Eficiencia de algoritmos de búsqueda de raíces para polinomios.

Fabián Astorga Cerdas
Email: astorgafabian6@gmail.com

Andrés Brais Cháves
Email: abcbrais@gmail.com

Sergio Vargas Delgado
Email: savd_08@outlook.com

***Abstract*—En este documento se analizan métodos de búqueda de raíces en polinomios, para encontrar todas las raíces de un polinomio dado. Se estudian los métodos de Múller y Laguerre que encuentran una sola raíz ya sea compleja o real, además de la técnica de deflación polinomial que ayuda a encuentrar todas las raíces en un polinomio de manera más eficiente. Se estudia también la variación en la precisión de las raíces y cómo se ve afectada cada una de ellas al ser pulida.**

***Index Terms*—Deflación polinomial, Pulido de raíces, Raíces de polinomios.**

I. INTRODUCCIÓN

Encontrar raíces de polinomios es un problema recurrente en ingeniería y otras ciencias aplicadas, que además resulta de gran importancia en el área de análisis numérico. No obstante, encontrar todas las ráices de un polinomio es un proceso propenso a errores debido a que el polinomio propiamente puede estar muy mal condicionado, lo que llevaría al algoritmo de búsqueda a realizar muchas más operaciones de las necesarias. En este documento se analizan los métodos de Müller y Laguerre para búsqueda de raíces y las técnicas de deflación polinomial y el pulido de raíces como medios de apoyo para lograr encontrar todas las raíces de un polinomio con la mayor precisión posible. Este análisis tiene como finalidad comparar el desempeño de los métodos en términos de tiempo de ejecución y cantidad de instrucciones en lenguaje de máquina, así como el error de las raíces cuando se varía la aplicación de la técnica de pulido o la precisión.

II. ANTECEDENTES

De acuerdo con el teorema fundamental del álgebra, un polinomio de *n-ésimo* orden tiene exactamente *n* raíces. Si todos los coeficientes del mismo son reales, las raíces complejas (si existen) aparecen en pares conjugados, de lo contrario las raíces complejas no presentan ningún tipo de relación entre ellas [1], [2]. Conociendo esto, se pueden emplear algoritmos de búsqueda de ráices para encontrar cada una de ellas, aunque es importante mencionar que los métodos de búsqueda cerrados son poco prácticos debido a que se les debe dar un intervalo de búsqueda. Es por esto que los algorimos más comúnmente empleados son los que corresponden a métodos abiertos, en especial el método de Newton [1], [2], [3], [4]. Otros algoritmos más recientes involucran la utilización de *eigenvectors* y *eigenvalues* [1], [5] o hasta computación cuántica [6].

III. PROPUESTA

Para este documento se procura utilizar los métodos de Müller y de Laguerre como una alternativa a los algoritmos presentados en la sección anterior, siguiendo la sugerencia que se plantea en [1]. A dichos métodos se les suman las técnicas de deflación polinomial y de pulido de raíces, también sugeridos en la misma fuente.

A. Método de Müller

El método de Mülller es una variante del método de la secante. En el método de la secante se usan dos puntos, en el cual se dibuja una recta y se toma el punto del eje "*x*" donde "*y*" es cero. En el caso de Müller es similar pero en este caso se usan 3 puntos iniciales por lo cuales se dirige una parábola. En este caso se toma dos "*x*" donde sus respectivas "*y*" son cero.

Otra ventaja de este método es que también puede determinar las raíces de complejas de un polinomio. Cómo se obtienen estas raíces no es tan sencillo verlo geométricamente. Es más evidente en el análisis de la función.

En el método de Müller se utiliza el siguiente polinomio

$$P(x) = a\,(x - x_2)^{\,2} + b\,(x - x_2) + c \tag{1}$$

Este polinomio describe la parabola que se va a utilizar para encontrar las dos intersecciones con el eje *x*. Se utilizan 3 ecuaciones para determiar a, b y c. Estas cambian en la variable del polinomio anterior. Las ecuaciones son las siguientes

$$f\,(x_0) = a\,(x_0 - x_2)^{\,2} + b\,(x_0 - x_2) + c \tag{2}$$

$$f\,(x_1) = a\,(x_1 - x_2)^{\,2} + b\,(x_1 - x_2) + c \tag{3}$$

$$f\,(x_2) = a\,(0)^{\,2} + b\,(0) + c \tag{4}$$

Con las (2), (3) y (4) se pueden obtener los valores de a, b y c. Estos son calculados con las siguientes ecuaciones

