

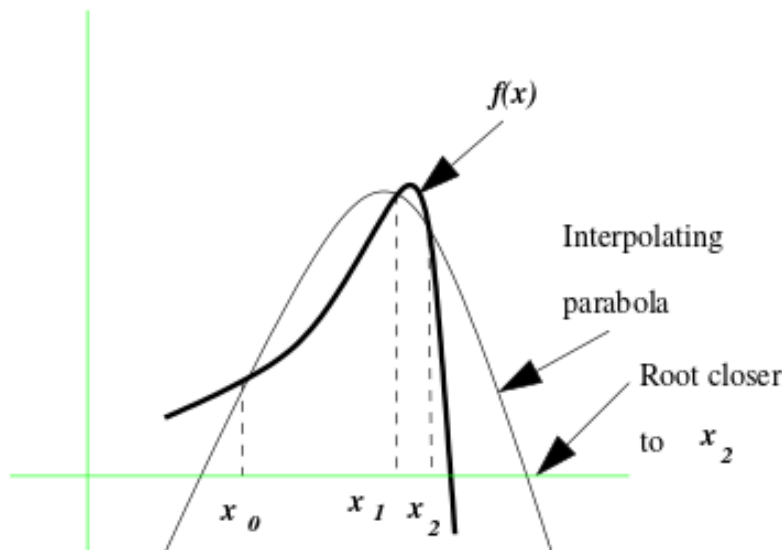
Tarea 3: Ceros de polinomios

Elaborado por Giselt Parra
Lunes, 2 de Noviembre de 2020.

Propuesta: Método de Müller usando Interpolación Polinómica de Lagrange Simplificado

Para la implementación de esta propuesta se propone combinar las ventajas de la técnica de deflación de Horner tanto para evaluar puntos en los polinomios así como para generar polinomios de menor grado cuando se evalúa con una raíz así como también se implementa el método de Müller que aproxima raíces de una manera similar al método de la Secante con la diferencia de que se trazan parábolas en tres puntos del polinomio. Siendo éstas representadas por ecuaciones cuadráticas, el método nos permite hallar raíces imaginarias aún cuando los iteradores iniciales son reales.

El método de Müller es un algoritmo que consiste en interpolar el polinomio p en tres puntos resultando así una ecuación cuadrática que representa la parábola que pasa por estos tres puntos y que aproximará a una raíz del polinomio p cuando corta con el eje x en la raíz de la parábola más cercano al tercer y último iterador calculado. Este punto de corte es un posible cero del polinomio que estamos evaluando, por lo que se hacen las comparaciones pertinentes respecto a su altura y diferencia con el iterador anterior para verificar si es una buena aproximación. De serlo, se considerará este punto una raíz y se procede a realizar una deflación tomando como valor a evaluar dicho punto que dará como resultado un nuevo polinomio de grado menor y así continuar la búsqueda de las siguientes raíces. En el caso contrario, este punto se tomará como nuevo iterador y se continuará operando de la misma forma descrita hasta hallar una buena aproximación.



Construcción de la parábola en el método de Müller ilustrado

Para hallar el polinomio interpolante de segundo grado que usaremos en cada iteración para el método de Müller, usaremos la base Lagrange que consiste en construir una base

$$B_L = \{l_0(x), l_1(x), l_2(x)\}$$

donde cada l_i se construye como

$$\begin{aligned} l_0(x) &= \prod_{j=0, j \neq 0}^2 \frac{x - x_j}{x_0 - x_j} = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \\ l_1(x) &= \prod_{j=0, j \neq 1}^2 \frac{x - x_j}{x_1 - x_j} = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \\ l_2(x) &= \prod_{j=0, j \neq 2}^2 \frac{x - x_j}{x_2 - x_j} = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

Ya obtenido los elementos que conforman la base, se escribe el polinomio a generar como la siguiente combinación lineal

$$p_n(x) = b_0 l_0(x) + b_1 l_1(x) + b_2 l_2(x)$$

Cómo se puede ver, la base de Lagrange cumple que

$$l_i(x_j) = \begin{cases} 1, & \text{si } i = j, \\ 0, & \text{si } i \neq j. \end{cases}$$

Por lo tanto, el sistema de ecuaciones que se obtiene al imponer las condiciones de interpolación $p(x_i) = y_i$ es un sistema trivial

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix}$$

Esto quiere decir que la solución del sistema es $b_i = y_i$ por lo que el polinomio interpolante es $p_2(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x)$ e implica una mejora al condicionamiento del sistema.

