Taller 1 - Análisis Numérico

Sergio Andrés Mejía Tovar
sergio.mejia@javeriana.edu.co

Julián David Parada Galvis
julian_parada@javeriana.edu.co

15 de febrero de 2019

1. **Evaluar el valor de un polinomio:** Para este problema se ha usado el método de Horner, el cual sirve para calcular con n multiplicaciones un valor aproximado de un polinomio $P(x)$ dado un x_0. Sea el método iterativo de Horner el siguiente: Para un polinomio $P(x) = a_0 + a_1x + ... + a_nx^n$, la evaluación en x_0 se dará como:

$$\begin{gathered} b_n = a_n \\ b_{n-1} = a_{n-1} + b_n * x_0 \\ ... \\ b_0 = a_0 + b_1 * x_0 \end{gathered}$$

   b_0 tendrá entonces el valor evaluado de $P(x_0)$. Sea $P_{Hi}(x_0)$ el valor del polinomio y $P_i(x); x = x_0$ usando el método de Horner.

   a) Comparación del valor real de la evaluación de tres polinomios con el valor arrojado por el método de Horner y las multiplicaciones realizadas.

   1)
   $$P_1(x) = 2x^4 - 3x^2 + 3x - 4; x_0 = -2 \tag{1}$$
   El valor real de $P_1(x_0) = 10$ se compara con $P_{H1}(x_0) = 10{,}0$ calculado usando únicamente 5 multiplicaciones.

   2)
   $$P_2(x) = 7x^5 + 6x^4 - 6x^3 + 3x - 4; x_0 = 3 \tag{2}$$
   El valor real de $P_1(x_0) = 2030$ se compara con $P_{H3}(x_0) = 2030{,}0$ calculado usando únicamente 6 multiplicaciones.

   3)
   $$P_3(x) = -5x^6 + 3x^4 + 2x^2 - 4x; x_0 = -1 \tag{3}$$
   El valor real de $P_1(x_0) = 4$ se compara con $P_{H3}(x_0) = 4{,}0$ calculado usando únicamente 7 multiplicaciones.

   b) **Demostración por medio de inducción matemática que el número mínimo de multiplicaciones es n siendo n el grado del polinomio**
   Sea $P_0(x) = a_0x^0 = a_0$. El número de multiplicaciones para hallar $P_0(x_0)$ es igual a 0. Por lo que se cumple para el primer caso $k = 0$.
   Se asume por lo tanto que $P_k(x) = a_0 + a_1x + ... + a_kx^k$ y que $P_k(x_0) = a_0 + a_1x_0 + ... + a_kx_0^k$ tiene k multiplicaciones y que el método de Horner se expresa de la siguiente manera:

$$b_k = a_k$$
$$b_{k-1} = a_{k-1} + b_k * x_0$$
$$...$$
$$b_0 = a_0 + b_1 * x_0$$

Y se debe llegar a la forma $k+1$ del polinomio:

$$P_{k+1}(x_0) = a_0 + a_1 x_0 + ... + a_{k+1} x_0^{k+1} \tag{4}$$

o reescrito de otra forma:

$$P_{k+1}(x_0) = a_0 + x(a_1 + x(a_2 + ... + x(a_k + x a_{k+1}))) \tag{5}$$

Se reescribe $P_k(x_0)$ y se reemplaza a_k por b_k (Primera instrucción del método):

$$P_k(x_0) = a_0 + x_0(a_1 + ... + x_0(a_{k-1} + x_0 * b_k)) \tag{6}$$

Se añade una iteración al método de Horner para b_{k+1}, añadiendo una multiplicación más al método ($mult_{k+1} = k+1$)

$$b_{k+1} = a_{k+1}$$
$$b_k = a_k + b_{k+1} * x_0$$
$$...$$
$$b_0 = a_0 + b_1 * x_0$$

Y se reemplaza b_k en $P_k(x_0)$:

$$P_k(x_0) = a_0 + x_0(a_1 + ... + x_0(a_{k-1} + x_0 * (a_k + b_{k+1} * x_0)) \tag{7}$$

Despejando la ecuación se llega a la forma:

$$P_k(x_0) = a_0 + x_0(a_1 + ... + x_0(a_{k-1} + x_0 * (a_k + b_{k+1} * x_0)) \tag{8}$$

La cual es equivalente a $P_{k+1}(x_0)$, quedando demostrado el número de múltiplicaciones iguales a k, el grado del polinomio.

