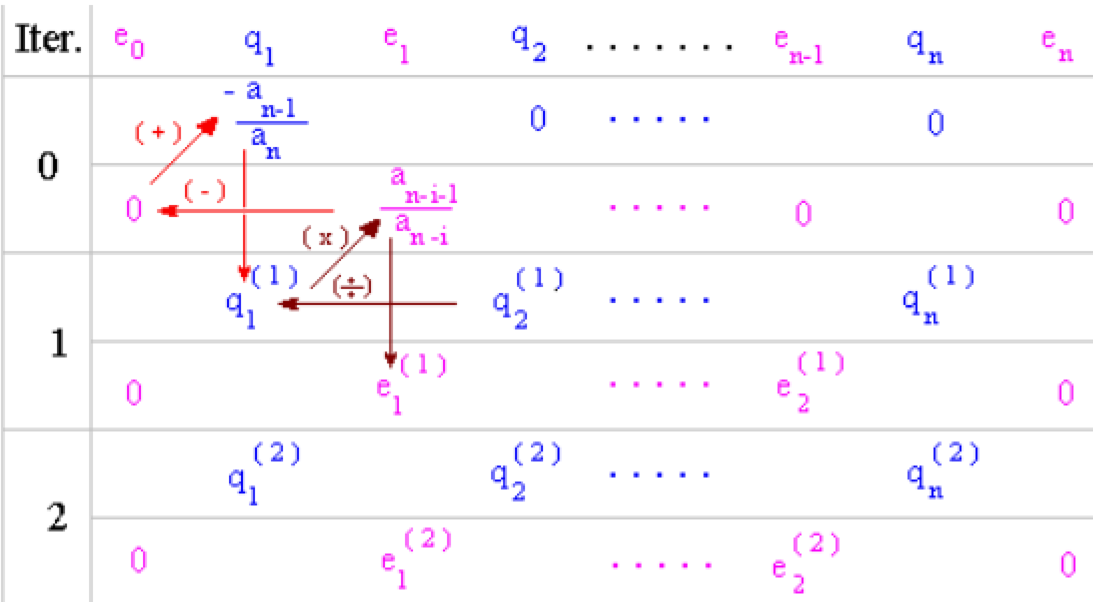
Trabajo Final Análisis Numérico:

Algoritmo Quotient-Difference (Diferencia de cocientes)



Profesor: Mg. Ing. Francisco A. Lizarralde

Grupo 8

Integrantes:
- Listorti, Aristide
- Iavicoli Dulcet Nicolas
- Loscalzo Bianchi, Bárbara

Objetivo

El objetivo de este trabajo es la implementación del algoritmo Q-D, probarlo en distintos casos posibles y compararlo con los métodos de obtención de raíces de polinomios vistos en la asignatura.

Introducción

El algoritmo Q-D es un método que permite obtener todas las raíces de un polinomio de grado n, aproximando los factores lineales y cuadráticos del mismo.
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 realizó una comparación de los resultados con otros métodos de obtención de raíces.

Algoritmo de diferencias de cocientes (QD):

Antes de describir el método se dará un breve vistazo a los métodos existentes para el cálculo de raíces polinómicas:

- Deflación Polinomial
- Método de Bairstow
- Método de Laguerre
- Método de Bernoulli
- **Método de diferencias de cocientes (algoritmo QD)**
- Método de Newton
- Método de Lehmer-Schur 79
- Método de raíz cuadrada de Graeffe

Presentación del algoritmo QD:

Dado un polinomio de grado n con coeficientes reales y completo, es decir con todos sus coeficientes diferentes de cero, se arma una tabla de valores, utilizando dos conjuntos de fórmulas: una para calcular los valores de la iteración cero y otro para calcular los valores del resto de las iteraciones.
Primero se construye una tabla de 2*n+1 columnas para un polinomio de grado n, utilizando el siguiente esquema iterativo:

1) Se calculan los valores iniciales de la tabla:

$$q_0^{(1)} = \frac{-a_{n-1}}{a_n}$$

$$q_i^{(1)} = 0 \qquad i = 1, 2, 3, \ldots, n$$

$$\epsilon_i^0 = \frac{-a_{n-i-1}}{a_{n-i}} \quad i = 1, 2, 3, \ldots, n\text{-}1$$

$$\epsilon_0^0 = \epsilon_n^0 = 0$$

2) Habiendo calculado los valores iniciales, comenzando a iterar :

$$q_i^{(m+1)} = \epsilon_i^m - \epsilon_{i+1}^m + q_i^m \qquad i = 1, 2, 3, \ldots, n - 1$$

$$\epsilon_i^0 = \frac{q_{i+1}^{(m+1)}}{q_i^{(m+1)}} \qquad i = 1, 2, 3, \ldots, n\text{-}1$$

$$\epsilon_0^{(m+1)} = \epsilon_n^{(m+1)} = 0$$

con m como el número de iteraciones.

En medida que pasen las iteraciones, q_i tenderá a valores reales mientras que ϵ_i el valor del error tenderá a cero. En el otro caso, los valores de ϵ_i oscilarán lo cual indica que el polinomio posee raíces complejas y se debe recurrir al método de Bairstow.

Algoritmo de Bairstow :

Cuando un polinomio presenta raíces complejas es necesario recurrir a este método. En la tabla de cálculos en el algoritmo QD si hay raíces complejas, los valores de ϵ_i se muestran de forma oscilante. El método de Bairstow por sí mismo es un método que halla las raíces de un polinomio dividiendo por un factor cuadrático o lineal, obteniendo así la raíz cuando el residuo de dicha división es nulo. Cuando se aplica este método al algoritmo Q-D se busca encontrar un factor cuadrático para refinarlo con el método iterativo y luego a ese polinomio de grado 2 se le calcula las raíces con la fórmula resolvente y se obtienen las dos raíces conjugadas complejas.

- Polinomio que contiene el factor cuadrático

$$P_n(\mathrm{x}) = a_0 x^n + a_1 x^{n-1} + \dots + a_n x^0$$

- El factor cuadrático aproximado:

$$F_2(x) = x^2 - u \cdot x - v$$

- Una cota para el error E

Antes de ver el método se presenta una pequeña simplificación para comprender el algoritmo:

$\frac{P_n(x)}{F_2(x)} = P_{n-2}(x) + R\ (x)$ lo que se busca con este método es encontrar el polinomio F_2(x) cuya división obtenga un residuo nulo.

