Reto 1

Andres García, Andres Ramirez, Felipe Becerra, Julian Rizo

13 - September 2020

1 Evaluación de las raíces de un Polinomio

Evaluar las raıces de un polinomio implica varios desafios, aun para el software profesional. Como se requiere poder evaluar las posibles raıces del polinomio, es necesario dedicarle un momento a los detalles como, las derivadas evaluadas en valores x0. Problema: Sea P (x) = a0xn + a1xn1 + ... + an un polinomio entonces, implementar en R y/o Python una modificacion del metodo de Horner que evalue de manera eficiente f' (x0) y f" (x0) la primera y segunda derivada en cualquier punto

1. Nota1: tenga en cuenta la precision de la computadora o unidad de redondeo. 2. Nota2: Tenga en cuenta que un polinomio puede tener como vector de coeficientes con numeros complejos y mas aun las raıces pueden ser de la forma + i

Problema Utilizar los resultados anteriores y el algoritmo de Laguerre para obtener un metodo de, Newton-Horner, de convergencia cuadratica en el que el algoritmo de Newton reemplaza al de Laguerre.

i Aplicar para el polinomio

$$x^4 - 9x^2 - 5x^3 + 155x - 250 \tag{1}$$

el algoritmo creado

ii Compara con el algoritmo de Laguerre

Solución 1

El algoritmo de Horner se basa en la factorizacion de la expresion a evaluar en un punto cualquiera, de esta manera:[3]

- Rewrite $p(x)$
  - $p(x) = a + bx + cx^2 + dx^3$
  - $p(x) = a + bx + x(cx + dx^2)$
  - $p(x) = a + x\big(b + x(c + dx)\big)$
  - $p(x) = a + x \cdot \big(b + x \cdot (c + d \cdot x)\big)$ ← 3 Adiciones, 3 Multiplicaciones
- In terms of a_n
  - $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_nx^n$
  - $p(x) = a_0x^0 + a_1x^1 + a_2x^2 + a_3x^3 + \cdots + a_nx^n = \sum_{i=0}^{n} a_ix^i$
  - $p(x) = a_0 + x\big(a_1 + x(a_2 + \cdots + x(a_{n-1} + xa_n) \dots)\big)$
  - $p(x) = p_0(x)$ where $p_i(x) = \begin{cases} a_i + x \cdot p_{i+1}(x) & \text{if } i < n \\ a_n & \text{if } i = n \end{cases}$

Para encontrar el valor de la funcion en un punto solo faltaría reemplazar x por el valor deseado, este proceso tambien se puede ver como una división sintetica, graficamente se puede visualizar de esta manera con **un ejemplo diferente al problema del reto**: [3]

$$p(x) = x^5 - 8x^4 - 72x^3 + 382x^2 + 727x - 2310$$

Solucionar $p(2)$

2	1	−8	−72	382	727	−2310
		2	−12	−168	428	2310
	1	−6	−84	214	1155	$\boxed{0}$

$$p(2) = 0$$

Sabiendo como se comporta el algoritmo de Horner a continuación se resuelve el problema dado en el reto:
Partiendo del polinomio especificado, calculamos la derivada la cual da como resultado:

$$4x^3 - 15x^2 - 18x + 155 \tag{2}$$

posteriormente lo almacenamos en un arreglo llamado coeficientesDerivada, luego usamos este arreglo en donde hacemos un ciclo con el número de iteraciones igual a la cantidad de coeficientes, que nos dará como resultado en cada iteración el coeficiente en orden descendente del nuevo polinomio formado por la **regla de ruffini**; dentro de ese ciclo evaluamos el valor de x=3 y utilizamos la siguiente formula:

$$res = res * x + coeficientesDerivada[j] \tag{3}$$

Con base en estas operaciones se obtuvieron los siguientes valores:

	ResultadosPrimeraDerivada
1	4
2	-3
3	-27
4	74

En la tabla se muestra el numero de iteraciones y el resultado de cada una, en donde el valor de la función evaluada en 3 es la ultima que en este caso es 74. Tambien cabe resaltar que los resultados en las anteriores iteraciones no son en vano ya que como antes se mencionó la **regla de ruffini** nos permite encontrar la forma factorizada del polinomio, en este caso la primera derivada de la función dada, usando los coeficientes hallados el usar el metodo de Horner, dandonos como resultado una expresión así:

$$(4x^2 - 3x - 27)(x - 3) + 74 \tag{4}$$

Se puede observar que se usan los coeficientes de la tabla "ResultadosPrimeraDerivada" y el valor de la función evaluada en el punto se usa como residuo en la expresión. Lo antes expuesto se prueba a continuación utilizando la herramienta WolframAlpha

Input:

$$(4x^2 - 3x - 27)(x - 3) + 74$$

Alternate forms:

$$4x^3 - 15x^2 - 18x + 155$$

Derivative:

$$\frac{d}{dx}\left(x^4 - 5x^3 - 9x^2 + 155x - 250\right) = 4x^3 - 15x^2 - 18x + 155$$

Posteriormente se realizó el mismo proceso para hallar el valor de la segunda derivada de la función en el punto escogido. Esto nos arrojo los siguientes resultados

	ResultadosSegundaDerivada
1	12
2	6
3	0

Tambien se pudo deducir una forma factorizada de la segunda derivada del polinomio dado con los resultados obtenidos, así:

$$(12x - 6)(x - 3) + 0 \tag{5}$$

Y a continuación la prueba:

Input:

$(12x + 6)(x - 3)$

Expanded form:

$12x^2 - 30x - 18$

Derivative:

$$\frac{d^2}{dx^2}\left(x^4 - 5x^3 - 9x^2 + 155x - 250\right) = 12x^2 - 30x - 18$$

Solución 1.2

Descripción método Laguerre y Newton: Para la siguiente solución se utiliza en primer lugar el método de Newton, el cual encuentra las raíces de un polinomio utilizando el polinomio y su derivada, este método presenta convergencia cuadrática. En segundo lugar se utiliza el método de Laguerre, el cual para hallar las raices utiliza la primera y la segunda derivada, este método utiliza convergencia cubica. A continuación se presentan las fórmulas utilizadas para calcular las raices de cada uno de los polinomios.

Método de Newton:

$$x = x_n - \frac{p(x_n)}{p'(x_n)} \tag{6}$$

Método de Laguerre:

$$\lambda = \frac{n}{(G \pm \sqrt{(n+1)(nH - G^2)})} \tag{7}$$

Manipulación de números complejos: Como algoritmo fue implementado en R, ahí existe la posibilidad de manejar números complejos de la siguiente forma; 2+9i, donde se puede ver una parte real y una imaginaria. En este orden de ideas, las raices que devuelven los métodos son las siguientes: Raiz: 2+0i Raiz: -5+0i Raiz: 4+3i Raiz: 4-3i

Procedimiento:

2 Algoritmo Brent

El algoritmo de Brent, utiliza en cada punto lo mas conveniente de las estrategias del de la biseccion y del de la secante (o Muller). En su acepcion moderna fue formulado en 1973 por Richard Peirce Brent, Australia, 1946.El metodo de Brent, tambien conocido como zeroin, ha sido el metodo mas popular para encontrar ceros de funciones desde que se desarrollo en 1972. Este metodo suele converger muy rapidamente a cero; para las funciones dificiles ocasionales que se encuentran en la practica.

Problema:Aplicar el algoritmo de Brent para encontrar las raıces del polinomio:

$$f(x) = (x^3)(2x^2) + (4x/3)(8/27) \tag{8}$$

Solución

El algoritmo de brent es uno de los mas eficientes para hallar la raiz de una función, ya que usa a conveniencia los metodos de bisección y de la secante, esto tambien le permite escapar de las limitaciones de estos anteriores mencionados, complementandolos en sus fallos, un ejemplo de su eficiencia en un polinomio diferente al problema tratado en el reto sería:

$$f(x) = x^3 - x^2 - x - 1$$

Starting values 1 and 2

$$\epsilon = 10^{-7}$$

Method	Steps	Root Found
Bisection	24	1.8392867743968964
Secant	6	1.839286755226924
Dekker	7	1.8392867552141612
IQI - $1, \frac{3}{2}, 2$	5	1.8392867552230825
Brent	6	1.8392867552141612