2. **La eficiencia de un algoritmo esta denotada por** $T(n)$
Dado el siguiente algoritmo:

procedure
Leer n
while $n > 0$ **do**
$d \leftarrow mod(n,2)$
$n \leftarrow fix(n/2)$
Mostrar d

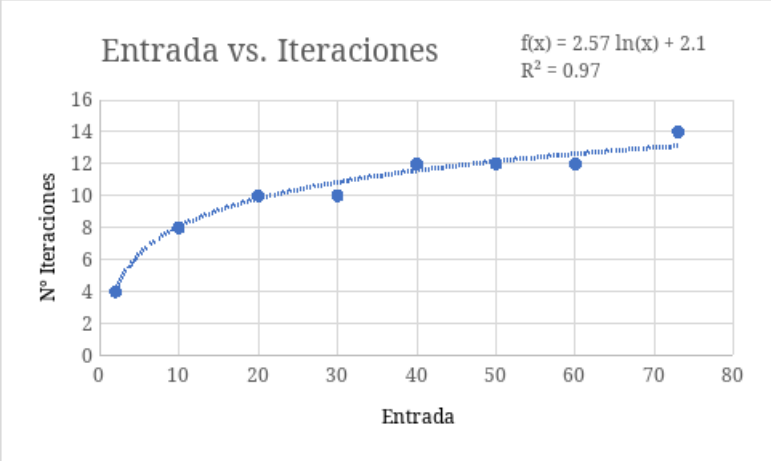
a) Evaluar el algoritmo con n = 73
procedure(73) = 1001001

b) **Encontrar T(n) y expresarlo en notación O( )**

Entrada=n	Num Operaciones
2	4
10	8
20	10
30	10
40	12
50	12
60	12
73	14

Tabla 1: Tabla de iteraciones del algoritmo dada una entrada n

Al graficar esta tabla se tiene:



Después de graficar los datos obtenidos encontramos que el polinomio de interpolación T(n) es:

$$y = 2{,}5699ln(x) + 2{,}0975 \tag{9}$$

Esto nos permite decir que el orden de eficiencia del algoritmo es $O(log_2(n))$ puesto que $ln(n) <= log_2(n)$

3. **Utilice R y el método de Newton para resolver el problema, muestre gráficamente cómo se comporta la convergencia a la solución**

Se tiene $\vec{R}(t) = (2cos(t), sen(t), 0)$ y el punto $P = (2, 1, 0)$. Se calcula la distancia entre el vector y el punto dado:

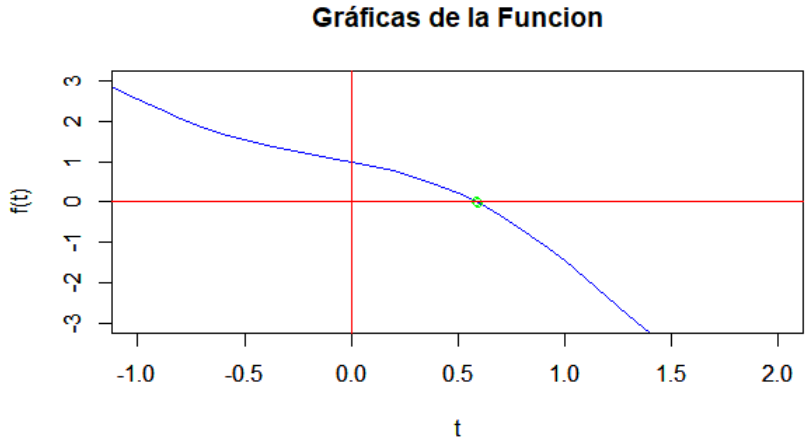
$$d = \sqrt{(2cos(t) - 2)^2 + (sen(t) - 1)^2} \tag{10}$$