Para ello, se obtienen los coeficientes del factor cuadrático:

$$u_0 = v_0 = \frac{-a_{n-1}}{a_n}$$

se aplica la división sintética sobre el polinomio P_n(x) para obtener:

$P_{n-2}(\mathrm{x}) = q_2 + q_3 x^{n-1} + \dots + q_n x^{n-2}$

$\mathrm{R(x)} = q_0 + q_1 x - q_1 u_m$

Estos coeficientes se calculan como:

$q_t = a_t + u_m \cdot q_{t-1} + v_m \cdot q_{t-2}$ con t=0,1,2,...,n

se adopta $q_{-2} = q_{-1} = 0$

Luego se divide nuevamente por el factor cuadrático:

$P_{n-2}(\mathrm{x}) = P_{n-4}(\mathrm{x}) + R\ (x)$

con : $P_{n-4}(\mathrm{x}) = p_2 + p_3 x^{n-1} + \dots + p_n x^{n-4}$

Estos coeficientes se calculan como:

$p_t = q_t + u_m \cdot p_{t-1} + v_m \cdot p_{t-2}$ con t=0,1,2,...,n una vez que se obtienen los valores de q_t y p_t se hallan los valores de los incrementos para u_m y v_m, de la siguiente forma:

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

el cálculo de estos incrementos se obtuvo gracias al procedimiento demostrado por bairstow, el cual encuentra las derivadas parciales como una división sintética:

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

con n=1,2,3,...

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

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

de esta manera se obtiene refinar el factor cuadrático obtenido inicialmente.
Este proceso se detiene cuando, en alguna m-ésima iteración se obtienen valores de $\mathbf{q_n}$ y $\mathbf{q_{n-1}}$ menores que la cota de error deseada, en caso contrario, el proceso continúa calculando nuevos valores de **q** y **p**.

La implementación de este método requiere que todos los coeficientes del polinomio sean diferentes de 0 y que las raíces del mismo posean una separación adecuada. En los casos donde esto no se cumple se aplican métodos de preprocesamiento para poder aplicar el algoritmo. A continuación se introducen algunos de los métodos que se pueden aplicar:

Traslación efectuada sobre la indeterminada:

Este preprocesamiento se utiliza en los casos en los que el polinomio presenta coeficientes nulos. Se basa en sumarle una constante c a las x transformando el polinomio de la siguiente manera:

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

Generando otro polinomio de la forma:

$$Q(x) = P(x + c) = \sum_{k=0}^{n} b_k (x - c)^k$$

Siendo $b_k = \frac{P_k * c}{k!}$

Así, si c<0 el polinomio se desplaza a la izquierda y si c>0 se desplaza a la derecha.
Para una buena aplicación del preprocesamiento hay que tener en cuenta que c no sea igual en módulo a algún coeficiente del polinomio para no anularlo.

Una vez que se hallan las raíces del nuevo polinomio Q(x), se vuelve a restar sobre cada raíz el coeficiente c sumado anteriormente y así obtenemos las raíces de P(x) buscadas.

Transformación recíproca:

Este preprocesamiento se realiza en los casos en los que los polinomios tienen raíces cercanas. Este no se encuentra en el código del algoritmo pero se explica para comprender cómo proceder en estos casos.
Este método consiste en reemplazar las x del polinomio por 1/x y multiplicar al polinomio por x^n, siendo n el grado del polinomio, de la siguiente manera:

$$P(x) \rightarrow x^n * P(\frac{1}{x})$$

Resultando un nuevo polinomio de la forma:

$$Q(x) = x^n * P(\frac{1}{x}) = x^n * \sum_{k=0}^{n} a_k (\frac{1}{x})^k = \sum_{k=0}^{n} a_k * x^{n-k}$$

De esta manera, aplicando el algoritmo Q-D al polinomio resultante Q(x) se obtienen las raíces del polinomio P(x) de manera recíproca:

$$Raíces\ de\ Q(x) = Raíces\ de\ P(x)^{-1}$$

Análisis y resultados:

Caso 1: Polinomio completo con todas sus raíces reales.

$$f(x) = x^3 - 4x^2 + x + 6$$

- Raíces exactas:

$$x_1 = 3 \qquad x_2 = 2 \qquad x_3 = -1$$

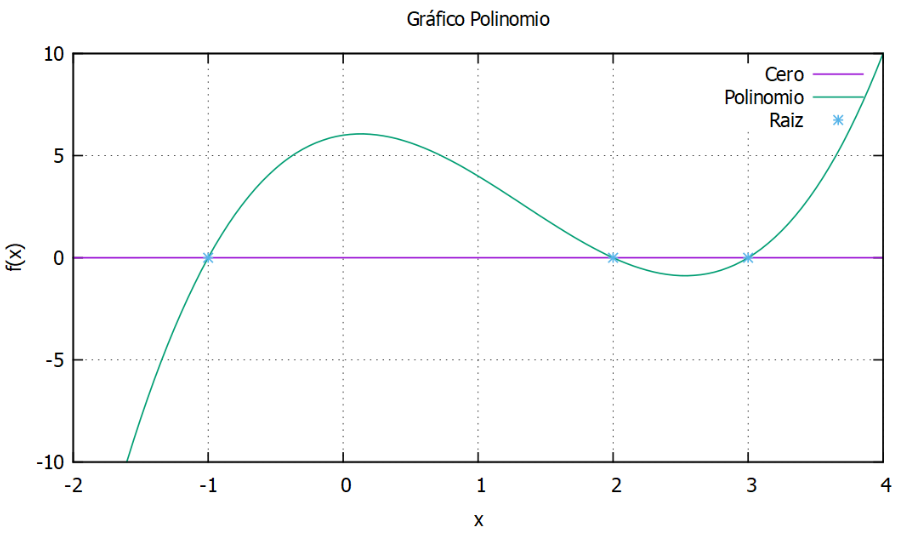
- Resultado obtenido con el algoritmo Q-D en 18 iteraciones con una tolerancia de 0.0001:

$$x_1 = 3.000401 \qquad x_2 = 1.999576 \qquad x_3 = -0.999977$$

Tabla 1: Resultados de las últimas iteraciones de q^i y e^i hasta su penúltima iteración del algoritmo Q-D para el polinomio $f(x) = x^3 - 4x^2 + x + 6$.

