Trabajo Final
Análisis Numérico:
Método Quotient-Difference

Integrantes:

- Sandoval, Agustín Pedro
- Sanna, María Paz
- Sauro, Maia
- Savarino, Jeremías

Fecha de entrega: 01/06/2021

Profesor: Francisco Lizarralde

Introducción

En el siguiente trabajo se analiza y desarrolla la implementación del algoritmo Quotient- Difference, el cual permite obtener todas las raíces de un polinomio de coeficientes reales y completos de grado n de manera simultánea. En caso de que el polinomio contenga raíces complejas o reales co-modulares, se procede a aplicar el método de Bairstow, también desarrollado, el cual mejora el factor cuadrático según una precisión deseada.
En ciertos casos, será necesario el uso de métodos de preprocesamiento de polinomios, para que estos cumplan los requisitos de uso del algoritmo Q-D.
Finalmente, a través de ejemplos, se realiza una comparación de los resultados con otros métodos de obtención de raíces.
Otras aplicaciones posibles de este método son el cálculo de autovalores y autovectores de una matriz, fracciones continuas, transformación LR de Rutishauser (**factorización** que resume el proceso de eliminación gaussiana aplicado a la **matriz** de una matriz).

Algoritmo Q-D

Este método sirve para aproximar los factores lineales y cuadráticos en los cuales se pueda expresar un polinomio de grado n. Dicho polinomio tiene como condición tener todos sus coeficientes reales y estar completo, es decir **con todos sus coeficientes no nulos.**
Sea el polinomio real:

$$p(x) = a_n\ x^n + a_{n-1}\ x^{n-1} + \ldots\ldots + a_1\ x + a_0$$

tal que $a_k \neq 0$ para todo k = n, n-1, ...,1,0

A continuación se arma una tabla de valores a partir de dos conjuntos de fórmulas que permiten calcular los valores de la iteración cero y los valores de las demás iteraciones.

Cálculo de los valores de qi y ei para:

Iteración 0:

$$q_1^{(0)} = \frac{-a_{n-1}}{a_n} \quad ; \quad | \quad q_i^{(0)} = 0 \quad \forall i = 2,3,\cdots,n$$

$$e_i^{(0)} = \frac{a_{n-i-1}}{a_{n-i}} \quad \forall i = 1,2,\cdots,n-2,n-1 \quad ; \quad e_0^{(0)} = e_n^{(0)} = 0$$

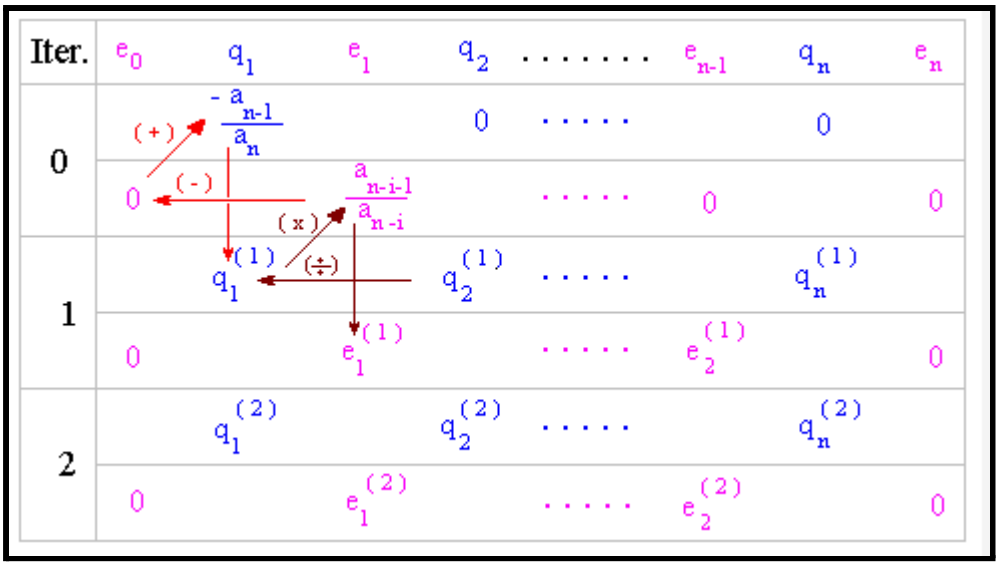
Iteraciones de orden (m+1): (resto de las iteraciones):

$$q_i^{(m+1)} = e_i^{(m)} - e_{i-1}^{(m)} + q_i^{(m)} \qquad \forall i = 1,2,\cdots,n$$

$$e_i^{(m+1)} = \frac{q_{i-1}^{(m+1)}}{q_i^{(m+1)}} \cdot e_i^{(m)} \qquad \forall i = 1,2,\cdots,n-1$$

$$e_0^{(m+1)} = e_n^{(m+1)} = 0$$

Se genera la siguiente tabla:



En las columnas q_i los valores de las raíces irán convergiendo a un valor real mientras que en las columnas e_i el valor del error tenderá a cero. En caso contrario si los valores de alguna columnae_i fluctúan quiere decir que estamos en presencia de raíces complejas y debemos recurrir al método de Bairstow.