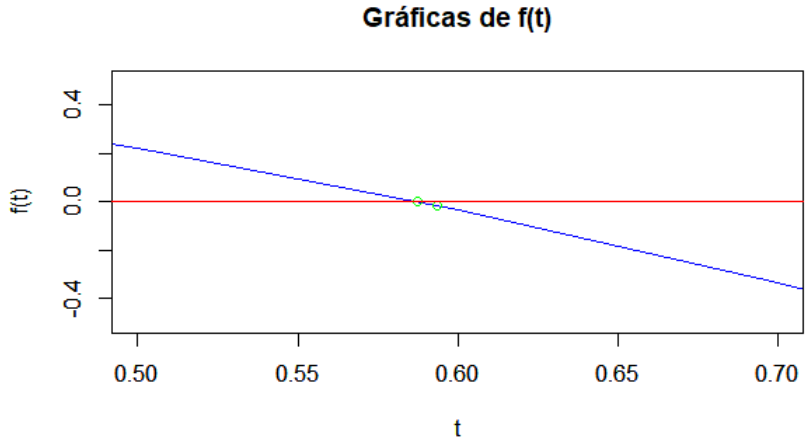
Se busca por lo tanto el punto mínimo de la distancia. Ya que es una función raíz solo tendrá un punto mínimo donde $d'(t) = 0$. Por aplicación del Método de Newton se tiene que:

$$f(t) = d'(t) = \frac{2cos(t)(sen(t) - 1) - 4sen(t)(2cos(t) - 2)}{2\sqrt{(2cos(t) - 2)^2 + (sen(t) - 1)^2}} = 0 \tag{11}$$

$$f(t) = 3sen(t)cos(t) - 4sen(t) + cos(t) = 0$$
$$f'(t) = -sin(t) - 4cos(t) + 3cos(2t)$$

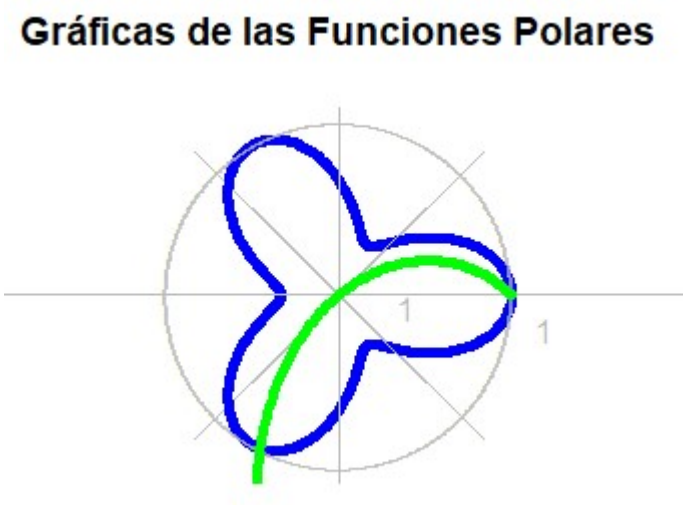
Después de tener $f(t)$ y $f\prime(t)$ se ingresan al Método de Newton con un intervalo de [0,1] y un error de 10^{-8}, Se grafica $f(t)$ y se le hace un seguimiento gráfico a la convergencia de la función.





Y se obtiene la siguiente salida: Número de Iteraciones = 4 ; *Resultado* $\approx$ 0,5872198, lo que nos indica que el tiempo en que el objeto se encuentra más cerca del punto es en t $\approx$ 0,5872.