Iteración	e0	q1	e1	q2	e2	q3	e3
	0.00000		-0.00102		-0.00110		0.00000
14		3.00203		1.99760		-0.99963	
	0.00000		-0.00068		0.00055		0.00000
15		3.00136		1.99883		-1.00018	
	0.00000		-0.00045		-0.00027		0.00000
16		3.00090		1.99901		-0.99991	
	0.00000		-0.00030		0.00014		0.00000
17		3.00060		1.99944		-1.00005	
	0.00000		-0.00020		-0.00007		0.00000
18		3.00040		1.99958		-0.99998	
	0.00000		-0.00013		0.00003		0.00000

Gráfico 1: Gráfico del polinomio $f(x) = x^3 - 4x^2 + x + 6$ con las aproximaciones de sus raíces calculadas por el método Q-D.



Resolución con otros métodos:

- Método Punto Fijo Sistemático:

Se aplica el método bajo la misma tolerancia 0.0001, y luego de graficar el polinomio, se selecciona los intervalos para encontrar las raíces:

Para x_1:

Intervalo [2.5:5], con un $x_0 = 2.5$, en 93 iteraciones: $x_1 = 3.00000$

Intervalo [2.7:3.5], con un $x_0 = 2.7$ en 20 iteraciones: $x_1 = 3.00000$

Tabla 2: Resultados de la raíz x_1 del polinomio $f(x) = x^3 - 4x^2 + x + 6$ con los intervalos, x_0 e iteraciones mencionadas anteriormente, calculados mediante el método de punto fijo sistemático.

```
SOLUCION DE LA ECUACION POR EL METODO DE PUNTO FIJO SISTEMATICO

ITERACION    ERROR           ITERACION    ERROR

   .            .                  1        0.649246
   .            .                  2        0.500359
   .            .                  3        0.358243
  66         0.002248              4        0.241355
  67         0.001998              5        0.155312
  68         0.001776              6        0.096788
  69         0.001579              7        0.059054
  70         0.001404              8        0.035552
  71         0.001248              9        0.021227
  72         0.001110             10         0.012611
  73         0.000986             11         0.007470
  74         0.000877             12         0.004417
  75         0.000779             13         0.002609
  76         0.000693             14         0.001540
  77         0.000616             15         0.000909
  78         0.000548             16         0.000536
  79         0.000487             17         0.000316
  80         0.000433             18         0.000186
  81         0.000385             19         0.000110
  82         0.000342             20         0.000065
  83         0.000304
  84         0.000270
  85         0.000240              La solucion es     3.000
  86         0.000213
  87         0.000190              Se hicieron  20 iteraciones
  88         0.000169
  89         0.000150
  90         0.000133
  91         0.000118
  92         0.000105
  93         0.000094

La solucion es     3.000

Se hicieron  93 iteraciones
```

Para x_2:

Intervalo [1.8:2.2], con un $x_0 = 1.8$, en 11 iteraciones: $x_2 = 2.0000$

Intervalo [1.9:2.1], con un $x_0 = 1.9$, en 5 iteraciones: $x_2 = 2.0000$

Tabla 3: Resultados de la raíz x_2 del polinomio $f(x) = x^3 - 4x^2 + x + 6$ con los intervalos, x_0 e iteraciones mencionadas anteriormente, calculados mediante el método de punto fijo sistemático.

```
SOLUCION DE LA ECUACION POR EL METODO DE PUNTO FIJO SISTEMATICO

ITERACION    ERROR             ITERACION    ERROR

   1        0.337071            1        0.071196
   2        0.119908            2        0.010760
   3        0.055311            3        0.001825
   4        0.023886            4        0.000305
   5        0.010664            5        0.000051
   6        0.004696
   7        0.002081
   8        0.000920
   9        0.000407            La solucion es     2.000
  10        0.000180
  11        0.000080            Se hicieron   5 iteraciones

La solucion es     2.000

Se hicieron  11 iteraciones
```

Para x_3:

Intervalo [-5:0], con un $x_0 = -5$, en 106 iteraciones: $x_3 = -1.0000$

Intervalo [-2:0], con un $x_0 = -2$, en 21 iteraciones: $x_3 = -1.0000$

Tabla 4: Resultados de la raíz x_3 del polinomio $f(x) = x^3 - 4x^2 + x + 6$ con los intervalos, x_0 e iteraciones mencionadas anteriormente, calculados mediante el método de punto fijo sistemático.

```
SOLUCION DE LA ECUACION POR EL METODO DE PUNTO FIJO SISTEMATICO

ITERACION    ERROR            ITERACION    ERROR
   .           .
   .           .                1        4.428226
   .           .                2        2.069655
  71        0.004571            3        1.088111
  72        0.004098            4        0.601864
  73        0.003674            5        0.341571
  74        0.003293            6        0.196567
  75        0.002953            7        0.114007
  76        0.002647            8        0.066419
  77        0.002373            9        0.038795
  78        0.002128            10        0.022694
  79        0.001907            11        0.013287
  80        0.001710            12        0.007783
  81        0.001533            13        0.004561
  82        0.001375            14        0.002673
  83        0.001232            15        0.001567
  84        0.001105            16        0.000918
  85        0.000991            17        0.000538
  86        0.000888            18        0.000316
  87        0.000796            19        0.000185
  88        0.000714            20        0.000108
  89        0.000640            21        0.000064
  90        0.000574
  91        0.000514             La solucion es    -1.000
  92        0.000461
  93        0.000413             Se hicieron  21 iteraciones
  94        0.000371
  95        0.000332
  96        0.000298
  97        0.000267
  98        0.000240
  99        0.000215
 100        0.000193
 101        0.000173
 102        0.000155
 103        0.000139
 104        0.000124
 105        0.000112
 106        0.000100

La solucion es    -1.000

Se hicieron 106 iteraciones
```

- Método de Newton-Raphson

Aplicamos el método bajo la misma tolerancia 0.0001, para este método seleccionamos los valores iniciales como los mismos que se tomaron en punto fijo sistemático para obtener:

Para x_1:

Valor inicial $x_0 = 2.5$ el programa converge erróneamente a la raíz -1, se cambió el valor inicial para corregir la convergencia.