Partiendo de la función problema, se hizo uso de el algoritmo de Brent de la libreria "Pracma" de RStudio, la cual recibe como parametros la función a la cual se le quiere encontrar las raices, un intervalo, y como argumentos opcionales se puede poner un numero maximo de iteraciones y una tolerancia. Este algoritmo retorna:
root: Valor de la raiz
f.root: Valor de la función evaluada en la raiz
f.calls: Numero de iteraciones
estim.prec: Precision estimada

```
> f <- function(x) x^3-2*x^2+(4*x/3)-(8/27)
>
> pracma::brent(f, 0.5, 1)
$root
[1] 0.6667608

$f.root
[1] 8.341106e-13

$f.calls
[1] 27

$estim.prec
[1] 3.056534e-05
```

3 Optima Aproximación Polinomica

La implementación de punto flotante de una función en un intervalo a menudo se reduce a una aproximación polinómica, siendo el polinomio típicamente proporcionado por el algoritmo Remez. Sin embargo, la evaluación de coma flotante de un polinomio de Remez a veces conduce a cancelaciones catastróficas. El algoritmo Remez es una metodología para localizar la aproximación racional minimax a una función. La cancelaciones que también hay que considerar en el caso de del método de Horner, suceden cuando algunos de los coeficientes polinómicos son de magnitud muy pequeña con respecto a otros. En este caso, es mejor forzar estos coeficientes a cero, lo que también reduce el recuento de operaciones. Esta técnica, usada clásicamente 1 de 2 para funciones pares o impares, puede generalizarse a una clase de funciones mucho más grande.

Problema: Aplicar esta tecnica de aproximacion polinómica, para :

$$f(x) = esinx - cosx2 \tag{9}$$

en el intervalo

$$[-2^{-8}; 2^{-8}] \tag{10}$$

con una precisión deseada doble - doble y un error menor de

$$2^{-90} \tag{11}$$

y comparar con la aproximación de Taylor.

Solución

Ya en Taylor el método que escogimos fue por medio de la librería Pracma, la cual nos ofrece la función Taylor (F,x0,n). Traducida como:
F = Función Diferenciable

X0: El punto de inicio de la expansión
N: El grado del polinomio que se desea obtener.
Para lograr determinar el grado del polinomio concordamos como grupo usar el polinomio de grado 3 debido a que su número de operaciones en el punto es mayor y el error no disminuía de manera relevante. Además de que el intervalo escogido, por consecuente decidimos trabajar con un polinomio no muy grande:

Para este caso se realizó primero la aproximación de Taylor para el polinomio para

$$f(x) = e^{(sinxcosx^2)} \tag{12}$$

en el intervalo

$$[-2^8; 2^8] \tag{13}$$

con una precisión deseada doble- doble y un error menor de

$$2^{-90} \tag{14}$$

.

Ya en Taylor el método que escogimos fue por medio de la librería Pracma, la cual nos ofrece la función Taylor (F,x0,n). Traducida como: F=Función Diferenciable
X0:El punto de inicio de la expansión
N: El grado del polinomio que se desea obtener.
Para lograr determinar el grado del polinomio concordamos como grupo usar el polinomio de grado 3 debido a que su número de operaciones en el punto es mayor y el error no disminuía de manera relevante. Además de que el intervalo escogido, por consecuente decidimos trabajar con un polinomio no muy grande:

$$p3(x) = ((1/3) * x^3) + x \tag{15}$$

-Para poder implementar el algoritmo de Remez se debe tener en cuenta una serie de pasos.
1. Se obtienen los nodos de Chebyshew que son susceptibles al fenómeno de Runge que es una larga oscilación que ocurren al final de los intervalos. Se deben evaluar en nuestro caso en la función de

$$f(x) = e^{sinxcosx^2} \tag{16}$$

. La fórmula para calcular los números de Cheviv es :

$$x_i = \frac{1}{2}\Big(a+b\Big) + \frac{1}{2}\left(b-a\right) \cdot \cos\left(\frac{2i-1}{2n}\pi\right)$$