Método de Bairstow

Si el polinomio posee raíces complejas, se observa que los valores de las columnas de error no se aproximan a cero sino que presentan valores oscilantes. El algoritmo de Bairstow permite refinar los factores cuadráticos de un polinomio, es decir aquellos que contienen raíces co-modulares , ya sean complejas o reales.

Para ello partimos de los siguientes datos :

a) p(x) = a0 X_n + a1 X_{n-1} + a2 X_{n-2} + ...+ an

b) un cierto **factor cuadrático** aproximado X^2 - u X - v

c) una cota para el error ε.

Para aproximar el factor cuadrático que contiene las **raíces co-modulares** (igual módulo) se procede de la siguiente manera:

Sea el factor cuadrático: X^2 - u X - v

Sea la iteración m-ésima de Q-D donde la columna ei fluctúa:

q_i	e_i	q_{i+1}
$q_i^{(m-1)}$	$e_i^{(m-1)}$	$q_{i+1}^{(m-1)}$
$q_i^{(m)}$		$q_{i+1}^{(m)}$

$$u = q_i^{(m)} + q_{i+1}^{(m)}$$
$$v = q_i^{(m-1)} * q_{i+1}^{(m)}$$

Si se tiene un gran número de iteraciones donde el proceso ya está estabilizado entonces puedo aproximar los valores de **u** y **v**.
Finalmente será conveniente aplicar Bairstow para mejorar el factor cuadrático según una precisión deseada

Explicación del Algoritmo de Bairstow

Partiendo de $u_0 = \frac{-a_{n-1}}{a_n}$ y $v_0 = \frac{-a_{n-1}}{a_n}$ se realiza un refinamiento del factor cuadrático a partir de obtener una sucesión de valores de ui y vi cada vez más aproximados tal que se consiga la exactitud deseada para poder hallar las raíces que conforman dicho factor.
Se calcula, en cada iteración, los valores de **qt** y **pt** , para t = 0,1,2, ... ,n por medio de las siguientes expresiones:

$$q_t = a_t + u_m \cdot q_{t-1} + v_m \cdot q_{t-2} \quad para\ 0 < t < n \quad y\ adoptando: \quad q_{-2} = q_{-1} = 0$$
$$p_t = q_t + u_m \cdot p_{t-1} + v_m \cdot p_{t-2} \quad para\ 0 < t < n\text{-}1 \quad y\ adoptando: \quad p_{-2} = p_{-1} = 0$$

Con los valores de **qt** y **pt** calculados, podemos obtener los valores de **h** y **k** :

$$p_{n-2} \cdot h + p_{n-3} \cdot k = q_{n-1}$$
$$p_{n-1} \cdot h + p_{n-2} \cdot k = q_n$$

o sea, despejando $\boldsymbol{h}$ *y* $\boldsymbol{k}$

$$h = \frac{q_n \cdot p_{n-3} - q_{n-1} \cdot p_{n-2}}{p_{n-2}^2 - p_{n-1} \cdot p_{n-3}}$$

$$k = \frac{q_{n-1} \cdot p_{n-1} - q_n \cdot p_{n-2}}{p_{n-2}^2 - p_{n-1} \cdot p_{n-3}}$$

Y con **h** y **k** obtenemos nuevos valores para u y v:

$$u_{m+1} = u_m + h_m$$
$$v_{m+1} = v_m + k_m$$

Este proceso se detiene cuando, en alguna *m-ésima* iteración obtenemos valores de **qn** y **qn-1** menores que la *cota de error deseada*, en caso contrario, continuamos con el proceso de calcular nuevos valores de q y p.

Preprocesamiento de un polinomio

Se entiende por preprocesamiento a un conjunto de transformaciones cuyo objetivo es adecuar el problema para poder resolverlo mediante el algoritmo Q-D, que contiene ciertas restricciones a la hora de utilizarlo.
Los métodos de transformación son:

1. Traslación efectuada sobre la indeterminada

Este método es útil para desarrollar el algoritmo Q-D cuando uno de los coeficientes es nulo debido a que consiste en realizar un cambio de base de la siguiente forma:

$$P(x) \rightarrow P(x + c)$$

$$Q(x) = P(x + c) = \sum b_k (x - c)^k \qquad \text{siendo } k=0...n$$

Siendo los coeficientes de Q(x) dados por:

$$b_k = \frac{P^{(k)}(c)}{k!}$$

Aplicando traslación se logra que si c < 0 la función se desplace hacia la derecha y si c > 0 se mueva hacia la izquierda. Por lo tanto z es cero de P(x) si y sólo si z – c es cero de Q(x)

$$P(z) = 0 \leftrightarrow Q(z-c) = 0$$

La elección de la constante c, al momento de implementar el cambio de base debe tener en cuenta que ningún coeficiente del polinomio sea anulado .