Valor inicial $x_0 = 2.6$, en 7 iteraciones: $x_1 = 3.00000$

Tabla 5: Resultados de la raíz x_1 del polinomio $f(x) = x^3 - 4x^2 + x + 6$ con el x_0 e iteraciones mencionadas anteriormente, calculados mediante el Método de Newton-Raphson.

```
SOLUCION DE LA ECUACION POR EL METODO DE NEWTON

ITERACION    ERROR
   1        1.800000
   2        0.759799
   3        0.418901
   4        0.179390
   5        0.039892
   6        0.002013
   7        0.000005

La solucion es   3.00000

Se hicieron   7 iteraciones
```

Para x_2:

Valor inicial $x_0 = 1.5$, en 4 iteraciones: $x_2 = 2.00000$

Tabla 6: Resultados de la raíz x_2 del polinomio $f(x) = x^3 - 4x^2 + x + 6$ con el x_0 e iteraciones mencionadas anteriormente, calculados mediante el Método de Newton-Raphson.

```
SOLUCION DE LA ECUACION POR EL METODO DE NEWTON

ITERACION    ERROR
   1        0.441176
   2        0.056804
   3        0.002017
   4        0.000003

La solucion es   2.00000

Se hicieron   4 iteraciones
```

Para x_3:

Valor inicial $x_0 =- 5.0$, en 7 iteraciones: $x_3 = 1.00000$

Tabla 7: Resultados de la raíz x_3 del polinomio $f(x) = x^3 - 4x^2 + x + 6$ con el x_0 e iteraciones mencionadas anteriormente, calculados mediante el Método de Newton-Raphson.

```
SOLUCION DE LA ECUACION POR EL METODO DE NEWTON

ITERACION    ERROR
   1        1.931034
   2        1.182893
   3        0.628706
   4        0.225866
   5        0.030937
   6        0.000563
   7        0.000000

La solucion es  -1.00000

Se hicieron   7 iteraciones
```

Se pudo observar que, con una misma tolerancia se obtienen mejores resultados con el método de Newton-Raphson y punto sistemático fijo que con el algoritmo Q-D. Pero estos dos primeros métodos no calculan las raíces simultáneamente y se le tiene que dar un intervalo en el cual se encuentra la raíz buscada, por lo tanto se debería conocer el resultado antes de la aplicación de los métodos.

Caso 2: Polinomio completo con raíces reales y complejas.

$$f(x) = x^5 - 3.5x^4 + 2.75x^3 + 2.125x^2 - 3.875x + 1.25$$

- Raíces exactas:

$$x_1 = 2 \qquad x_2 = 1 + 0.5i \qquad x_3 = 1 - 0.5i$$
$$x_4 = -1 \qquad x_5 = 0.5$$

- Resultado con el método Q-D en 100 iteraciones con una tolerancia de 0.0001:

$$x_1 = 2.00000 \qquad x_2 = 1.00000 + 0.50000i \quad x_3 = 1.00000 - 0.500000i$$

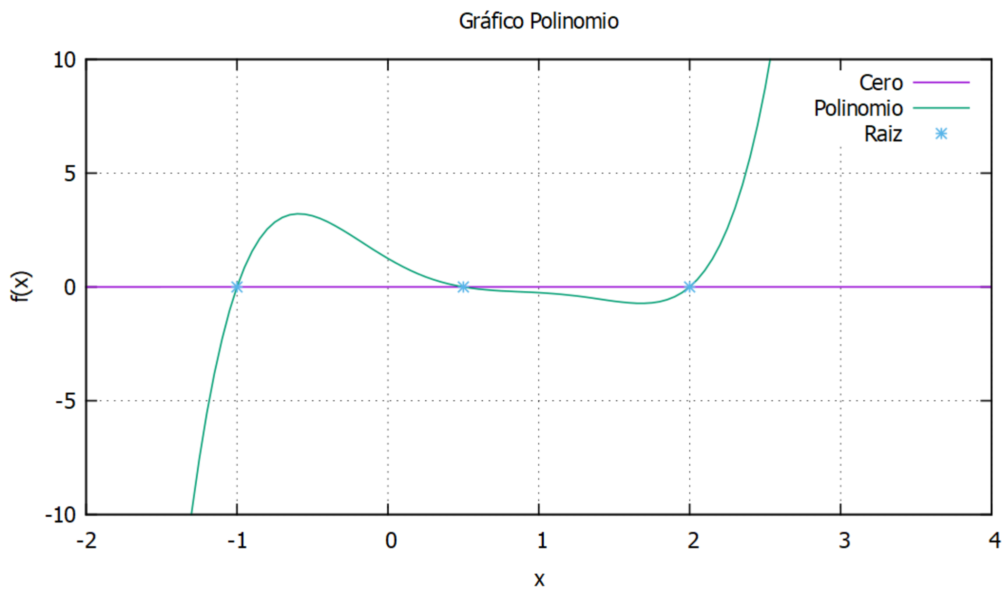
$$x_4 = -1.00006 \qquad x_5 = 0.50000$$

Con el método de Bairstow se hallaron x_2 y x_3.

Tabla 8: Resultados de las últimas iteraciones de q^i y e^i hasta su penúltima iteración del algoritmo Q-D para el polinomio $f(x) = x^5 - 3.5x^4 + 2.75x^3 + 2.125x^2 - 3.875x + 1.25$.