$$a = \frac{(x_1 - x_2)[f(x_0) - f(x_2)] - (x_0 - x_2)[f(x_1) - f(x_2)]}{(x_0 - x_2)(x_0 - x_1)(x_1 - x_2)} \tag{5}$$

$$b = \frac{(x_0 - x_2)^2 [f(x_1) - f(x_2)] - (x_1 - x_2)^2 [f(x_0) - f(x_2)]}{(x_0 - x_2)(x_0 - x_1)(x_1 - x_2)} \tag{6}$$

$$c = f(x_2) \tag{7}$$

Con estas ecuaciones se determina los coeficientes de la parabola usada para determinar el tercer punto para determiar un cuarto punto en x donde y sea cero y además este más cerca de del $x_2$que es por así decirlo el centro de la búsqueda. Esta x se determina con (8):

$$x_3 = x_2 - \frac{2c}{b \pm \sqrt{b^2 - 4ac}} \tag{8}$$

Con esta ecuación se calcula un cuarto término, se elige la que este más cerca de $x_2$lo cual se puede determinar usando el signo de b. Como se puede determinar de la ecución, este nuevo término puede dar imaginario por lo tanto este análisis también calcula las raíces complejas por la geometría de las soluciones. El método continúa cambiando x_0 por x_1, x_1 por x_2 y x_2 por x_3. Con estos cambios se vuelve a hacer el cálculo hasta que la precisión sea igual o menor a la establecida.

B. Método de Laguerre

El método de Laguerre es capaz de encontrar raíces reales, complejas, simples y múltiples, por lo que es una opción muy versátil cuando se trata de encontrar todas las raíces de un polinomio. Si un polinomio posee solo raíces reales, el método garantiza converger a alguna de ellas dado cualquier punto de inicio, mientras que si existen raíces complejas no se puede garantizar convergencia, aunque si lo hay para la mayoría de los casos [1]. Dado un polinomio $P(x) = 0$ de grado *n*, para aproximar una raíz *r*, se emplean (9), (10), (11) y (12) iterativamente hasta obtener un valor de *r* satisfactorio, usualmente comparándolo con el de la iteración anterior [1].

$$G = \frac{-P'(r)}{P(r)} \tag{9}$$

$$H = G^2 - \frac{P''(r)}{P(r)} \tag{10}$$

$$C = \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}} \tag{11}$$

$$r = r + C \tag{12}$$

C. Deflación Polinomial

La deflación polinomial no es más que un caso especial de división polinomial donde es el dividendo, $(x - r)$, un monomio que representa una raíz, el divisor (por esta misma razón el residuo debe ser cero), y $Q(x)$ el cociente. El objetivo de realizar deflación polinomial es reducir el esfuerzo computacional de calcular todas las raíces de un polinomio: al encontrar una raíz utilizando cualquier método, se divide el polinomio original por esa raíz, lo que da como resultado un polinomio de menor grado al que se le busca una raíz y se divide por la misma. El proceso se repite hasta haber calculado todas las raíces. Una ventaja clara de hacer esto es que con cada iteración se disminuye el grado del polinomio, por lo que la búsqueda de raíces se simplifica, además de que al deflacionar se quita la posibilidad de que el algoritmo de búsqueda llegue a la misma raíz varias veces. Sin embargo, la gran desventaja de la deflación polinomial yace en que realizar división polinomial es una fuente de error, por lo que las raíces obtenidas utilizando este proceso también estarán impactadas por dicho error [1], [7].

D. Pulido de Raíces

El proceso de pulido de raíces es básicamente la búsqueda de ráices, utilizando cualquier método, a partir de una raíz con una componente de error; es decir, un valor muy cercano a una raíz. Se pulen raíces cuando un método de búsqeda haya alcanzado su límite de iteraciones (lo que quiere decir que no se obtuvo un valor satisfactorio para la raíz) o cuando se emplea deflación polinomial, pues las raíces que se obtienen acarrean el error de cada iteración de la deflación [1].

E. Algoritmo

Para encontrar todas las raíces de un polinomio, se deben integrar los métodos de búsqueda y las técnicas de las secciones anteriores de manera que se obtenga una raíz inicial a través del método que el usuario seleccione, con la que luego se procede a deflacionar el polinomio original tantas veces como sea necesario, para finalmente pulir dichas raíces (si el usuario así lo desea) y poder mostrar todas las raíces del polinomio.

