Problemas 1 – Análisis Numérico

Juan Francisco Hamón (hamon_juan@javeriana.edu.co)

Diego Mauricio Bulla (diegobulla@javeriana.edu.co)

Juan Diego Campos Neira (juand.campos@javeriana.edu.co)

11 de agosto de 2019

Problema N°1: 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

Solución: Para poder resolver el problema, primero, se expresó el número a almacenar de forma que quedara sin parte entera y se ajustó su magnitud en potencias de 10. Es decir, se pasó

$$Numero = 536.78 \rightarrow Numero = 0.53678\, \boldsymbol{x}\, 10^3$$

Luego se descompuso el número en la suma de dos números, dejando el primero con la cantidad de cifras que el dispositivo permite almacenar y el otro con lo que sobrase del número original, obteniendo:

$$Numero = 0.5367\, \boldsymbol{x}\, 10^3 + 0.00008\, \boldsymbol{x}\, 10^3$$

Con esta descomposición sabemos que el dispositivo almacena el valor aproximado de $0.5367\, \boldsymbol{x}\, 10^3$. Sin embargo, lo que nos interesa encontrar es el error de redondeo que supone dicha aproximación, por lo que se utilizó la propiedad que dice:

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

Por lo que al aplicar esta propiedad se obtiene:

$$E = 0.00008\, \boldsymbol{x}\, 10^3 \rightarrow E = 0.8\, \boldsymbol{x}\, 10^{3-4} \rightarrow E = 0.8\, \boldsymbol{x}\, 10^{-1}$$

La solución a este problema se encuentra en el documento "Error redondeo.cpp" en la misma carpeta en la que se encuentra este documento; en el programa se utiliza el proceso descrito para llegar a la respuesta encontrada, obteniendo como respuesta:

$$Error\ de\ redondeo = \ 0.08$$

El programa solicita que se ingresen los datos del número a almacenar (en este caso 536.78) y luego la cantidad de cifras que almacena el dispositivo (en este caso 4), para luego proceder a mostrar el mensaje con el resultado de la operación.

Problema N°2: Implemente en cualquier lenguaje el siguiente algoritmo que sirve para calcular la raíz cuadrada. Aplíquelo para evaluar la raíz cuadrada de 7, analice su precisión, como podría evaluar la convergencia y validez del algoritmo.

```
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
```

Solución: El algoritmo se implementó en el programa "raiz_cuadrada.R", siguiendo el proceso descrito anteriormente y llegando a una respuesta aproximada del valor de la raíz cuadrada de 7. La convergencia del algoritmo propuesto es lineal, pues al calcular los errores de truncamiento se ve que estos decrecen de forma lineal y, como se ve en el algoritmo, el valor de Y siempre está siendo afectado por el $\frac{1}{2}$ que podría considerarse un valor cercano al valor de convergencia del algoritmo.

Por su parte, la precisión dependerá del error permitido que se ingrese en el programa, puesto que este dato (junto con el valor inicial) es el que hace que el ciclo para calcular el valor estimado (que se detiene al aplicar el error absoluto como condición de parada) siga las veces necesarias y entre mayor operaciones se hagan, más cerca se estará de la respuesta real.

El programa retorna como salidas el resultado aproximado de la raíz evaluada junto con el número de iteraciones en las que se realizó el proceso:

$$Resultado = 2.645751\ ; Iteraciones = 3$$

Finalmente, para hacer una validación de que la respuesta se aproxima al valor real, se eleva el resultado obtenido al cuadrado, es decir, se hace el proceso inverso para ver qué tan cerca se está del valor inicial para calcular la raíz.

Problema N°3: Utilizando el teorema de Taylor hallar la aproximación de $e^{0.5}$ con cinco cifras significativas.

Solución:

Este problema se resolvió con un algoritmo muy parecido al de la raíz cuadrada, dado que se utiliza la misma forma de resolver, entonces, como se puede observar, el algoritmo tiene una convergencia lineal, dado que es solamente un término y que las derivadas de él mismo, van a ser iguales siempre por el tipo de expresión. Por lo tanto, lo que se planteó fue lo siguiente:

n = exp(0.5); -> Se guarda la función a trabajar

e = 0.00000001; -> Se guarda el error que se permite

x = 0.0001; -> se crea una aproximación

y; -> Se crea una variable donde se guardará la respuesta

it = 0; -> Contador para tener el número de iteraciones

Se declararon variables necesarias para el programa.

Después se realiza un ciclo donde se evalúan las respuestas guardadas en la variable "y" hasta que alcance el error permitido, el cual se mostrará a continuación, utilizando el mismo proceso de la raíz; calculando de esta manera, se calculó la aproximación de $e^{0.5}$. La solución al problema se encuentra en el archivo "Exponencial.cpp"

Problema N°4: 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)

Solución: Para resolver este problema se utilizaron los conceptos de la propagación del error de redondeo en las operaciones aritméticas, hallando que el error absoluto y relativo que se da en las multiplicaciones es:

$$E_{abs} = xE_y + yE_x \; ; \; E_{rel} = \frac{E_y}{y} + \frac{E_x}{x}$$

Lo que, en términos del problema, se puede reescribir como:

$$E_{abs} = vE_t + tE_v \; ; \; E_{rel} = \frac{E_v}{v} + \frac{E_t}{t}$$

La solución a este problema se encuentra en el archivo "Errores operaciones aritmeticas.cpp", en donde se pide el valor de v, Ev, t y Et (en ese orden) para luego proceder a calcular internamente tanto el error absoluto de la operación como el relativo, además del resultado de la operación $d = vt$; obteniendo las siguientes salidas:

$$v * t = 20$$

$$error\ relativo = 4.5\%$$

$$error\ absoluto = 0.9$$

Problema N°5: Evaluar el valor de un polinomio es una tarea que involucra para la maquina realizar un número de operaciones la cual debe ser mínimas. Como se puede evaluar el siguiente polinomio con el número mínimo de multiplicaciones.

$$f(x) = 2x^4 - 3x^2 + 3x - 4 \ \text{ en } x_0 = -2$$

Solución: Para solucionar este problema se utilizó el método de Horner, el cual permite calcular el valor de un polinomio evaluado en un valor determinado de X. Así mismo, el algoritmo establece que, tomando el grado del polinomio como g, solo se requieren g sumas y g multiplicaciones.

La implementación del método de Horner se encuentra en el archivo "Operaciones minimas.cpp" en la misma carpeta que se encuentra este documento; el programa primero pide ingresar la cantidad de coeficientes que tiene el polinomio para luego pedir ingresar cada uno de los coeficientes (de izquierda a derecha en el polinomio) del polinomio:

$$Cantidad\ de\ coeficientes = 4$$

$$Coeficiente\ 1 = 2; Coeficiente\ 2 = -3; Coeficiente\ 3 = 3; Coeficiente\ 4 = -4$$

Finalmente, el programa pide el valor a evaluar de X (en este caso -2) para así poder realizar el proceso del método de Horner, el cual está descrito de la siguiente forma:

$$Resultado_{i+1} = Resultado_i * ValorX + Coeficiente_i$$

Luego de aplicar dicho proceso, el programa muestra las siguientes salidas para el problema a resolver:

$$Resultado = -38$$

$$Cantidad\ de\ sumas/restas = 4$$

$$Cantidad\ de\ multiplicaciones = 4$$

$$Cantidad\ de\ operaciones = 8$$