Algoritmo Q-D (Quotient Difference)

Este Algoritmo permite obtener todas las raíces de un polinomio, es decir permite aproximar a todos factores lineales y cuadráticos, en los cuales se pueda expresar un polinomio de grado **n**.
Dado un polinomio con **coeficientes reales y completo**, es decir <u>con todos sus coeficientes diferentes de cero</u>, armaremos una tabla de valores, utilizando dos conjuntos de fórmulas, la primera, para calcular los valores de la iteración cero, y la segunda, para calcular los valores de de las demás iteraciones.

Para ésto utilizaremos el siguiente esquema iterativo :

Cálculo de los valores de q_i y e_i para la iteración 0.

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

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

Cálculo de los valores de q_i y e_i para el resto de las iteraciones.

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

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

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

Disposición de valores en la Tabla de QD

Iter.	e_0	q_1	e_1	q_2	....	e_{n-1}	q_n	e_n
0		$\frac{-a_{n-1}}{a_n}$		0	.....		0	
	0		$\frac{a_{n-i-1}}{a_{n-i}}$		.....	0		0
1		$q_1^{(1)}$		$q_2^{(1)}$	.....		$q_n^{(1)}$	
	0		$e_1^{(1)}$		.....	$e_2^{(1)}$		0
2		$q_1^{(2)}$		$q_2^{(2)}$	.....		$q_n^{(2)}$	
	0		$e_1^{(2)}$		.....	$e_2^{(2)}$		0

Ejemplo Numérico:

Dado un polinomio con coeficientes reales y completo, es decir <u>con todos sus coeficientes diferentes de cero</u>, armaremos 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. Dado el polinomio.

$$p(x) = 128x^4 - 256x^3 + 160x^2 - 32x + 1$$

e_0	q_1	e_1	q_2	e_2	q_3	e_3	q_4	e_4
	2		0		0		0	
0		-0.625		-0.2		-0.03125		0
	1.375		0.425		0.16875		0.03125	
0		-0.19318		-0.07941		-0.00579		0
	1.18182		0.53877		0.24237		0.03704	
0		-0.08807		-0.03572		-0.00088		0
	1.09375		0.59111		0.27722		0.03792	
0		-0.0476		-0.01675		-0.00012		0
	1.04615		0.62196		0.29384		0.03804	
0		-0.0283		-0.00792		-0.00002		0
	1.01786		0.64234		0.30175		0.03806	
0		-0.01786		-0.00792		-0.00000		0
	1		0.65476		0.30546		0.03806	
0		-0.01172		-0.00173		-0.00000		0
	0.98828		0.66647		0.30719		0.03806	
0		-0.00791		-0.0008		-0.00000		0
	0.98037		0.67358		0.3799		0.03806	
0		-0.00543		-0.00036		-0.00000		0

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. En este caso, para determinar los valores de dichas raíces, es necesario recurrir al método de Bairstow. Éste algoritmo permite refinar el factor cuadrático de un polinomio, es decir aquel que contiene raíces co-modulares *(de igual módulo)*, ya sean complejas o reales. Para ello partimos del siguiente conjunto de datos :

a) El polinomio que contiene al factor cuadrático: $p(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \cdots + a_n$
b) el factor cuadrático aproximado $x^2 - u \cdot x - v$
c) una cota para el error ε

Partiendo de $\mathbf{u_0 = -\,a_{n-1}/a_n}$ y $\mathbf{v_0 = -\,a_{n-1}/a_n}$, la idea consiste en obtener una sucesión de valores de $\mathbf{u_i}$ y $\mathbf{v_i}$ cada vez mas aproximados, de forma tal, que una vez que se hayan obtenido con la exactitud deseada, nos permitan hallar las raíces que conforman dicho factor cuadrático.

Para ello, hay que calcular en cada iteración los valores de $\mathbf{q_t}$ y $\mathbf{p_t}$, para t = 0,1,2, ... ,n por medio de las siguientes expresiones :

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

con los valores de $\mathbf{q_t}$ y $\mathbf{p_t}$ calculados, podemos obtener los valores de **h** y **k** , que son solución del siguiente sistema de ecuaciones lineales :

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

*o sea, despejando **h** y **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 por último calculamos los nuevos valores de **u** y **v**, por medio de la siguiente expresión :

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

Este proceso se detiene cuando, en alguna **m-ésima** iteración obtenemos valores de $\mathbf{q_n}$ y $\mathbf{q_{n-1}}$ menores que la **cota de error** deseada, en caso contrario, continuamos con el proceso de calcular nuevos valores de **q** y **p**.

Consideraciones para el Trabajo Final :

Con respecto al tema planteado, el mismo ofrece varias alternativas para explorar

- Implementación del algoritmo y comparación con otros métodos de obtención de raíces de polinomios vistos en la asignatura.

- Este método requiere que todos los coeficientes del polinomio sean diferentes de cero y que las raíces del mismo posean una separación adecuada por lo que es necesaria la implementación de métodos de preprocesamiento de ciertos polinomios para lograr este objetivo.

Para cualquier consulta con respecto a este tema comunicarse con Francisco Lizarralde por medio de la mensajería del Campus.