Iteración	e0	q1	e1	q2	e2	q3	e3	q4	e4	q5	e5
	0.00000		0.00000		-11.95024		-0.00021		-0.00000		0.00000
95		2.00000		-11.86005		13.86000		-0.99995		0.50000	
	0.00000		-0.00000		13.96540		0.00002		0.00000		0.00000
96		2.00000		2.10535		-0.10538		-0.99996		0.50000	
	0.00000		-0.00000		-0.69905		0.00015		-0.00000		0.00000
97		2.00000		1.40630		0.59381		-1.00011		0.50000	
	0.00000		-0.00000		-0.29517		-0.00025		0.00000		0.00000
98		2.00000		1.11112		0.88874		-0.99986		0.50000	
	0.00000		-0.00000		-0.23610		0.00028		-0.00000		0.00000
99		2.00000		0.87503		1.12511		-1.00014		0.50000	
	0.00000		-0.00000		-0.30357		-0.00025		0.00000		0.00000
100		2.00000		0.57146		1.42843		-0.99989		0.50000	

Gráfico 2: Gráfico del polinomio $f(x) = x^5 - 3.5x^4 + 2.75x^3 + 2.125x^2 - 3.875x + 1.25$ con las aproximaciones de sus raíces reales calculadas por el método Q-D.



Resolución con otros métodos:

- Método Punto Fijo Sistemático:

Este método no puede aplicarse para raíces complejas ya que estas raíces no intersectan al eje x por consiguiente el método diverge cuando se calcula g(x) y se la intersecta con y = x.

- Resultado con Newton Raphson:

Para aplicar este método en raíces complejas se tuvo que modificar el algoritmo, de tal manera que acepte valores complejos, se obtiene :

Para x_1:

Valor inicial $x_0 = -1.5$, en 5 iteraciones: $x_1 = -1.00000$

Para x_2:

Valor inicial $x_0 = 1.7$, en 12 iteraciones: $x_2 = 2.00000$

Para x_3:

Valor inicial $x_0 = 1.0$, en 5 iteraciones: $x_3 = 0.50000$

Para x_4:

Valor inicial $x_0 = 0.8 + 0.4i$ en 6 iteraciones: $x_4 = 1.00000 + 0.50000\,i$

Tabla 9: Resultados de la raíz x_4 del polinomio $f(x) = x^5 - 3.5x^4 + 2.75x^3 + 2.125x^2 - 3.875x + 1.25$ con el x_0 e iteraciones mencionadas anteriormente, calculados mediante el Método de Newton-Raphson.

```
 RESOLUCION MEDIANTE EL METODO DE NEWTONRAPSHON
 ---------------------------------------

Iteracion: 1    Valor de x: 1.079369 + 0.311924i
Error en y:  0.2239629
Error en x : 0.2929239
 ---------------------------------------
Iteracion: 2    Valor de x: 0.857584 + 0.480486i
Error en y:  0.2028442
Error en x : 0.2785707
 ---------------------------------------
Iteracion: 3    Valor de x: 1.004271 + 0.437572i
Error en y:  0.0873977
Error en x : 0.1528351
 ---------------------------------------
Iteracion: 4    Valor de x: 0.994133 + 0.510672i
Error en y:  0.0203099
Error en x : 0.0737990
 ---------------------------------------
Iteracion: 5    Valor de x: 0.999643 + 0.500124i
Error en y:   0.0006155
Error en x :  0.0119006
 ---------------------------------------
Iteracion: 6    Valor de x: 1.000000 + 0.500000i
Error en y:  0.0000006
Error en x : 0.0003776
 ---------------------------------------

La raiz encontrada es:   1.000000  0.500000
```

Para x_5:

Valor inicial $x_0 = 0.8 - 0.4i$ en 6 iteraciones: $x_5 = 1.00000 - 0.50000i$

Tabla 10: Resultados de la raíz $x_5$del polinomio

$f(x) = x^5 - 3.5x^4 + 2.75x^3 + 2.125x^2 - 3.875x + 1.25$ con el x_0 e iteraciones mencionadas anteriormente, calculados mediante el Método de Newton-Raphson.

```
 RESOLUCION MEDIANTE EL METODO DE NEWTONRAPSHON
 --------------------------------------
Iteracion: 1   Valor de   1.079369 -0.311924 i
Error en y:   0.223963
Error en x :   0.292924
 --------------------------------------
Iteracion: 2   Valor de   0.857584 -0.480486 i
Error en y:   0.202844
Error en x :   0.278571
 --------------------------------------
Iteracion: 3   Valor de   1.004271 -0.437572 i
Error en y:   0.087398
Error en x :   0.152835
 --------------------------------------
Iteracion: 4   Valor de   0.994133 -0.510672 i
Error en y:   0.020310
Error en x :   0.073799
 --------------------------------------
Iteracion: 5   Valor de   0.999643 -0.500124 i
Error en y:   0.000616
Error en x :   0.011901
 --------------------------------------
Iteracion: 6   Valor de   1.000000 -0.500000 i
Error en y:   0.000001
Error en x :   0.000378
 --------------------------------------
La raiz encontrada es:   1.000000 -0.500000 i
```

Caso 3: Polinomio completo con raíces cercanas:

$$f(x) = x^3 - 21.009x^2 + 147.126026x - 343.441182$$

- Raíces exactas:

$x_1 = 7.002$ $\quad$ $x_2 = 7.003$ $\quad$ $x_3 = 7.004$

- Resultado obtenido con el algoritmo Q-D en 373 iteraciones con una tolerancia de 0.0001:

$x_1 = 7.04045$ $\quad$ $x_2 = 7.00291$ $\quad$ $x_3 = 6.96564$

Tabla 11: Resultados de las últimas iteraciones de q^i y e^i hasta su penúltima iteración del algoritmo Q-D para el polinomio $f(x) = x^3 - 21.009x^2 + 147.126026x - 343.441182$.

Iteración	e0	q1	e1	q2	e2	q3	e3
	0.00000		-0.00010		-0.00010		0.00000
367		7.04107		7.00290		6.96503	
	0.00000		-0.00010		-0.00010		0.00000
368		7.04096		7.00290		6.96513	
	0.00000		-0.00010		-0.00010		0.00000
369		7.04086		7.00290		6.96524	
	0.00000		-0.00010		-0.00010		0.00000
370		7.04076		7.00291		6.96534	
	0.00000		-0.00010		-0.00010		0.00000
371		7.04066		7.00291		6.96544	
	0.00000		-0.00010		-0.00010		0.00000
372		7.04056		7.00291		6.96554	
	0.00000		-0.00010		-0.00010		0.00000

Gráfico 3: Gráfico del polinomio $f(x) = x^3 - 21.009x^2 + 147.126026x - 343.441182$ con las aproximaciones de sus raíces reales calculadas por el método Q-D.

