Taller 1

Estiben Giraldo y Julio Andrés Mejía

Febrero 14 2020

Contents

1 Problemas Iniciales

1.1 Problema 1: Error de redondeo

Los métodos numéricos operan con datos que pueden ser inexactos y con dispositivos para representar a los número 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 un computador. Si estos errores se producen constante entonces se propagará este error de forma significativa dependiendo de la cantidad de operaciones necesarias.

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.

Es por esto, que se necesita calcular el error de redondeo de un número real N, que para este problema es igual a **536.78**.

Lo primero que se debe hacer es normalizar el número, es decir, tomando su parte decimal y ajustando su magnitud en potencias de 10. Para esto se se realiza utilizando el siguiente algoritmo:

```
while N>1 do
    N = N/10
end while
```

Luego el número queda de la forma $N = 0,53678 * 10^3$. Sin embargo, como solo se puede almacenar los cuatros primeros números decimales, es necesario descomponer el número en parte real y un aproximado, donde su diferencia tendrá como resultado el valor real.

El nuevo valor de N es: $N = 0,53678 * 10^3 + 0,00008 * 10^3$

En general, si n es la cantidad de eneteros del número normalizado con potencia 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 redondero está acotado por:

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

$$\boxed{\text{El error de redondeo es de: } E = 0,08 * 10^-1}$$

1.2 Problema 2: Raiz cuadrada

Para este problema se propone un algoritmo que permite calcular una aproximación a la raíz cuadrada de un número real N. El algoritmo es de la siguiente forma:

```
y = ((1/2 (x + n/x))
while —x+y—¡E do
    x=y
    y = ((1/2 (x + n/x))
end while
```

Donde:

- n, Dato de entrada
- x, Valor inicial
- E, Error máximo
- y, Repuesta calculada con error E

Luego de desarrollar el algoritmo se procede a calcular la raíz cuadrada de un número real N para determinar si el algoritmo es preciso y si converge. El valor de prueba que se utiliza es la raiz cudrada de 7. Los datos de entrada, entonces son:

x = 0,1

$E = 10^-8$

n = 7

Vale la pena aclarar que para verificar la convergencia del método se utilizó un contador de iteraciones.

La aproximación de la raíz cuadrada del dato de prueba es de: 2.645751. La cantidad de iteraciones para coverger fueron de: 7

1.3 Problema 3: Teorema de Taylor

El teroma de Taylor representa una función usando un polinomio con una expresión para el error. Eso permite reemplazar una función por un polinomio y estimación del error.

Teorema de Taylor: Sea f una función con sus primeras n+1 derivadas continuas en [a,b] y sea $x, x_0 \in [a, b]$ Entonces,

$$f(x) = P_n(x) + R_n(x)$$

Donde:

$$P_n(x) = \sum \frac{f^n(x_0}{k!} * (x - x_0)^k$$

Una vez se conoce la expresión en el polinomio de Taylor se procede a implementar dicha expresión en un algoritmo para poder calcular un valor aproximado de la función *f(x)*, en este caso es $f(x) = e^x$, con un $x_0 = 0.5$. Para la evaluación de este polinomio se realizó un polinomio de grado 5, sin embargo, si se desea, el grado puede ser modificado en el código fuente. Los resultados obtenidos fueron:

El resultado real es igual a 1.64872 El resultado aproximado con Taylor es igual a 1.6487

1.4 Problema 4: Error de operaciones artiméticas

Al ser un metodo de directo se tiene que considerar el error de cada dato y como este propaga, siendo este mas significativo a medida que la operacion es mas grande o complicada.

Las entradas de este ejercicio son:

- $v = 4m/s$ Velocidad.
- $E_v = 0.1m/s$ Error de medicion de la velocidad.
- $t = 5s$ Tiempo recorrido.
- $E_t = 0.1s$ Error de medicion del Tiempo.