Considerando que el objetivo del método de Müller es hallar las raíces de la función a través de aproximaciones con parábolas a la función, el polinomio interpolante será de un polinomio de segundo grado en todas las iteraciones. Sabiendo esto, se puede tomar la ventaja de utilizar la fórmula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ para hallar raíces de ecuaciones cuadráticas y conociendo de qué manera funciona la interpolación usando bases de Lagrange, se pueden realizar manipulaciones algebraicas para calcular el valor de los coeficientes de una manera más directa.

Al calcular los elementos que conforman la base de Lagrange, se puede notar que el denominador de la productoria resultará en una constante a la que llamaremos d_i

$$l_0(x) = \prod_{j=0, j \neq 0}^2 \frac{x - x_j}{x_0 - x_j} = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - x_1)(x - x_2)}{d_0}$$

$$l_1(x) = \prod_{j=0, j \neq 1}^2 \frac{x - x_j}{x_1 - x_j} = \frac{(x - x_0)(x - x_2)}{d_1}$$

$$l_2(x) = \prod_{j=0, j \neq 2}^2 \frac{x - x_j}{x_2 - x_j} = \frac{(x - x_0)(x - x_1)}{d_2}$$

Reescribimos $p_2(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x)$ reemplazando l_i

$$p_2(x) = y_0 \frac{(x - x_1)(x - x_2)}{d_0} + y_1 \frac{(x - x_0)(x - x_2)}{d_1} + y_2 \frac{(x - x_0)(x - x_1)}{d_2}$$

expandiendo el producto de diferencias en cada numerador y tomando $m_i = \frac{y_i}{d_i}$

$$p_2(x) = m_0(x^2 - x(x_1 + x_2) + x_1 x_2) + m_1(x^2 - x(x_0 + x_2) + x_0 x_2) + m_2(x^2 - x(x_0 + x_1) + x_0 x_1)$$

de esta forma se pueden extraer los coeficientes que corresponden a cada término del polinomio de segundo grado de la forma $ax^2 + bx + c$

Al desarrollar las cuentas tenemos que:

$$a = m_0 + m_1 + m_2$$

$$b = -m_0(x_1 + x_2) - m_1(x_0 + x_2) - m_2(x_0 + x_1)$$

$$c = m_0 x_1 x_2 + m_1 x_0 x_2 + m_2 x_0 x_1$$

Implementación y Resultados

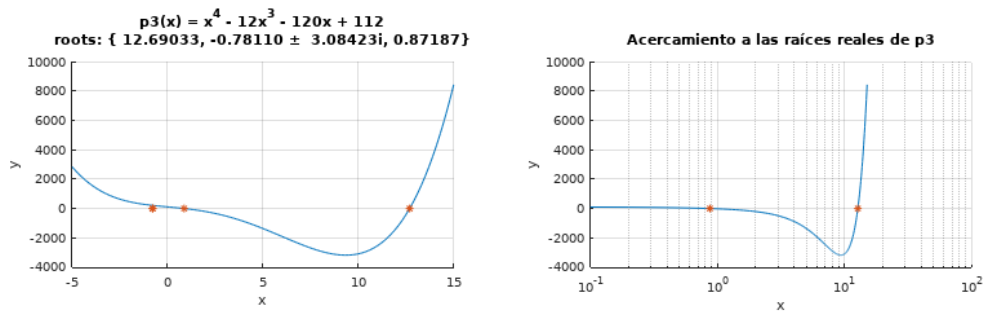
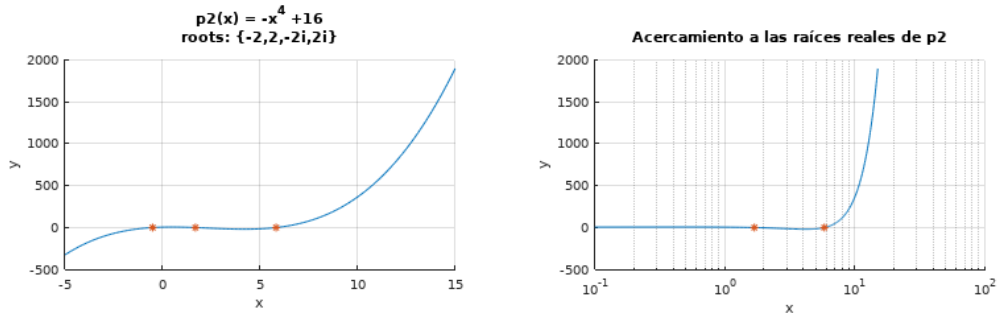
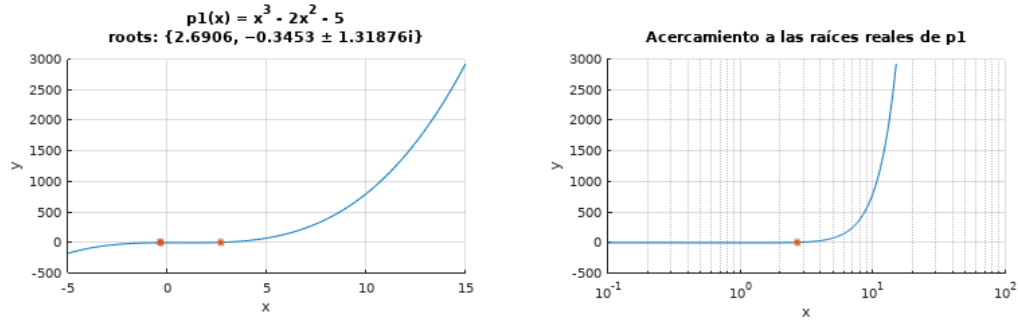
Código