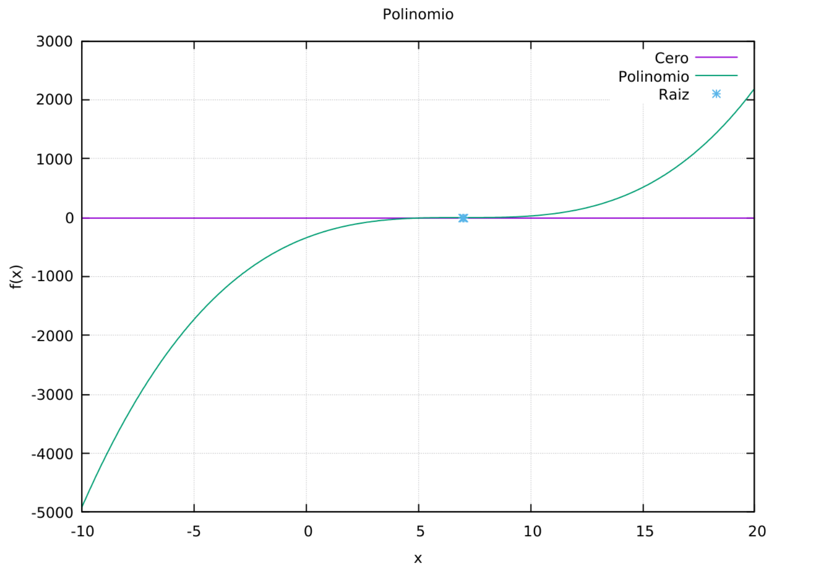
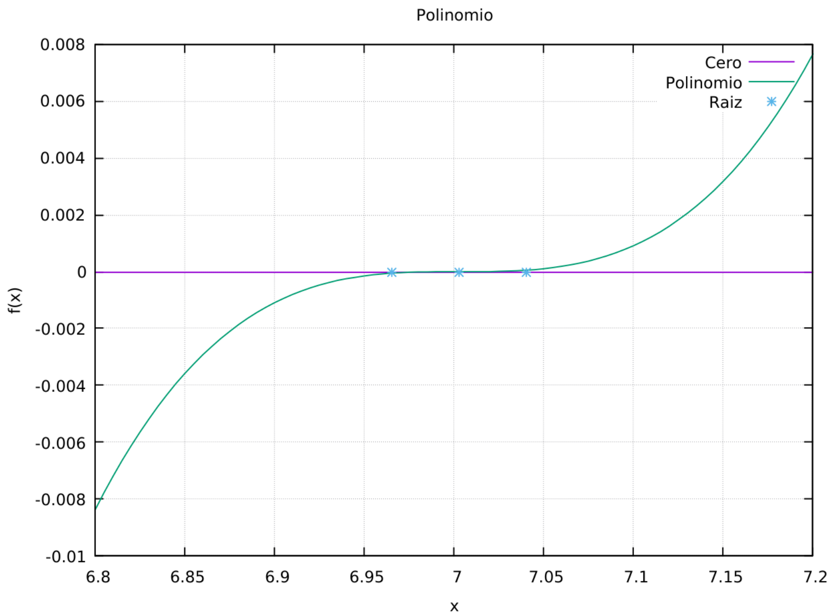


Gráfico 4: Acercamiento del gráfico 3.



Resolucion con otros métodos:

- Método Punto Fijo Sistemático:

Para este caso donde las raíces del polinomio están muy cerca una de otra, este método no nos es útil ya que convergerá rápidamente a una de las raíces pero habrá divergencia cuando se ajuste el intervalo para hallar las cercanas.

- Método de Bisección:

Para este caso se implementó este método ya que las raíces son muy cercanas y este método permite fijar el intervalo donde se va a buscar la raíz, con una tolerancia de 0.0001

Para x_1:
Intervalo seleccionado : [7.001:7.0023], en 4 iteraciones: $x_1 = 7.002$
Para x_2:
Intervalo seleccionado : [7.002:7.0035], en 4 iteraciones: $x_2 = 7.002$

Para x_3:
Intervalo seleccionado : [7.003:7.0045], en 4 iteraciones: $x_3 = 7.004$

Caso 4: Polinomio incompleto con raíces reales y co-modulares:

$$f(x) = x^4 - 10x^2 + 9$$

- Raíces exactas:

$x_1 = 3$ $x_2 = 1$ $x_3 = -1$ $x_4 = -3$

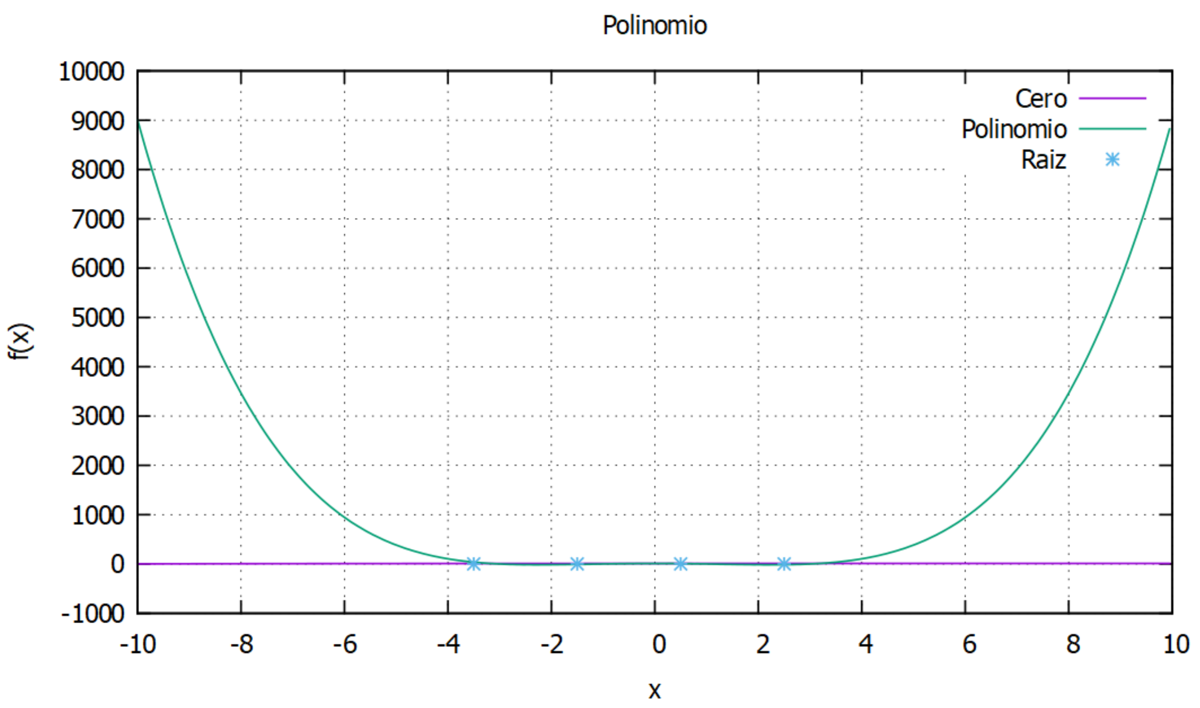
- Resultado con el método Q-D, en 32 iteraciones con una tolerancia de 0.0001 y utilizando un factor de traslación c=0.5:

$x_1 = -3.49995$ $x_2 = 2.49995$ $x_3 = -1.50000$ $x_4 = 0.50$

Tabla 12: Resultados de las últimas iteraciones de q^i y e^i hasta su penúltima iteración del algoritmo Q-D para el polinomio $f(x) = x^4 - 10x^2 + 9$, con un c=0.5.

Iteración	e0	q1	e1	q2	e2	q3	e3	q4	e4
	0.00000		-0.00059		0.00000		-0.00000		0.00000
27		-3.50025		2.50025		-1.50000		0.50000	
	0.00000		0.00043		-0.00000		0.00000		0.00000
28		-3.49982		2.49982		-1.50000		0.50000	
	0.00000		-0.00030		0.00000		-0.00000		0.00000
29		-3.50013		2.50013		-1.50000		0.50000	
	0.00000		0.00022		-0.00000		0.00000		0.00000
30		-3.49991		2.49991		-1.50000		0.50000	
	0.00000		-0.00015		0.00000		-0.00000		0.00000
31		-3.50006		2.50006		-1.50000		0.50000	
	0.00000		0.00011		-0.00000		0.00000		0.00000

Gráfico 5: Gráfico del polinomio $f(x) = x^4 - 10x^2 + 9$, con un c=0.5, con las aproximaciones de sus raíces reales calculadas por el método Q-D.



Se calculó nuevamente con el método Q-D en 1751 iteraciones, con una tolerancia de 0.0001 y un factor de traslación c=0.01, resultando:

$$x_1 = -3.01005 \quad x_2 = 2.99005 \quad x_3 = -1.01000 \quad x_4 = 0.99000$$

Tabla 13: Resultados de las últimas iteraciones de q^i y e^i hasta su penúltima iteración del algoritmo Q-D para el polinomio $f(x) = x^4 - 10x^2 + 9$, con un c=0.01.

Iteración	e0	q1	e1	q2	e2	q3	e3	q4	e4
	0.00000		-0.00010		0.00000		-0.00000		0.00000
1745		-3.01005		2.99005		-1.01000		0.99000	
	0.00000		0.00010		-0.00000		0.00000		0.00000
1746		-3.00995		2.98995		-1.01000		0.99000	
	0.00000		-0.00010		0.00000		-0.00000		0.00000
1747		-3.01005		2.99005		-1.01000		0.99000	
	0.00000		0.00010		-0.00000		0.00000		0.00000
1748		-3.00995		2.98995		-1.01000		0.99000	
	0.00000		-0.00010		0.00000		-0.00000		0.00000
1749		-3.01005		2.99005		-1.01000		0.99000	
	0.00000		0.00010		-0.00000		0.00000		0.00000
1750		-3.00995		2.98995		-1.01000		0.99000	
	0.00000		-0.00010		0.00000		-0.00000		0.00000

Se pudo observar que al aumentar el c, el número de iteraciones disminuye pero la aproximación a las raíces es menos exacta. Utilizar un factor de traslación reducido implica mayor esfuerzo del programa al ejecutarlo pero se logra una mayor exactitud en el cálculo.

Resolución con otros métodos:

- Método Punto Fijo Sistemático:

Aplicamos el método bajo la misma tolerancia 0.0001, y luego de graficar el polinomio, seleccionamos los intervalos para encontrar las raíces

Para x_1

Intervalo [2.0:4.0], con un x_0=2.0 en 47 iteraciones: x_1=2.999998
Intervalo [2.7:3.5], con un x_0=2.7 en 20 iteraciones: x_1=2.999999

Para x_2

Intervalo [0.5:1.9], con un x_0=0.5 en 29 iteraciones: x_2=1.000005
Intervalo [0.7:1.5], con un x_0=0.7 en 8 iteraciones: x_2=0.999996

Para x_3

Intervalo [-1.9:-0.5], con un x_0=-1.9 en 5 iteraciones: x_3=-1.000002
Intervalo [-1.5:-0.7], con un x_0=-1.5 en 4 iteraciones x_3=-1.000003

Para x_4

Intervalo [-4.0:-2.0], con un x_0=-4.0 en 38 iteraciones: x_4=-3.000002
Intervalo [-3.5:-2.7], con un x_0=-3.5 en 18 iteraciones: x_4=-3.000002

Se puede observar que el método punto fijo no tuvo problemas para hallar las raíces aunque el polinomio no esté completo y lo halló con menos iteraciones.