4. **Encuentre la intersección entre las dos funciones polares dadas usando dós metodos diferentes**
   $r = 2 + cos(3t)$
   $r = 2 - e^t$
   Se grafican las dos funciones:

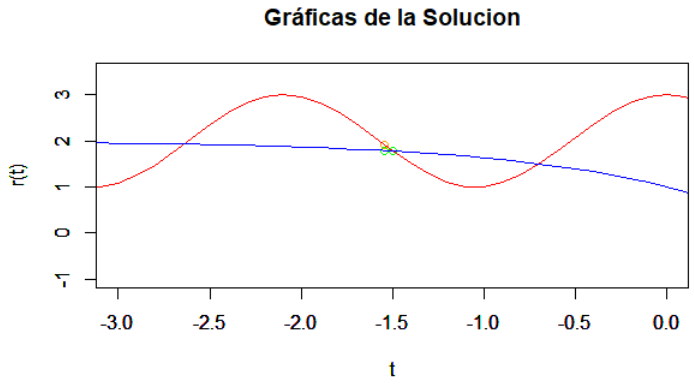


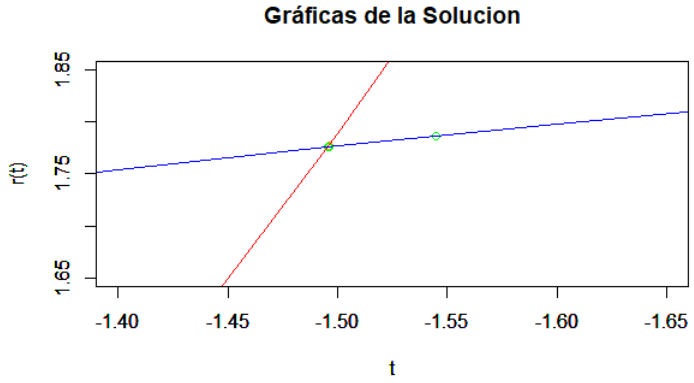
Se tiene las dos funciones. Para encontrar su punto de intersección se procede a igualar las funciones entre sí y se despejan en una única función igualándose a cero. Para el uso del método de Newton también se calculará la primera derivada

$$2 + cos(3t) = 2 - e^t$$
$$cos(3t) + e^t = 0$$

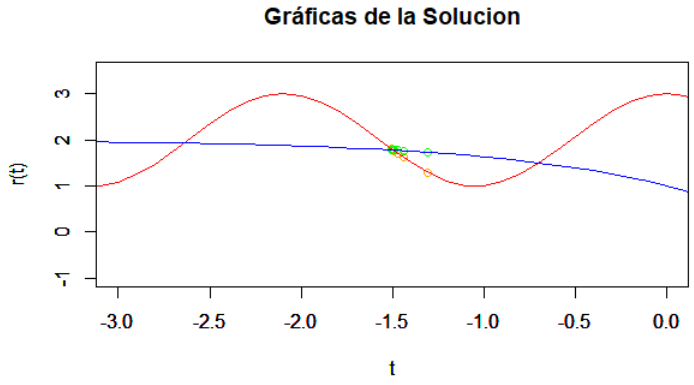
$$g(x) = cos(3t) + e^t = 0 \tag{12}$$
$$g'(x) = -sin(3t) + e^t \tag{13}$$