2. Transformación Recíproca

El método consiste en realizar la siguiente operación:

$$P(x) \rightarrow x^n p(1/x) \text{ donde n es el grado del polinomio P(x)}$$

Por lo tanto el nuevo polinomio Q (x) será:

$$Q(x) = x^n P(1/x) = x^n \sum a_k (1/x) = \sum a_{n-k} x^k = \sum a_k x^{n-k}$$

Un número complejo z distinto de 0 será raíz del polinomio P(x) si, y sólo si, su recíproco 1/z es cero del polinomio transformado Q(x).

$$P(z) = 0 \leftrightarrow Q(1/z) = 0$$

3. Homotecia sobre un polinomio

Es un método que se implementa para realizar un ajuste de escalas en el gráfico y consiste en realizar el producto de todos los coeficientes de un polinomio por un factor, ocasionando que se altere el factor de escala y dejando invariantes a las raíces.

$$P(x) \rightarrow c\,P(x) = Q(x) \quad (c \neq 0)$$

El resultado obtenido es un nuevo polinomio que posee todos los coeficientes de P(x) pero ahora multiplicados por el valor c.

4. Homotecia sobre la indeterminada

Esta técnica consiste en una combinación de dos procesos:

- Se modifica el eje x en base a un factor |c|.
- El sentido del eje x se invierte si el valor c es negativo.

$$P(z) = 0 \leftrightarrow Q(z/c) = 0 \text{ o}$$

$$P(cz) = 0 \leftrightarrow Q(z) = 0$$

En caso de que el factor c=-1 dan los siguientes sucesos:

1) Los coeficientes del polinomio Q(x) = P(-x) se obtienen como:

$b_k = [(-1)]^k a_k$ siendo a_k el coeficiente de x_k en P(x).

2) Los ceros del polinomio transformado son los opuestos del polinomio original P(x).

$$P(-z) = 0 \leftrightarrow Q(z) = 0$$

Casos Analizados

Caso 1: Todas raíces reales

Se tiene el polinomio $P(x)= 25x^3+75x^2-15x-1$ cuyas raíces son:

- x1=3.184469958
- x2= 0.2373840453
- x3= -0.05291408719

Se prosigue a aplicar el algoritmo Q-D y obtener la siguiente tabla de valores:

Iteración	e0	q1	e1	q2	e2	q3	e3
0		-3.000000		0.000000		0.000000	
	0.000000		-0.200000		0.066667		0.000000
1		-3.200000		0.266667		-0.066667	
	0.000000		0.016667		-0.016667		0.000000
2		-3.183333		0.233333		-0.050000	
	0.000000		-0.001222		0.003571		0.000000
3		-3.184555		0.238126		-0.053571	
	0.000000		0.000091		-0.000803		0.000000
4		-3.184464		0.237232		-0.052768	
	0.000000		-0.000007		0.000179		0.000000

Tabla 2. Tabla con los valores de q y e de las iteraciones para el caso 2

Interacciones máximas= 100
Tolerancia=0.0001

Raíces obtenidas por método de Q-D:

- x1= -3.18447
- x2= 0.23742
- x3= -0.05295

Las tres raíces se encuentran marcadas en el siguiente gráfico:

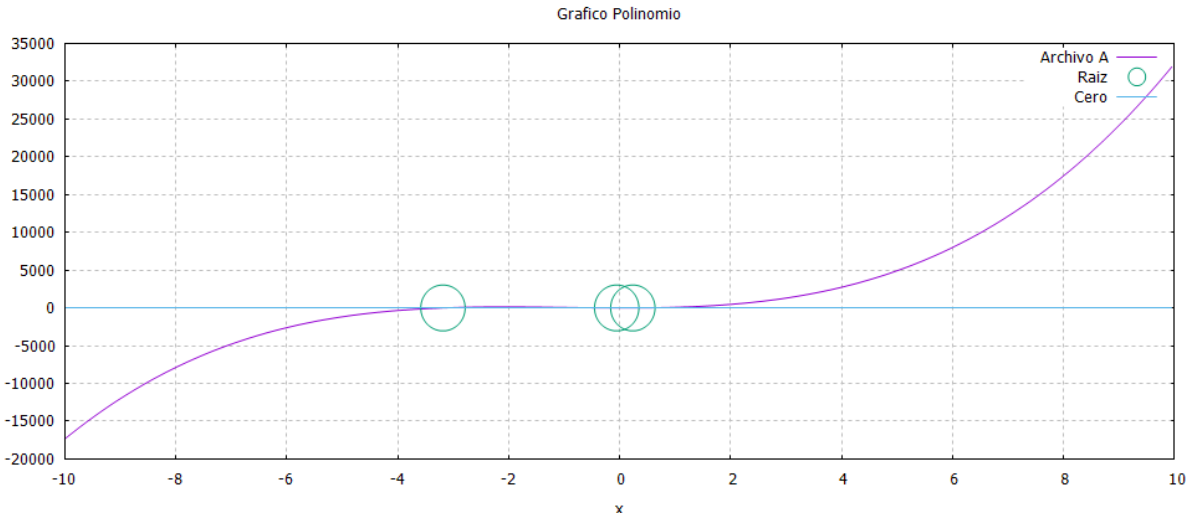


Gráfico 1. Gráfica del polinomio P(x) para el caso 1

Raíces obtenidas por método de Bisección con una tolerancia de 0.0001:

- x1= -3,18447
  Utilizando un intervalo [-3.15;-3.19] . Se obtuvo luego de 9 iteraciones.