Esto nos da como resultado:

$d = 20m$ Distancia
$[19.1, 20.9]$ Rango de Error
$e_r = 4.5\%$

1.5 Problema 5: Evaluar un polinomio

Para evaluar el polinomio $P(x) = 2x^4 - 3x^2 + 3x - 4$, con $x = -2$ se uso el metodo de Horner, claculando las n multiplicaciones y las m sumas que se hagan, para esto se usan los coeficientes que usa cada x y el valor de x este siendo el valor que se calcula cada función.

Sea $P_0(x) = a_0x^0 = a_0$ El numero de multiplicaciones para hallar $P_0(x_0)$ es igual a 0. Se asume entonces que $P_k(x) = a_0 + a_1x + a_2x^2 + ... + a_kx^k$ y que $P_k(x_0) = a_0 + a_1x_0 + ... + a_kx_0^k$ tienen k multiplicaciones.

Gracias a esto aplicado al algoritmo se entonces que:

El resultado de $P(x) = 2x^4 - 3x^2 + 3x - 4$ en $x = -2$ es igual a 10
El numero de multiplicaciones requeridas fueron de 5

1.6 Problema 6: Interpolación silueta de un perro

Para la solución de la silueta de un perro se tomaron los puntos suministrados, partiendo de la suposición que fueron bien tomados. Ahora bien, para producir una buena aproximación de dicha silueta a partir de los datos se utilizó el método de spline cúbico. Un trazador o spline es una banda flexible que se usa para dibujar curvas suaves a través de un conjunto de puntos. Esta función consiste en la unión de polinimios cúbicos por intervalo. En otras palabras un trazador cúbico es una función a trozos que interpola a f en los $n + 1$ puntos $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$ con $a = x_0 < x_1... < x_n = b$. La imagen resultante es la siguiente:

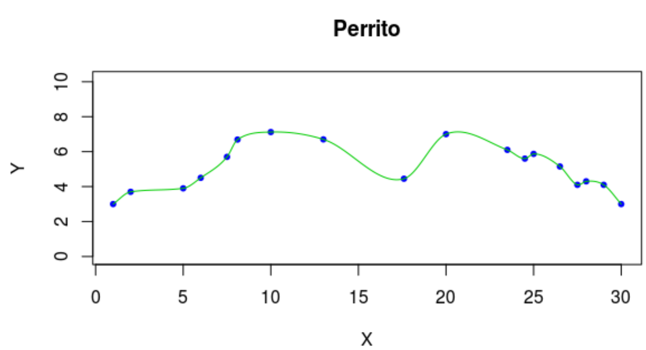


Figure 1: Silueta Perrito

2 Problemas Primera Parte

2.1 Problema 1: Teorema de Horner

El método de Horner, se conoce como una multiplicación anidada, el cual consiste en determinar de manera eficiente el valor del polinomio de grado n utilizando el menor número de productos. Un polinomio de grado n puede evaluarse en n multiplicaciones y n adiciones. Existen diferentes métodos que realizan diferentes número de multiplicaciones; para este caso, se realizará el segundo método, el cual es $2n - 1$. Para demostrarlo se utilizará inducción matemática.

Base: Polinomio de grado *1*
$P(x_0) = 3 * x + 4$
Como se puede observar la cantidad de multiplicaciones a realizar es de 1
Inductivo: Polinomio de grado *4*
$P(x_0) = 2x^4 - 3x^2 + 3x + 4$
Descomponiendo el polinomio, por cantidad de operaciones, entonces:
$P(x_0) = 2(x) * (x^3) + 0 * (x) * (x^2) - 3(x) * (x) + 3 * x + 4$
Si se cuenta el número de multiplicaciones que deben realizarse se llega a que se necesita realizar un total de 7 operaciones. Por lo tanto, para un polinomio de grado n:
$P(x_0) = a_0 + a_1x + a_2x^2 + ... + a_nx^n = 2n + 1$