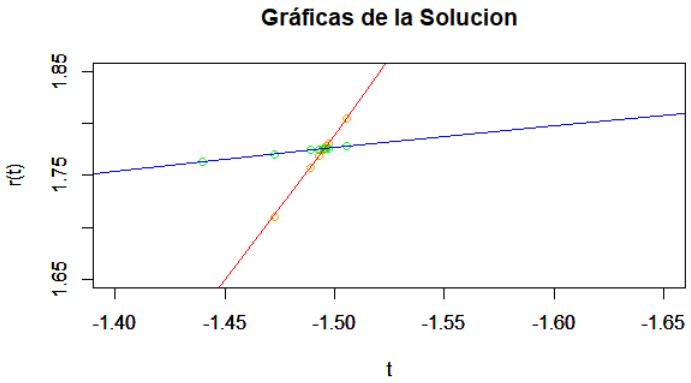
Teniendo la función y su derivada se aplica el método de Newton y de Bisección para resolverla en un intervalo dado y así encontrar su intersección:
Método de Newton. Dado un intervalo de $-\pi/3$ a $-\pi/2$ y un error máximo de 10^{-8} se obtiene que:
Entrada $\rightarrow$ Intervalo: $(-\pi/3, -\pi/2)$
Salida $\rightarrow$ $Iteraciones = 5, t = -1{,}495439 rad, r(t) = 1{,}77585$





Método de Bisección. Dado un intervalo de $-\pi/3$ a $-\pi/2$ y un error máximo de 10^{-8} se obtiene que:
Entrada $\rightarrow$ Intervalo: $(-\pi/3, -\pi/2)$
Salida $\rightarrow$ $Iteraciones = 22, t = -1{,}495439 rad, r(t) = 1{,}77585$





Para el método de la bisección, se usó el Método de Convergencia Acelerada o Proceso Δ^2 de Aitken para acelerar la convergencia de los valores hallados. Este proceso parte de la secuencia de valores hallados y utilizando la fórmula siguiente, se hallan unos valores que convergen más rápidamente a la solución:

$$\hat{x}_{n+2} = x_{n+2} - \frac{(x_{n+2} - x_{n+1})^2}{x_{n+2} - 2x_{n+1} + x_n} \tag{14}$$

Así, se pueden calcular unos valores que convergen más aceleradamente a la solución:

Biseccion	Aitken
-1.308997	-1.570796
-1.439897	-1.48353
-1.505346	-1.48353
-1.472622	-1.505346
-1.488984	-1.494438
-1.497165	-1.494438
-1.493075	-1.497165
-1.49512	-1.495802
-1.496143	-1.49512
-1.495631	-1.495461
-1.495376	-1.495461
-1.495503	-1.495376
-1.49544	-1.495418
-1.495408	-1.49544
-1.495424	-1.49544
-1.495432	-1.49544
-1.495436	-1.49544
-1.495438	-1.49544
-1.495439	-1.49544
-1.495439	
-1.495439	

Tabla 1: Tabla de iteraciones del algoritmo dada una entrada n

5. **Encuentre el error de redondeo para x = 0.4 al convertirse al estándar IEEE 754 para 64bit**

   a) Según la norma IEEE-754 un número infinito (+inf o -inf) se representa de forma que los bits correspondientes al exponente mínimo se colocan todos en 1 (8 para 32 bits, 11 para 64 bits) y el bit del signo indicará el signo del infinito.

   b) En el proceso de redondeo se usa el valor de la siguiente para obtener un resultado más aproximado del valor. El corte (truncamiento) corta el número de cifras decimales sin tener en cuenta las siguientes para el valor de la última cifra. Se puede decir que el redondeo es más exacto en el valor decimal final, sin embargo, un redondeo a n decimales y corte a n decimales podrán ser los mismos dependiendo de la regla usada para el redondeo.

   c) $x = 0{,}4$

   Según el proceso para convertir un número decimal a su representación en binario el valor será:

$$\begin{gathered} 0{,}4 * 2 = 0{,}8 \rightarrow 0 \\ 0{,}8 * 2 = 1{,}6 \rightarrow 1 \\ 0{,}6 * 2 = 1{,}2 \rightarrow 1 \\ 0{,}2 * 2 = 0{,}4 \rightarrow 0 \\ 0{,}8 * 2 = 1{,}6 \rightarrow 1 \\ \dots \\ = 0{,}01100110011001100110011... \\ = 1{,}10011001100110011001100110011... * 2^{-2} \end{gathered}$$