- x2= -0.05291
  Utilizando un intervalo [-0.1;0.0]. Se obtuvo luego de 10 iteraciones

- x3= 0.2374023
  Utilizando un intervalo [0.2;0.24] . Se obtuvo luego de 9iteraciones.

Caso 2: Par de raíces co-modulares

Se tiene el polinomio $P(x)= x^3+x^2-9x-9$ cuyas raíces son:

- x1=-3
- x2 =-1
- x3= 3 .

Se prosigue a aplicar el algoritmo Q-D y obtener la siguiente tabla de valores:

89	-9.000000	9.000000	-1.000000	
	0.000000	8.000000	-0.000000	0.000000
90	-1.000000	1.000000	-1.000000	
	0.000000	-8.000000	0.000000	0.000000
91	-9.000000	9.000000	-1.000000	
	0.000000	8.000000	-0.000000	0.000000
92	-1.000000	1.000000	-1.000000	
	0.000000	-8.000000	0.000000	0.000000
93	-9.000000	9.000000	-1.000000	
	0.000000	8.000000	-0.000000	0.000000
94	-1.000000	1.000000	-1.000000	
	0.000000	-8.000000	0.000000	0.000000
95	-9.000000	9.000000	-1.000000	
	0.000000	8.000000	-0.000000	0.000000
96	-1.000000	1.000000	-1.000000	
	0.000000	-8.000000	0.000000	0.000000
97	-9.000000	9.000000	-1.000000	
	0.000000	8.000000	-0.000000	0.000000
98	-1.000000	1.000000	-1.000000	
	0.000000	-8.000000	0.000000	0.000000
99	-9.000000	9.000000	-1.000000	
	0.000000	8.000000	-0.000000	0.000000
100	-1.000000	1.000000	-1.000000	
	0.000000	-8.000000	0.000000	0.000000

Tabla 2. Tabla con los valores de q y e de las últimas iteraciones para el caso 1

Interacciones máximas= 100
Tolerancia=0.0001

Raíces obtenidas por el método Q-D

En este caso se observa que se obtuvieron dos raíces por el método de Bairstow, más allá de que estas sean reales .Esto se debe a que en el método iterativo de Q-D los valores de q1 y q2 serán muy similares generando que estas no convergen al valor deseado ni tampoco e1. Se obtienen exactamente -3 y 3. La restante raíz, como se puede ver en la tabla es -1.

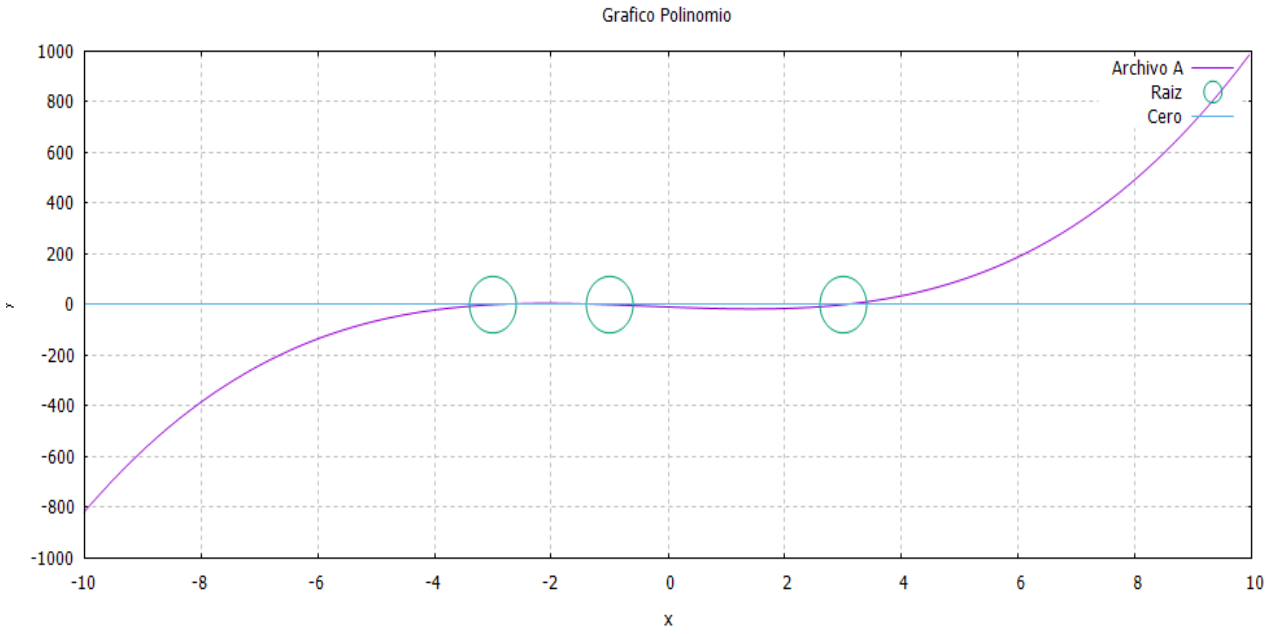


Gráfico 2. Gráfica del polinomio P(x) para el caso 2

Comparación con Punto Fijo Sistemático con una tolerancia de 0.0001:

- x1= 2.99998
  Utilizando un intervalo [2 ; 4]. Se obtuvo luego de 4 iteraciones.