IV. Resultados

De acuerdo con mediciones realizadas con la herramienta Callgrind, del paquete Valgrind, la cantidad de instrucciones en lenguaje de máquina al utilizar el método de Müller representa el 99.80% de las instrucciones totales del programa, mientras que este porcentaje es del 0,04% para el método de Laguerre. Es evidente el esfuerzo computacional adicional que requiere el método de Müller, aunque se esperaba que la diferencia no fuera tan marcada.

Para realizar una comparación entre los métodos de Laguerre y Muller, se utilizaron tres polinomios de prueba (véase la tabla 1).

Para el polinomio $x^4 - x^3 + 2x^2 - 4x - 8 = 0$, se puede observar que, por medio del método de Muller, se encontraron las raíces sin realizar el proceso de pulimiento y se obtuvieron valores exactos para dos de las cuatro las raíces obtenidas, así

Tabla 1 Comparación de métodos y con pulir y sin pulir raíces

Método		sin pulir	con pulir	sin pulir	con pulir	sin pulir	con pulir	sin pulir	con pulir
Raíz		-1		2		-2i		2i	
Muller	Resultado	-1	-1	2 -1.0996×10^{-10}	2	-1.99999i	-2i	2i	2i
	Error	0	0	5.04×10^{-9}	0	5.0×10^{-6}	0	0	0
Laguerre	Resultado	-1.000000319	-1.00000639	2.00000018	2.000000036	3.15×10^{-13}-2.000089i	2.000000178i	2.00000089i	2.00000178i
	Error	3.19×10^{-8}%	6.39×10^{-8}%	9.0×10^{-7}%	1.8×10^{-6}%	4.5×10^{-7}%	8.9×10^{-8}%	4.5×10^{-7}%	8.9×10^{-7}%

como dos valores muy aproximados para las raíces restantes. Si se utiliza el proceso de pulimiento, este promueve un aumento en la precisión de las raíces resultantes, como se puede apreciar. Obteniendo así valores exactos como resultado.

Al utilizar el método de Laguerre para la obtención de las raíces del polinomio anterior, sin utilizar el proceso de pulimiento, se aprecia que se encontraron las raíces del polinomio, de forma similar al comportamiento del método de Muller. Sin embargo, se observa que para ninguna raíz se obtuvo un valor exacto, ya que todos se acercan de manera aproximada al valor catalogado como correcto. Si se aplica el proceso de pulimiento, las raíces seguirán siendo muy aproximadas al valor real, pero afectan la precisión de los cálculos ya que se aumenta el porcentaje de error de las mismas, así que no siempre el pulimiento significa mayor precisión.

V. CONCLUSIONES

Si bien en algunos casos las raíces encontradas con el método de Müller son más precisas, la cantidad de esfuerzo computacional no aparenta ser proporcional al beneficio obtenido. El pulido de raíces ayuda a incrementar la precisión de las raíces encontradas cuando se trata del método de Müller. Sin embargo, para el método de Laguerre parece ser detrimental, incrementando el error casi en un factor de dos. Se comprobó que un polinomio de grado uno se considera como una excepción para el método de Laguerre, ya que este utiliza cálculo diferencial y diversas evaluaciones del polinomio en ciertos puntos a evaluar. Además de que las variables que componen el método contienen algunas divisiones, por lo que se indefine a la hora de evaluar con valores muy pequeños que tienden a cero, y por ende no se puede encontrar un valor aproximado para la raíz.

REFERENCES

[1] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, Numerical Recipes 3rd Edition: The Art of Scientific Computing, 1st ed. New York: Cambridge University Press, 2007.

[2] Y. Jia, Roots of Polynomials. Ames: Department of Computer Science, Iowa State University, 2016.

[3] P. Du, L. Cheng, H. Jiang and F. Wang, "Some Accurate Methods for Finding Simple Roots of Polynomials in Floating Point Arithmetic", 2012 Fourth International Conference on Computational and Information Sciences, 2012.

[4] D. Horman, T. Marusenkova and I. Yurchak, "Algorithm of polynomials' root separation for ARM-based microcontrollers", The Experience of Designing and Application of CAD Systems in Microelectronics, 2015.

[5] A. Ivanov, H. Aliyev and B. Bakiyev, "Analytical method of finding polynomial roots by using the eigenvectors, eigenvalues apparatus", 2015 Twelve International Conference on Electronics Computer and Computation (ICECCO), 2015.

[6] S. Guodong, S. Shenghui and X. Maozhi, "Quantum Algorithm for Polynomial Root Finding Problem", 2014 Tenth International Conference on Computational Intelligence and Security, 2014.

[7] B. Dumitrescu and I. Tabus, "How to deflate polynomials in LSP computation", 1999 IEEE Workshop on Speech Coding Proceedings, 1999.