Para probar el teorema de Horner se utiliza un algoritmo que evalua el polinomio de la forma que muestra el método 2, para este paso se utilizó el polinomio $P(x_0) = 2x^4 - 3x^2 + 3x + 4$ en $x_0 = -2$, además se realizó una comparación del resultado del método con la expresion equivalente $Q(x) = (x^{51} - 1)/(x - 1)$ obteniendo así los siguientes resultados:

El resultado del polinomio evaluado en $x_0 = -2$ es igual a: 10
El resultado del polinomio evaluado en $x_0 = 1.0001$ es igual a: 50.1227
El resultado del polinomio evaluado en $x_0 = 1.0001$ es igual a: 51.12771
El error de cálculo, al comprarlo con la expresion es de: 1.005012

2.2 Problema 2: Numeros Binarios

Los números decimales se convierten de base 10 a base 2 con el fin de almacenar números en una computadora y para simplicar las operaciones hechas por la computadora, como la suma y la multiplicación.

Los números binarios se expresan como:

$$...b_2b_1b_0b_-1b_-2...$$

Donde, cada dígito binario, o bit, es 0 o 1. El equivalente en bes 10 de un núumero es:

$$...b_22^2 + b_12^1 + b_02^0 + b_{-1}2^{-1} + b_{-2}2^{-2}...$$

- Primeros 15 bits de la representacion binaria de π :

La entrada del problema es: 3.14159265358979
Su parte Binaria es:
11.001001000011111101101010100010001000010110100011

- Convertir los siguientes núumeros binarios a base 10:

– 1010101 = 85
– 1011.101 = 11.625
– 10111.010101 = 23.328
– 111.1111 = 7.9375

- Convierta los siguientes núumeros de base 10 a binaria:

– 11.25 = 1011.01
– 2/3 = 0.10101010101010101...
– 30.6 = 11110.1001100110011...
– 99.9 = 1100011.111001100110...

2.3 Problema 3: Representación del Punto Flotante de los número reales

Existen dos tipos de formatos para los números, los cuales son:

Punto Fijo: Los números de punto dijo se usan para almacenar enteros. Por lo general se almacenan como máximo 2^{32} números diferentes.

Punto Flotante: Representación que utiliza la computadora, definido en el estandar de la IEEE 754. Este número épsilon de la máquina indica cuántos dígitos el computador es capaz de representar en números binarios. En el caso de la doble precisión son de 2^{-52} y 2^{-53} esto quiere decir que el trabajor a lo sumo es capaz de trabajar con 16 dígitos.

Es por esto que el computador para ajustar un número binario infinito en un número finito de bits en el caso de la doble precisión lo que hace es utilizar el recorte, el cual, consta de representar en forma binaria (2^{52}) y los bits que no alcancen a ser representados de esta forma, son truncados. Es aca donde radica la diferencia entre redondeo y recorte, ya que, el redondeo aparece por la necesidad de redonder números con expansión decimal finita, por otro lado, el recorte está relacionado con modelos matemáticos que necesitan ser "discretizados"o que la máquina no es capaz de representar dicho número por desbordamiento, esto acarrea dos tipos de errores, los cuales son errores de redondero y error de truncamiento.

2.4 Problema 4: Raíces de una Ecuación

1. Método para sumar la submatriz triangular superior e inferior de una matriz cuadrada: Para hacer esta sumatoria se tomo la matriz cadrada y se le dio un valor de 0 a cada valor de la triangular contraria a la que se estuvira calculando.

Por ejemplo para un n = 5 la matriz es:

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    6   11   16   21
[2,]    2    7   12   17   22
[3,]    3    8   13   18   23
[4,]    4    9   14   19   24
[5,]    5   10   15   20   25
```

Para cacular la suma de la triangular inferior la matriz queda asi:

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    2    7    0    0    0
[3,]    3    8   13    0    0
[4,]    4    9   14   19    0
[5,]    5   10   15   20   25
```

