Reto 1

Estiben Giraldo y Julio Andrés Mejía

Febrero 23 2020

Contents

1 Evaluación de un Polinomio

1.1 Aplicación del método de Horner a un polinomio derivado

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 probar el teorema de Horner en un polinomio derivado se utiliza un algoritmo que evalua el polinomio de la forma que muestra el método 2, para este paso se utilizó el polinomio $P(x) = 2x^4 - 3x^2 + 3x + 4$ cuya derivada es $P\prime(x) = 8x^3 - 6x + 3$, como se muestra en la *Figura 1* sus respectivas gráficas dentro del intervalo [-5,5]. Para probar el algoritmo se realizaron varias pruebas con el número π para evaluar la cantidad de cifras que el computador es capaz de manejar que es el punto flotante, es decir, 2^{32} que en otras palabras es 16 cifras signicativas para representar tanto la derivada del polinomio como para representar el resultado calculado en un punto x_0.

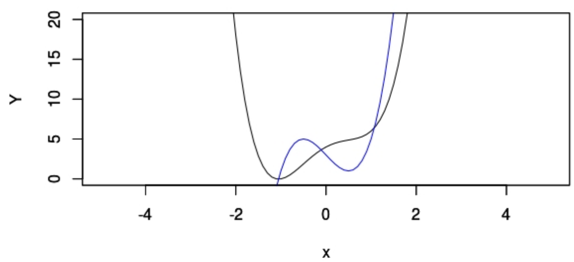

Figure 1: Polinomio y derivada

Los resultados obtenidos son los que se presentan en la siguiente tabla:

x_0	y'
$\pi/4$	2.163395604652787
$\pi/6$	1.006787964199089
$\pi/8$	1.12827858293734
$\pi/12$	1.572751250428714
$\pi/14$	1.744000456567347

Table 1: Resultados

Como se puede observar el algoritmo tiene en cuenta el manejo del error del computador, ya que muchas veces al realizar operaciones como suma, resta, multiplicación o división tiende a truncar algunos valores. Sin embargo, al haber tenido en cuenta la unidad de redondeo o precisión del computador se logró contrarestar dicho error.

1.2 Aplicación con números complejos

Ahora bien, como se menciona en la sección anterior el método de Horner se utiliza para el cálculo de la derivada de un polinomio en un punto x_0, pero, qué pasa si también deseamos realizar el cálculo de dicha derivada del polinomio en un número complejo. Para este caso, se implementó el mismo algoritmo pero en este caso se utilizó la función *complex()* de R que permite instanciar un número y su parte imaginaria. Un número complejo es de la forma $z = a + bi$ donde a representa su parte real y bi representa su parte imaginaria. Es por esto que la siguiente tabla muestra diferentes pruebas realizadas al algoritmo con diferentes números complejos. En este caso se utilizó el mismo polinomio $P(x) = 2x^4 - 3x^2 + 3x + 4$ para calcular su derivada en un número complejo

Como se puede evidenciar en los resultados obtenidos el manejo de los números complejos en R mediante esta función facilita el manejo también de

x_0	y'
$2+5i$	$-1145-550i$
$3+\sqrt{2}i$	$57+274.3574311003805i$
$3+9\sqrt{2}i$	$-11463-13822.52335863463i$
$3+\pi\sqrt{2}i$	$-1220.223033756868+231.413465013695i$
$e+\sqrt{3}i$	$-48.3416872343042+255.1957697727234i$

Table 2: Resultados

los número que el computador es capaz de representar, porque también tiene en cuenta la precisión doble, es decir que el computador pueda representar $[2^{-32}, 2^{32}]$ posibles numeros, es por esto que en el algoritmo se procuró al igual que la sección anterior reducir error de rendondeo. Por lo tanto la aproximación obtenida es bastante buena con precisión de 16 bits.

2 Optima Aproximación Polinómica

2.1 Aproximación 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 contiunas en [a,b] y sea $x, x_0 \in [a, b]$ Entonces,

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

Donde:

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

Ahora, para realizar la aproximación de Taylor para la función $sin(x)$ en los intervalos $[\pi/64, \pi/64]$ se puede aproximar de la forma:

$$sin(x) \approx \sum_{k=0}^{n} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$$

Esta aproximación se realizó dentro del intervalo anteriormente mencionado tomando como punto $\pi/128$. Ahora bien, para verificar la eficiencia del algoritmo se realizaron dos procedimientos. El primero fue analizar el error relativo y el segundo la cantidad de iteraciones. La primera conclusión a la cual se llegó fue que la cantidad de iteraciones depende del grado y que esta es *n+1* siendo n el grado del polinomio deseado. Al haber realizado 10 pruebas se procedió a graficar el error relativo obteniendo lo siguiente:

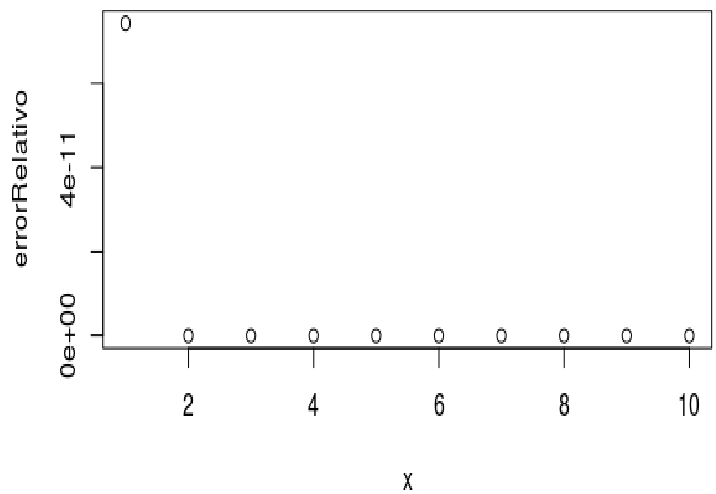

Figure 2: Error relativo del algoritmo

Como se puede evidenciar en la figura en el eje X se encuentra el grado del polinomio y en el eje Y el error relativo esperado en $\pi/128$, así mismo se decidió graficar el polinomio de grado 4 para analizar la aproximación obtenida.

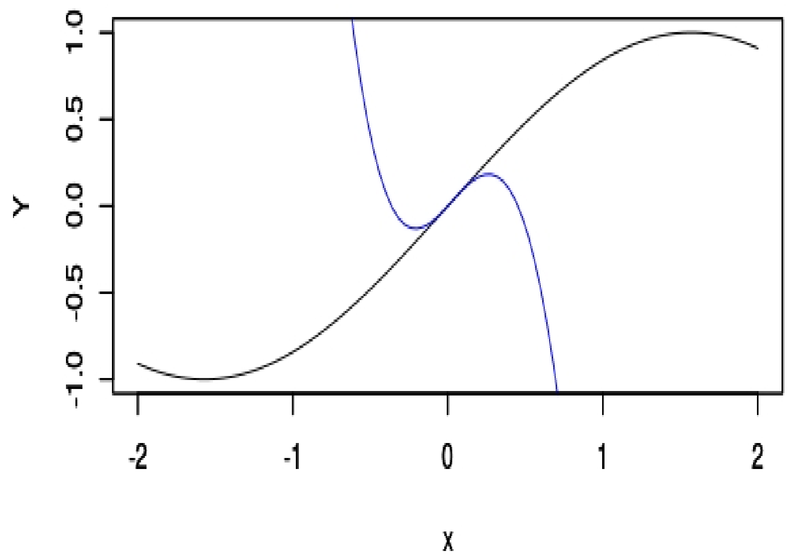

Figure 3: Polinomio grado 4 de $sin(x)$

En conlusión esta aproximación resulta bastante buena, ya que dentro del intervalo deseado el polinomio es bastante preciso teniendo también en cuenta la precisión doble de la máquina. Vale la pena aclarar que es posible que entre más grados de polinomio se utilice mejor aproximación se obtendrá.

2.2 Método de Remez

El metodo de Remez es una mejora al metodo de Horner donde tiene en cuenta las cancelaciones del algoritmo y se tiente en cuenta la aproximacion a los minimos y maximos de la ecuacion de la que se este hablando, para aplicar el metodo de Remez en la funcion de $sin(x)$ en el intervalo $[-\pi/64, \pi/64]$ es necesario comparar la grafica de la funcion Seno con el polinomio de Remez que se esta aplicando, el cual es un polinomio de grado 3, para resolverlo, graficarlo y compararlo, primero es necesario plantear un sistema de ecuaciones lineales y darle una unica solucion, luego de esto se plantearan una serie de puntos en el plano los cuales seran la guia de comparacion en el plano.

Para cuando se halla graficado el polinomio y se tenga la comparacion entre la grafica original de Seno y el polinomio de Remez podemos determinar tanto el error relativo como el error absoluto, haciendo la diferencia con los puntos escogidos entre las funciones, Seno y el polinomio planteado, para despues con el resultado dividirlo en los puntos del polinomio y multiplicarlo por 100:

Absoluto $= |sin(x) - (a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3)|$
Relativo $= |(sin(x) - (a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3))/(a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3)| * 100$

Los resultados obtenidos son los siguientes:

Error Absoluto Promedio $= 4.004999 * 10^{-}10$
Error Relativo Promedio $= 2.980809 * 10^{-}6$
Numero total de operaciones $= 8$

Sus respectivas graficas son:

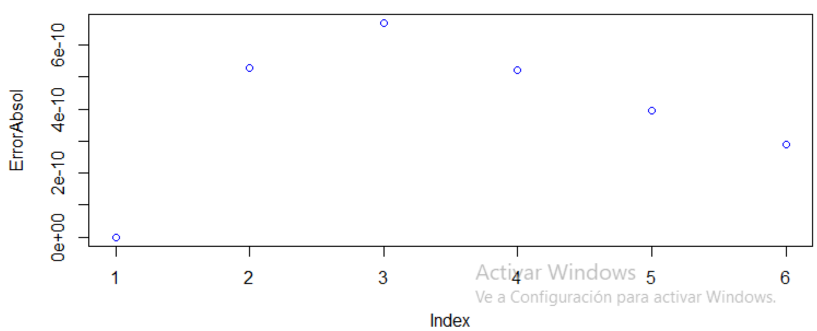

Figure 4: Error Absoluto

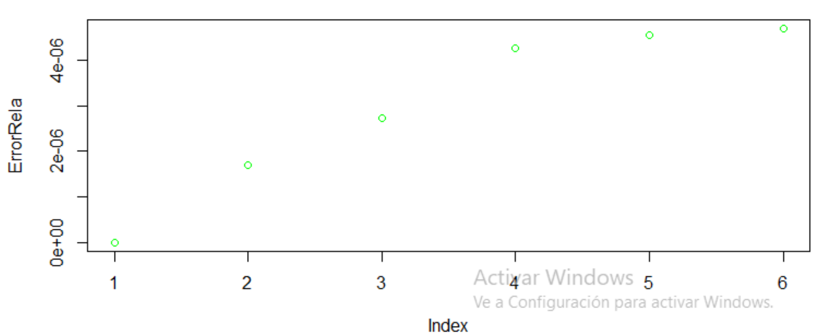

Figure 5: Error Relativo

Las graficas resultantes de la compracion $sin(x)$ con el polinomio de Remez son:

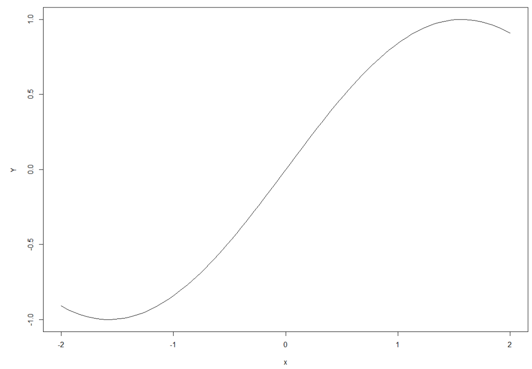

Figure 6: $sin(x)$

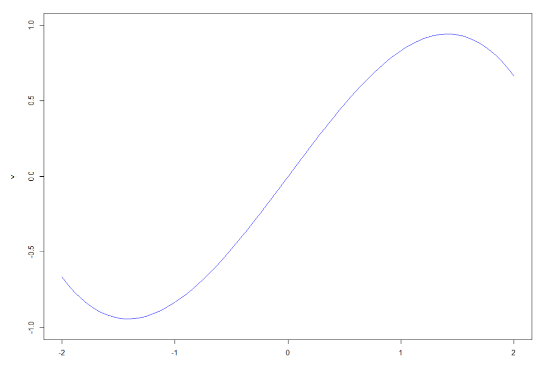

Figure 7: Polinomio de Remez

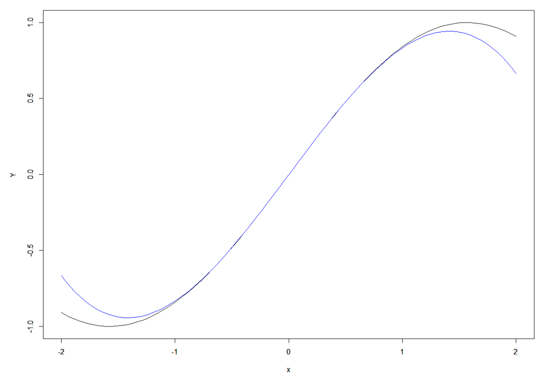

Figure 8: Comprativa entre $sin(x)$ y el Polinomio de Remez

References

[1] Vega,M.A., *Aproximación Minimax, Remez , Tchebyshev*, From: https://www.studocu.com/es/document/universidad-de-santiago-de-chile/matematicas-i/apuntes/15-aproximacion-minimax-remez-tchebyshev/2541506/view, Chile, Santiago, 2015.

[2] Aarts,R.M.; Bond,C.; Mendelsohn,P. y Weisstein, E.W., *Remez Algorithm.*, From: http://mathworld.wolfram.com/RemezAlgorithm.html

[3] Mora,F.W.,*Introducción a los MétodosNuméricos. Implementaciones en R*, primera edicón,Revista digital Matemática, Educación e Internet, From: https://tecdigital.tec.ac.cr/revistamatematica/Libros/WMora-MetodosNumericos/2017-Principal-MetodosNumericos-con-R.pdf, 2015.

[4] Hurtado,N.A. y Domínguez,S.F.,*Métodos Numéricos Aplicados a la Ingeniería*, primera edicón EBook, Patria, México, 2014