   Sin embargo, el carácter de este número es periódico infinito, por lo que para poder ser representado en un computador se utiliza la norma IEEE 754 para representaciones de números en coma flotante.
   Usando un tamaño de número de 64 bits, divididos en 1 bit para el signo, 11 bits para el exponente mínimo (Biased Exponent) y 52 bits para la precisión de la fracción (Fraction) se encuentra que:

   $Signo = 0$
   $BiasedExponent = (1021)_{10} = (01111111101)_2$
   $Fraccin = (1001100110011001100110011001100110011001100110011001)$

   Por lo tanto, se tiene que el valor de $fl(0{,}4)$ es igual en el estandar IEEE 754 a:
   $(0{,}4)_{10} \approx (0011111111011001100110011001100...)_2 = (0{,}3999999999999999966693309261245)_{10}$

   d) Para encontrar el error de conversión al estándar IEEE 754, se usará la fórmula:

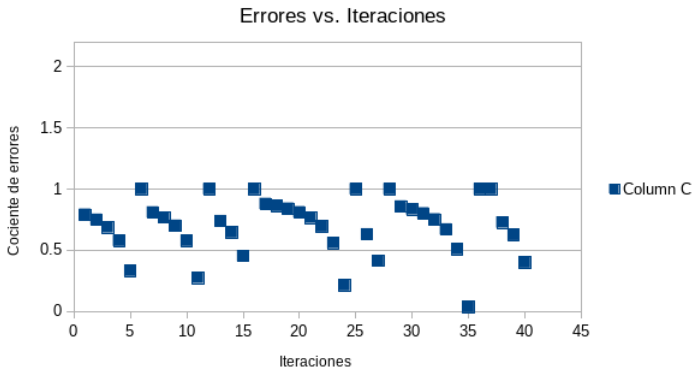
$$\begin{gathered} \frac{|fl(x)-x|}{|x|} \leq \frac{e_{maq}}{2} \\ \frac{|0{,}3999999999999999966693309261245-0{,}4|}{|0{,}4|} \leq \frac{2^{-52}}{2} \\ \frac{|0{,}3999999999999999966693309261245-0{,}4|}{|0{,}4|} \leq 2^{-53} \\ 8{,}32667268468875 * 10^{-17} \leq 1{,}110223 * 10^{-16} \end{gathered}$$

   El error de redondeo es por lo tanto $\approx 8{,}327 * 10^{-17}$ o $8{,}327 * 10^{-15}$ %.

6. *a*) **Encontrar un algoritmo iterativo para calcular la raíz n-ésima de un número**
Se tiene la función $\sqrt[n]{A} = x$ que se puede reescribir como $x^n - A = 0$. Así, se puede usar el Método de Newton para encontrar la raíz de la función y para hallar una función iterativa específica para la raíz n-ésima de un número. Sea el método iterativo de Newton un método que reciba la función $f(x) = x^n - A = 0$ y su derivada $f'(x) = nx^{n-1}$ Por lo tanto:

$$
\begin{aligned}
x_{n+1} &= x_n - \tfrac{f(x_n)}{f'(x_n)} \\
x_{n+1} &= x_n - \tfrac{x_n^n - A}{n x_n^{n-1}} \\
x_{n+1} &= x_n - \tfrac{1}{n}(x_n - \tfrac{A}{x_n^{n-1}}) \\
x_{n+1} &= x_n + \frac{1}{n}(\frac{A}{x_n^{n-1}} - x_n) \qquad (15) \\
&\textit{hasta que} \mid x_{n+1} - x_n \mid < E
\end{aligned}
$$

b) **Análisis de método para encontrar raíces** Para que la raíz exista, el intervalo tomado $[a, b]$ debe ser elegido de tal forma que: $f(a)f(b) < 0$



Gráfica de órdenes de convergencia

Observando la gráfica, se nota que el comportamiento de la relación entre errores es un poco variante, aunque se observa un comportamiento relativamente lineal. Por lo tanto, el orden de convergencia del método es una **Convergencia Lineal**