Siendo a,b el intervalo en el cual se requieren calcular los nodos , i la iteración actual del nodo , y n el número total de nodos a calcular. 2. Tomamos

n=3 debido a que nos daba un error bueno lo suficientemente bajo y nos da un buen rango de eficiencia.
3. Después de que calculamos los Nodos de Chebyshev dentro de un sistema de ecuaciones lineal especial :
Para hallar los coeficientes del polinomio interpolado, el algoritmo de Remez sugiere un sistema de ecuaciones lineales utilizando los nodos y los valores resultantes que se calcularon en la función. En la siguiente grafica se genera el sistema de ecuaciones donde bi son coeficientes del polinomio minmax interpolado y E es su respectivo error.

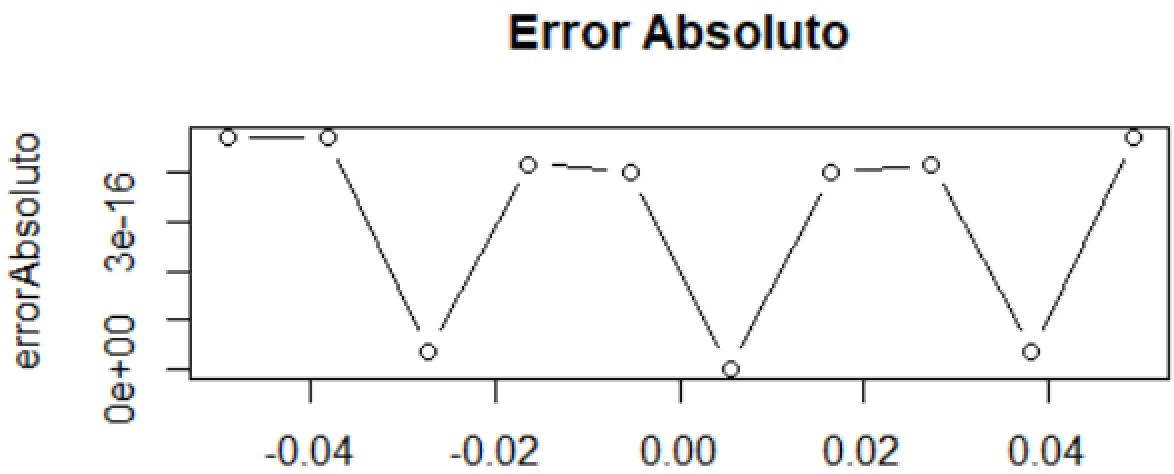
$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & E \\ 1 & x_1 & x_1^2 & \cdots & -E \\ \vdots & & & & \\ 1 & x_{n+1} & x_{n+1}^2 & \cdots & (-1)^i E \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \\ E \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ \vdots \\ f(x_n + 1) \end{pmatrix}$$

La matriz es de tamaño (n+2)(n+2). Esta matriz a su vez cuenta con un error menos a