CerosDePolinomios.m

Para la implementación en el método de Müller con el propósito de reducir la complejidad espacial, cada polinomio generado luego de realizar una depuración, se almacenará en él mismo evitando crear un nuevo arreglo de coeficientes por cada depuración. Otro aspecto considerado con el fin de hacer el algoritmo más eficiente es que la simplificación al método de Lagrange

para hallar el polinomio interpolante implica realizar todos los cálculos requeridos para hallar los coeficientes del polinomio de segundo grado de manera estática, por lo que se evita el uso de ciclos para realizar las productorias correspondientes al método así como también el hecho de trabajar luego con un sistema de ecuaciones trivial que mejora el condicionamiento del sistema.

A continuación, los resultados dados por el programa junto a la gráfica de los polinomios escogidos para la prueba del método.



Para todas la pruebas realizadas se escogieron polinomios con raíces reales e imaginarias iniciando el algoritmo con iteradores reales para así mostrar la efectividad del método para hallar todas las raíces del polinomio sin la limitación que presentan otros métodos de escoger iteradores reales con el fin de hallar raíces reales.

Propuesta Adicional: Interpolación Inversa

Esta propuesta es similar a la expuesta previamente con la diferencia de que se construirá un polinomio q_n que interpole a la inversa de la función f . Para esto se tomarán como nodos las alturas de los puntos dado de la función f .

Se plantea este camino como una solución para aproximar raíces ya que el polinomio q_n evaluado en una altura de la función resulta en un punto en el dominio de f , por tanto, sólo bastará con evaluar el polinomio cuando $y = 0$ dando como resultado una aproximación a una raíz de la función. Esto simplifica mucho más el cálculo para la obtención de la raíz ya que al evaluar $q_n(y) = ay^2 + by + c$ cuando $y = 0$, los primeros dos términos se multiplican por cero quedando así c como la raíz aproximada.

La razón por la que se expone esto como una alternativa y no como propuesta principal es por el inconveniente que se presenta para hallar raíces imaginarias con iteradores iniciales reales. Para explicar la razón procederemos a dar una noción visual de cómo se puede interpretar gráficamente la existencia de las raíces imaginarias en el plano cartesiano.

Se sabe que el discriminante $= b^2 - 4ac$ de una ecuación cuadrática nos indica el número de raíces que posee. Si < 0 quiere decir que la parábola que representa dicha ecuación no cortará en ningún punto del dominio con el eje x , por esta razón y sumado el hecho de que en este método se prescinde de usar la fórmula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ para hallar raíces, sino que directamente tomamos el valor del término que representa c , no basta con que determinemos donde la función f corta con el eje x .

Importante: Para la interpolación inversa se debe cumplir que todas las alturas y_i dadas deben ser diferentes y garantizar que

$$f^{-1}(y_i) = x_i$$

Si ocurre que dado un y_s , si $f^{-1}(y_s)$ posee dos valores diferentes, entonces f^{-1} no es una función.

Referencias

- [1] Ward Cheney y David Kincaid. *Numerical Mathematics and Computing, Sixth edition*. The University of Texas at Austin.
- [2] Michael T. Heath. *Scientific Computing: An Introductory Survey*. University of Illinois at Urbana-Champaign, 1997.
- [3] Biswa N. Datta, Luis M. Hernández-Ramos y Marcos Raydan. *Análisis Numérico. Teoría y Práctica*. [2018]