".La anterior es una función de un paquete de R que recibe como parámetros los vectores con los puntos y un numero n, y retorna la misma cantidad n de puntos sobre funciones cubicas que unen los puntos que se enviaron como parámetros. A continuación se muestra la gráfica que es resultado de pintar los puntos entregados por la función "spline"



Gráfica que muestra la aproximación final a la silueta del perro

7. Bibliografía

- Análisis Numérico Básico con Python.