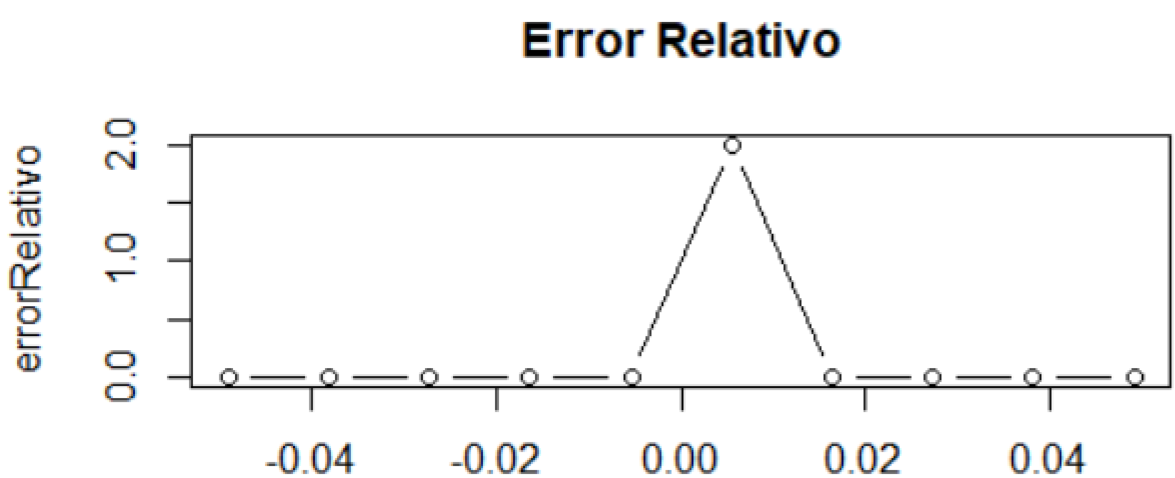
$$2^-9 \tag{17}$$

4. Se resuelve el sistema de ecuaciones y la ecuación resultante es la que se compara con para

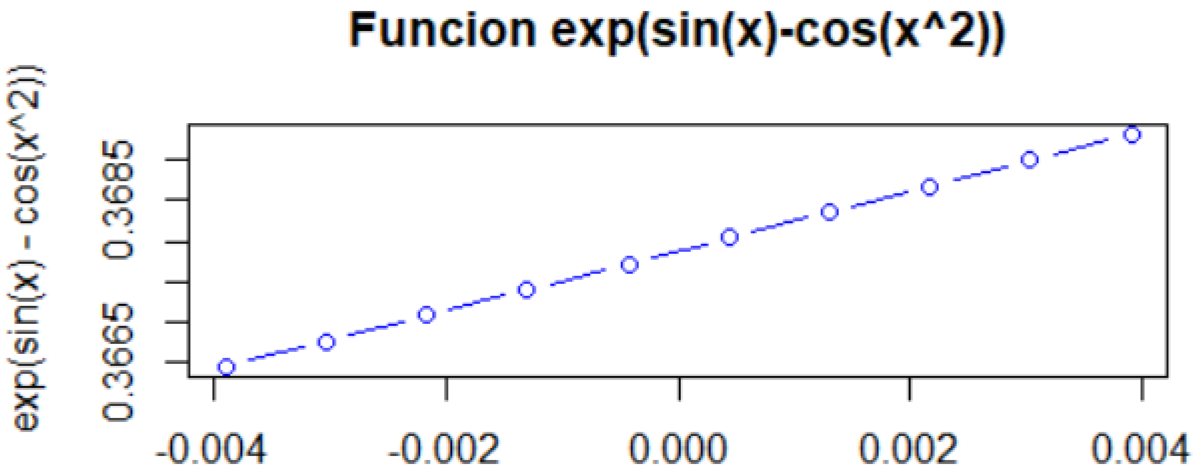
$$f(x) = e^{sinxcosx^2} \tag{18}$$



Error Absoluto
errorAbsoluto
0e+00
3e-16
-0.04
-0.02
0.00
0.02
0.04



Error Relativo
errorRelativo
0.0
1.0
2.0
-0.04
-0.02
0.00
0.02
0.04



Esta última grafica muestra la relación entre la función con la aproximación que realizamos.

4 Conclusiones

- El metodo de Horner halla el valor de una función evaluada en un pounto de una forma mas eficiente.[1]
- Al realizar el metodo de Horner se halla una forma de factorizacion de la funcion que se esta evaluando usando la regla de Ruffini (cociente)(binomio)+residuo, siendo el residuo el valor de la función evaluada en el punto.[3]
- El metodo de Brent selecciona de acuerdo al polinomio que se trabaje la forma de buscar las raices, usando los metodos de biseccion y de la secante, para no tener los mismos problemas que estos tienen por separado.[2]

References

[1] W. G. Horner. *A New Method of Solving Numerical Equations of All Orders, by Continuous Approximation.* Royal Society, 1819.

[2] Oscar Veliz. Brent's method, 2018.

[3] Oscar Veliz. Horner's method, 2018.