- x2= -3.00004
  Utilizando un intervalo [-4;-2]. Se obtuvo luego de 32 iteraciones.

- x3= -1.00002
  Utilizando un intervalo [-2;0]. Se obtuvo luego de 6 iteraciones.

<u>Caso 3:</u> Raíces complejas y reales

Siendo el polinomio $P(x)= x^4 +2x^3 -2x^2 +7x +10$

-Sus raíces reales son

- x1= -1

- x2= -3.054239

-Sus raíces complejas son:

- x3= 1.027119592 + 1.489685589 i
- x4=1.027119592 - 1.489685589 i

Se prosigue a aplicar el algoritmo Q-D y obtener la siguiente tabla de valores:

90		-3.054239		2.690374		-0.636135		-1.000000		
	0.000000		-0.000000		-1.853117		0.000000		0.000000	
91		-3.054239		0.837257		1.216982		-1.000000		
	0.000000		0.000000		-2.693571		-0.000000		0.000000	
92		-3.054239		-1.856314		3.910553		-1.000000		
	0.000000		0.000000		5.674338		0.000000		0.000000	
93		-3.054239		3.818023		-1.763784		-1.000000		
	0.000000		-0.000000		-2.621332		0.000000		0.000000	
94		-3.054239		1.196691		0.857548		-1.000000		
	0.000000		0.000000		-1.878444		-0.000000		0.000000	
95		-3.054239		-0.681753		2.735992		-1.000000		
	0.000000		0.000000		7.538522		0.000000		0.000000	
96		-3.054239		6.856769		-4.802530		-1.000000		
	0.000000		-0.000000		-5.280034		0.000000		0.000000	
97		-3.054239		1.576735		0.477504		-1.000000		
	0.000000		0.000000		-1.599026		-0.000000		0.000000	
98		-3.054239		-0.022291		2.076531		-1.000000		
	0.000000		0.000000		148.955788		0.000000		0.000000	
99		-3.054239		148.933497		-146.879258		-1.000000		
	0.000000		-0.000000		-146.901242		0.000000		0.000000	
100		-3.054239		2.032255		0.021984		-1.000000		
	0.000000		0.000000		-1.589102		-0.000000		0.000000	

Tabla 3. Tabla con los valores de q y e de las últimas iteraciones para el caso 3

Interacciones máximas= 100

Tolerancia=0.0001

Raíces obtenidas por método de Q-D

Se puede observar una fluctuación en los valores de la columna e2, volviéndose inestable a medida que aumentan las iteraciones, dichos valores de error no tienden a cero, lo que indica que dos de las raíces del polinomio son complejas por lo que se procede a implementar el método de Bairstow para obtener el par de raíces complejas restantes:

- x1=1,02712 + 1,48969 i
- x2=1,02712 - 1,48969 i

y se obtienen 2 raíces reales:

- x3= -3,05424
- x4= -1,0000

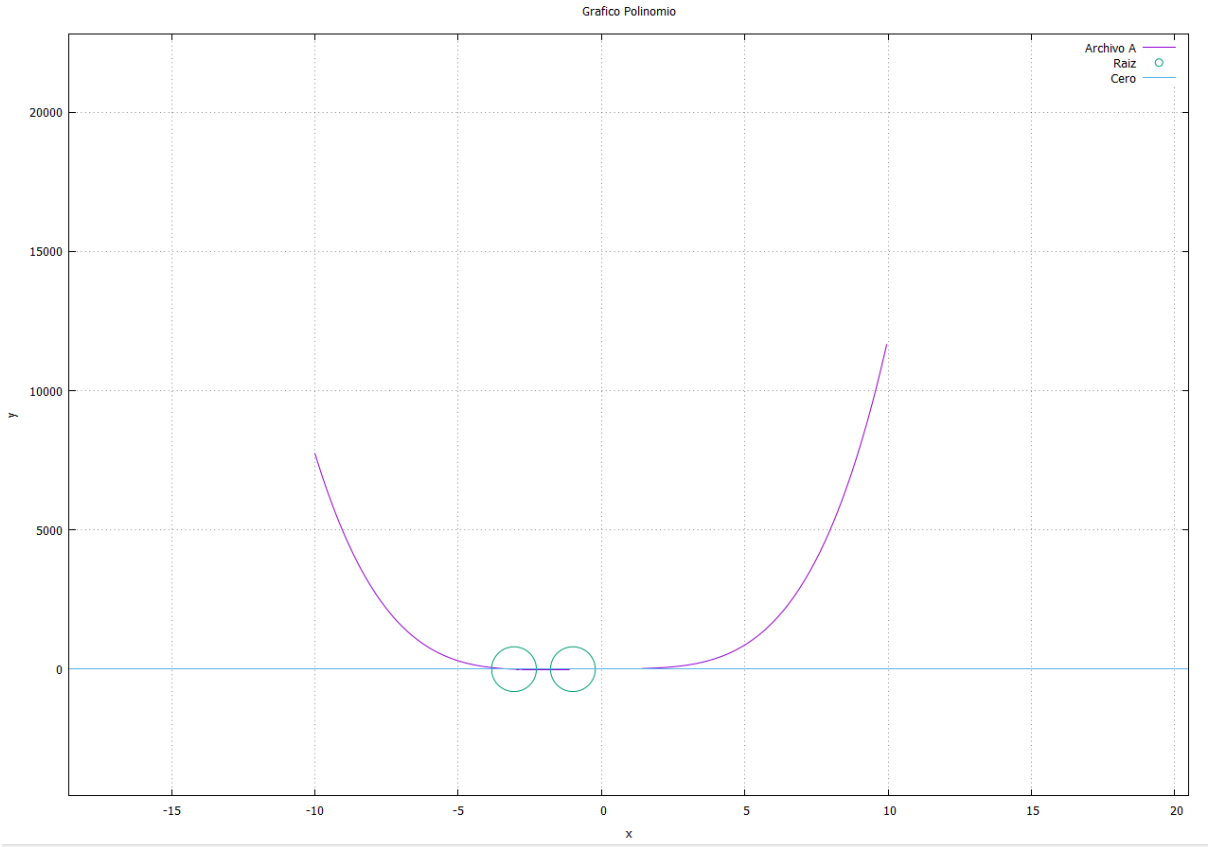


Gráfico 3. Gráfica del polinomio P(x) para el caso 3

Comparación con el método de Newton Raphson con una tolerancia de 0.0001 :

Implementando el método de Newton se obtuvieron las siguientes raíces reales:

- x1= -3.0542392
  Partiendo de x0= -5. Se obtuvo luego de 9 iteraciones.

- x2=-1.00000
  Partiendo de x0= -2. Se obtuvo luego de 4 iteraciones

Además, modificando el método de Newton para hallar raíces complejas se encontraron las raíces:

- x3=1.02712 + 1.489689 i
  Partiendo de x0= (0,2.4 i). Se obtuvo luego de 9 iteraciones.

- x4=1.02712 -1.48969 i
  Partiendo de x0= (1, -10 i). Se obtuvo luego de 12 iteraciones.

<u>Caso 4:</u> Coeficientes iguales a 0

Siendo el polinomio $P(x)= x^3 -2x^2 -1$ cuyas raíces son:

- x1= 2.20556943
- x2= -0.1027847152 + 0.6654569512 i
- x3= -0.1027847152 - 0.6654569512 i

Este polinomio no cumple con la condición necesaria de que todos los coeficientes del polinomio sean distintos de cero para poder aplicar el algoritmo de Q-D. Por lo que se procede a realizar una traslación efectuada sobre la indeterminada y así poder calcular sus raíces.

P(x) ------> P(x+c)

Utilizando c=2 se consigue el polinomio:

$Q(x)= x^3 + 4x^2 + 4x - 1$

Luego a partir de este nuevo polinomio se prosigue a aplicar el algoritmo Q-D y obtener la siguiente tabla de valores:

92	-4.652422	0.446852	0.205569		
	0.000000	1.492445	0.000000		0.000000
93	-3.159977	-1.045592	0.205569		
	0.000000	0.493829	-0.000000		0.000000
94	-2.666148	-1.539421	0.205569		
	0.000000	0.285135	0.000000		0.000000
95	-2.381013	-1.824556	0.205569		
	0.000000	0.218497	-0.000000		0.000000
96	-2.162516	-2.043053	0.205569		
	0.000000	0.206427	0.000000		0.000000
97	-1.956090	-2.249480	0.205569		
	0.000000	0.237388	-0.000000		0.000000
98	-1.718702	-2.486868	0.205569		
	0.000000	0.343488	0.000000		0.000000
99	-1.375214	-2.830355	0.205569		
	0.000000	0.706938	-0.000000		0.000000
100	-0.668276	-3.537293	0.205569		
	0.000000	3.741939	0.000000		0.000000

Tabla 4. Tabla con los valores de q y e de las últimas iteraciones para el caso 4

Interacciones máximas= 100

Tolerancia=0.0001

Raíces obtenidas por método de Q-D:

Mirando la tabla se aprecia que los valores de ei están fluctuando, esto nos quiere decir que las raíces del polinomio son complejas, por lo que debemos hallarlas mediante el método de Bairstow.
Se adquieren como raíces, recordando que se realizó una traslación con c=2:

Raíces de Q(x):

- x1 = -2.10278 + 0.66546 i
- x2 = -2.10278 - 0.66546 i
- x3 = 0.20557

Por lo tanto las raíces del polinomio original P(x) obtenidas por el algoritmo Q-D son:

- x1= 2.20557
- x2= -0.10278 + 0.66546 i
- x3= -0.10278-0.66546 i

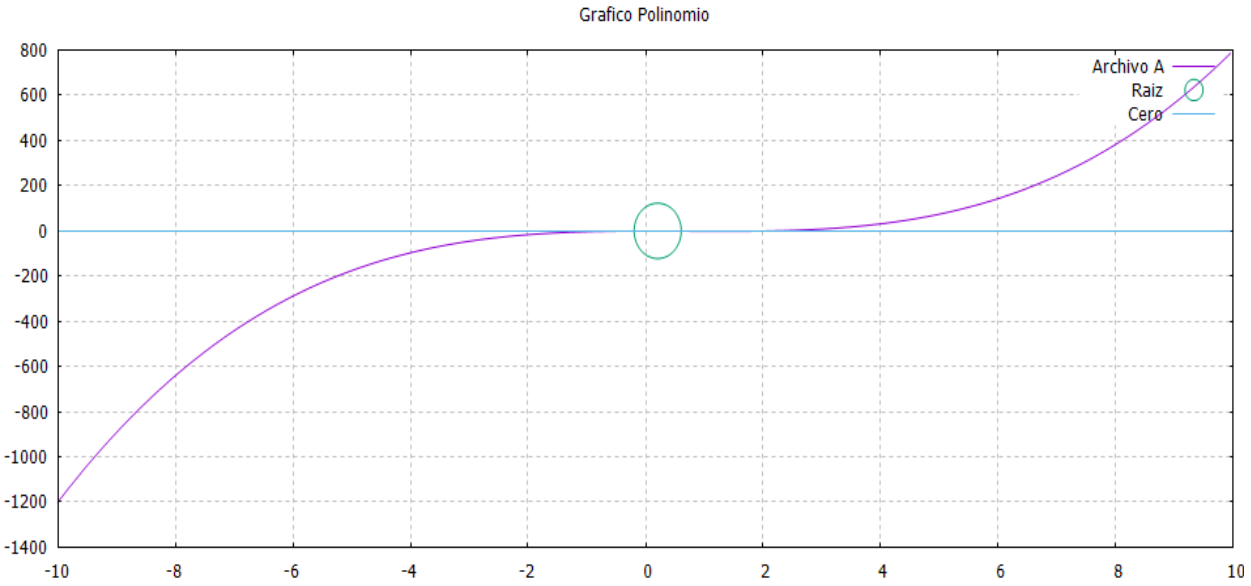


Gráfico 4. Gráfica del polinomio P(x) para el caso 4

Comparación con el método de Newton Raphson con una tolerancia de 0.0001:

A partir del método de Newton se obtuvo la siguiente raíz real:

- x1=2.20558
  Partiendo de x0= 1. Se obtuvo luego de 11 iteraciones.

Además se obtuvieron las siguientes raíces imaginarias:

- x2=-0.10278 + 0.66546i
  Partiendo de x0=(1,i). Se obtuvo luego de 9 iteraciones.

- x3= -0.10278 - 0.66546 i
  Partiendo de x0=(1,-i). Se obtuvo luego de 9 iteraciones.

<u>Caso 5:</u> Raíces muy cercanas

Se tiene el polinomio $P(x)=x^2 -8.001x +16.004$ cuyas raíces son:

- x1=4
- x2= 4,001.

Se prosigue a aplicar el algoritmo Q-D y obtener la siguiente tabla de valores:

91		4.044000		3.957000
	0.000000		-0.000467	0.000000
92		4.043533		3.957467
	0.000000		-0.000457	0.000000
93		4.043076		3.957925
	0.000000		-0.000448	0.000000
94		4.042628		3.958373
	0.000000		-0.000438	0.000000
95		4.042189		3.958811
	0.000000		-0.000429	0.000000
96		4.041760		3.959240
	0.000000		-0.000421	0.000000
97		4.041339		3.959661
	0.000000		-0.000412	0.000000
98		4.040927		3.960073
	0.000000		-0.000404	0.000000
99		4.040523		3.960477
	0.000000		-0.000396	0.000000
100		4.040127		3.960873
	0.000000		-0.000388	0.000000

Tabla 5. Tabla con los valores de q y e de las últimas iteraciones para el caso 5

Interacciones máximas= 100

Tolerancia=0.0001

Las raíces obtenidas mediante el algoritmo Q-D son:

- x1= 4.00197
- x2= 3.99903

A continuación se observan los gráficos obtenidos:

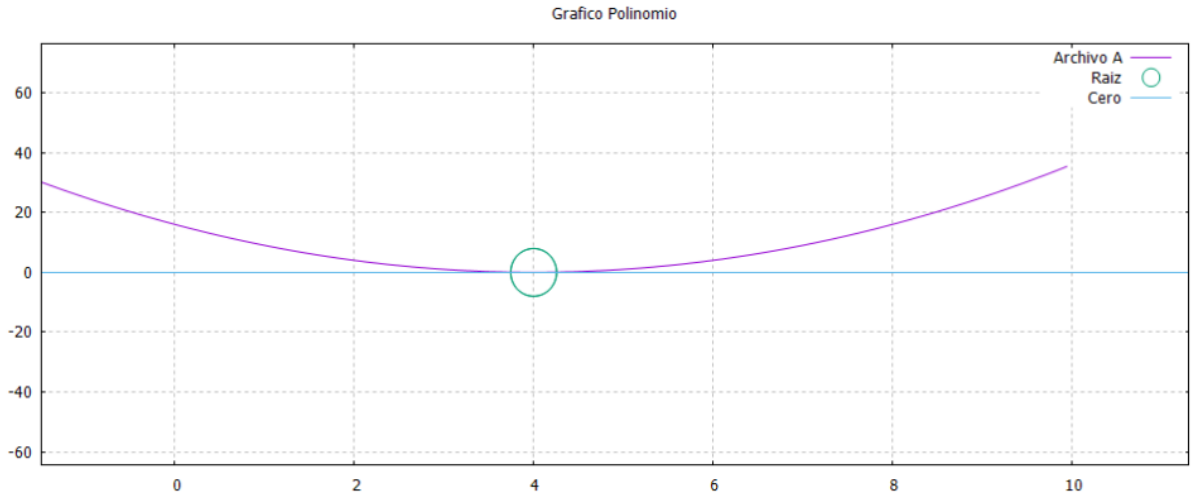


Gráfico 5. Gráfica del polinomio P(x) para el caso 5

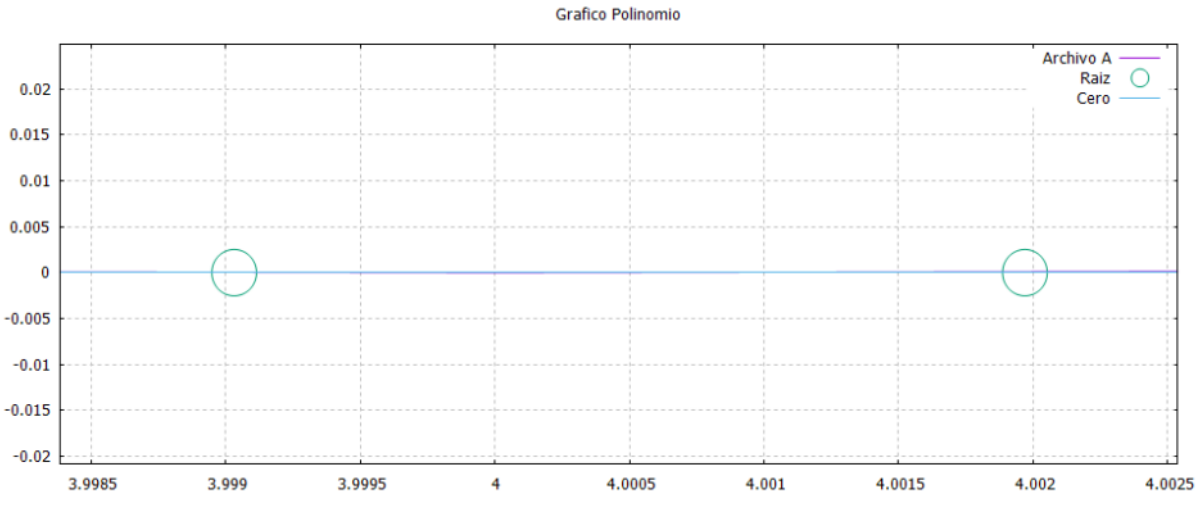


Gráfico 6. Aumento del gráfico del polinomio P(x) para el caso 5

Comparación con el método de Newton Raphson con una tolerancia de 0.0001:

A partir del método de Newton se obtuvo la siguiente raíz real:

- x1= 3,99259
  Partiendo de x0= 0, se obtuvo luego de 11 iteraciones.

- x2=4,00840
  Partiendo de x0= 5, se obtuvo luego de 7 iteraciones.

Conclusión

La obtención directa de todas las raíces del polinomio es una de las ventajas principales del algoritmo Q-D, a diferencia de lo que sucede en los otros métodos vistos. Por ejemplo, en el método de Newton Raphson se obtendrán de a una mediante un valor semilla o por medio de una deflación por cada raíz.

Se pudo observar mediante los ejemplos desarrollados en cada uno de los casos, una gran diferencia en la cantidad de iteraciones necesarias entre los métodos, siendo estas considerablemente mayores en el algoritmo Q-D. Por lo tanto, en el caso de un polinomio de bajo grado (1, 2, 3 o 4) no es conveniente el uso de Q-D para el cálculo de las raíces, ya que aplicando cualquier otro método visto en clase, se obtendrá un número total de iteraciones mucho menor al obtenido.

Como desventaja del método Q-D encontramos que el polinomio debe estar completo, es decir que todos sus coeficientes sean distintos de cero ya que, en caso contrario hay que aplicarle un algoritmo de preprocesamiento a dicho polinomio, lo que conlleva la implementación de cálculos extras. Sin embargo, como se puede observar en el código, es posible aplicar métodos de transformación al polinomio con el objetivo de que cumpla las condiciones requeridas y así poder aplicar el algoritmo.

Otro problema de este método es que si las raíces a encontrar no poseen una separación lo suficientemente grande, es posible que no encuentre todas. Para solucionarlo, también se podría aplicar un algoritmo de preprocesamiento, como el de Transformación Recíproca.

Cuando las raíces son complejas o co-modulares, se debe aplicar el método de Bairstow. Éste ajusta el factor cuadrático según una tolerancia determinada, de modo que sea posible hallar las raíces para las cuales Q-D falla. Este problema se puede observar en las columnas de error

correspondientes, cuando a medida que avanzan las iteraciones, estas en lugar de tender a cero toman valores fluctuantes.

Se concluye que el algoritmo, a pesar de poseer varias limitaciones, puede aplicarse de todas formas, teniendo en cuenta estas restricciones y, en caso de ser necesario, debe forzarse al polinomio a cumplirlas, aplicando los métodos de transformación mencionados.