Caso 5: Polinomio completo con raíces reales co-modulares:

$$f(x) = x^4 + 2x^3 - 3x^2 - 8x - 4$$

- Raíces exactas:

$x_1 = 2$ $\quad$ $x_2 =- 2$ $\quad$ $x_3 =- 1$ $\quad$ $x_4 =- 1$

- Resultado con el método Q-D en 100 iteraciones con una tolerancia de 0.0001:

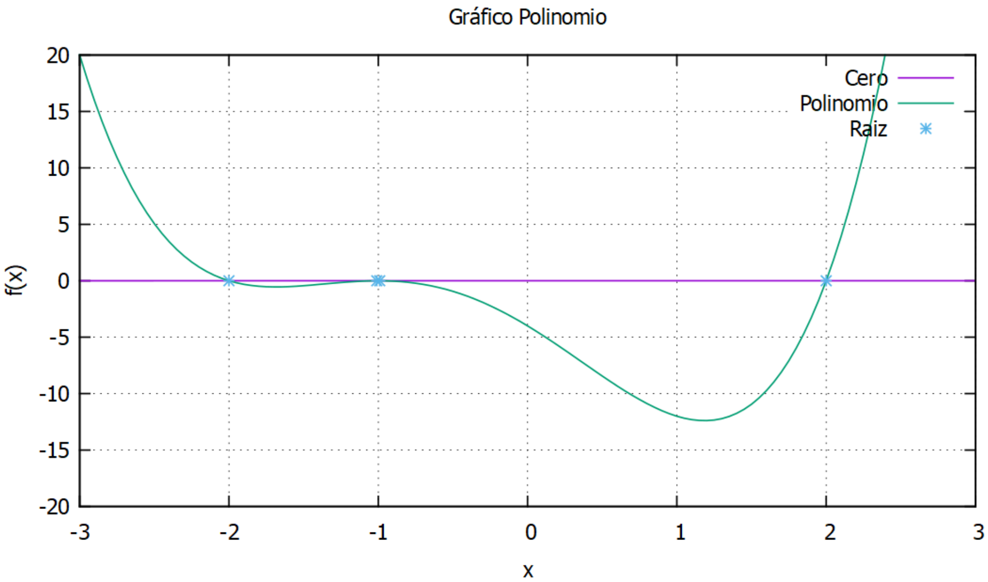
$x_1 = 2.000000$ $\quad$ $x_2 =- 2.000000$ $\quad$ $x_3 =- 1.009868$ $\quad$ $x_4 =- 0.990132$

Con el método de Bairstow se hallaron x_1 y x_2.

Tabla 14: Resultados de las últimas iteraciones de q^i y e^i hasta su penúltima iteración del algoritmo Q-D para el polinomio $f(x) = x^4 + 2x^3 - 3x^2 - 8x - 4$.

Iteración	e0	q1	e1	q2	e2	q3	e3	q4	e4
	0.00000		-0.90000		0.00000		0.00011		0.00000
95		-2.50000		2.50000		-1.01049		-0.98951	
	0.00000		0.90000		-0.00000		0.00011		0.00000
96		-1.60000		1.60000		-1.01038		-0.98962	
	0.00000		-0.90000		0.00000		0.00011		0.00000
97		-2.50000		2.50000		-1.01027		-0.98973	
	0.00000		0.90000		-0.00000		0.00010		0.00000
98		-1.60000		1.60000		-1.01017		-0.98983	
	0.00000		-0.90000		0.00000		0.00010		0.00000
99		-2.50000		2.50000		-1.01007		-0.98993	
	0.00000		0.90000		-0.00000		0.00010		0.00000
100		-1.60000		1.60000		-1.00997		-0.99003	
	0.00000		-0.90000		0.00000		0.00010		0.00000

Gráfico 6: Gráfico del polinomio $f(x) = x^4 + 2x^3 - 3x^2 - 8x - 4$ con las aproximaciones de sus raíces reales calculadas por el método Q-D.



Resolución con otros métodos:

- Método Punto Fijo Sistemático:

Aplicamos el método bajo la misma tolerancia 0.0001, y luego de graficar el polinomio, seleccionamos los intervalos para encontrar las raíces.

Para x_1:

Intervalo [1.0;3.0], con un x_0=1.0 en 48 iteraciones: x_1=1.999998

Intervalo [1.5;2.5], con un x_0=1.5 en 21 iteraciones: x_1=1.999998

Para x_2

Intervalo [-2.2;-1.9], con un x_0=-2.2, en 22 iteraciones: x_2=-1.999976

Intervalo [-2.2;-1.9], con un x_0=-2.0, en 10 iteraciones: x_2=-2.00000

Para x_3 , x_4

Intervalo [-1.5:-0.5], con un x_0=-1.5 en 65 iteraciones: x_3=-1.005746

Intervalo [-1.2:-0.8], con un x_0=-1.2 en 49 iteraciones: x_3=-1.005756

Conclusiones:

El método Quotient-difference (QD) es un método de obtención de raíces de polinomio con la ventaja de conseguir todas las raíces en un mismo proceso, a diferencia de otros métodos como deflación donde se calcula las raíces una a una, añadiendo complejidad y perdiendo eficacia, ya que en los casos analizados se tuvo que añadir un valor inicial cercano a la raíz buscada o un intervalo donde se encuentra de la misma, suponiendo que se conocen los valores de las raíces buscadas. Esto hace al método QD bastante atractivo para polinomios de alto grado, especialmente si posee todos sus coeficientes no nulos.

Un inconveniente de este método es que se necesita que el polinomio posea coeficientes no nulos, de no ser así se deben emplear otros métodos de preprocesamiento para compensar esa falta, pero aun así es posible alcanzar el resultado deseado.

Para polinomios de grado bajo es preferible emplear otros métodos vistos dado que, como se pudo observar en los datos vistos en cada caso, se puede llegar al resultado a menor número de iteraciones por los otros métodos.

En presencia de raíces co-modulares y complejas es necesario aplicar el método de Bairstow, dado que el método QD por sí mismo no posee herramientas para resolver dichas raíces, se puede observar cuando las columnas de error fluctúan en su valor en lugar de tender a anularse.

En conclusión, el algoritmo Quotient-difference es un método con problemas para ciertos casos que requieren que el polinomio pase por un preprocesamiento o recurrir a un método extra para poder llegar al resultado, pero también a ciertas escalas es una buena opción para trabajar sobre todo si se quiere obtener varias raíces o no se cuenta con valor de cercanía a las mismas.