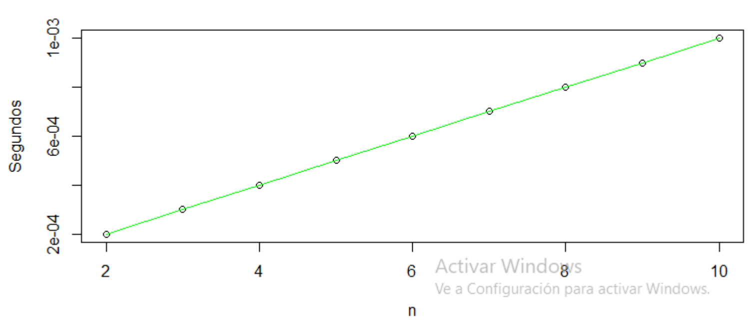
El resultado de la sumatoria de la matriz triangular inferior es: **155**

Para la matriz superior su triangular quedara asi:

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    6   11   16   21
[2,]    0    7   12   17   22
[3,]    0    0   13   18   23
[4,]    0    0    0   19   24
[5,]    0    0    0    0   25
```

El resultado de la sumatoria de la matriz triangular superior es: **235**

Y este al ser un metodo directo el tiempo que tarde en hacerse esta operacion depende que tan grande sea n en el momento subiendo este con cada n:

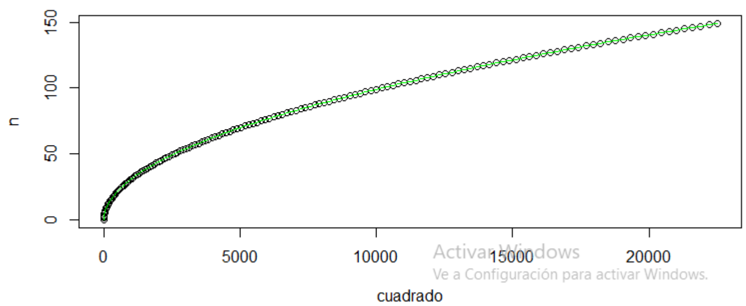


2. Metodo para hacer la sumatoria de los primeros n^2 terminos:

La tardanza de este metodo es lineal ya que depende del tamaño de n pero su crecimento no es exponencial porque a pesar del tamaño no es una operacion complicada para una maquina:

Ejeplos:

- Para n = 20 el resultado de la sumatoria es: 2870
- Para n = 50 el resultado de la sumatoria es: 42925
- Para n = 150 el resultado de la sumatoria es: 1136275

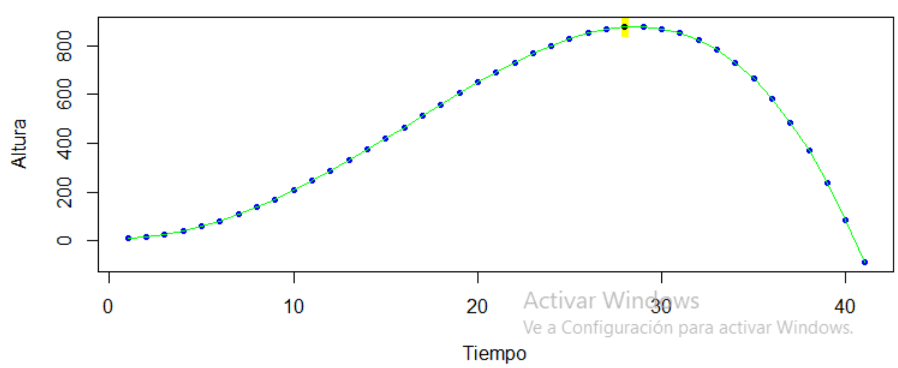


Calcular la trayectoria de un cohete: En este caso se necesita calcular la altura maxima lo cual se hizo comparando cada valor dado en y(t) y si este resultaba mayor al anterior seria el nuevo maximo, demostrando asi que teniendo el modelo:

$$\boxed{y(t) = 6 + 2.13t^2 - 0.0013t^4}$$

Su altura maxima sea:

$$\boxed{\textbf{876.87 metros despues de 27 segundos}}$$



2.5 Problema 5: Converencia de métodos iterativos