

Tarea 6 - Problema 1 - Programación avanzada

Nicolás Aylwin

Para este problema creé un archivo de funciones template llamado “ceros_poli.h” especializado en calcular raíces de polinomio. Dentro de esta usé mis clases de polinomios, matrices y complejos. Todas las funciones tienen dos posibles tipos (doble template) por si quiero encontrar raíces complejas de polinomios con coeficientes reales (por ejemplo).

Todas las funciones de los métodos se programaron de forma estándar y son comprensibles solo con los comentarios. Por otro lado, hay dos funciones especiales; una que permite encontrar todas las raíces de un polinomio con Newton-Raphson o Bisección, esto a través de la división de polinomios, se parte con el polinomio original y luego de encontrar una raíz α con algún método se divide este por $(x - \alpha)$, lo que permite (de forma iterativa), encontrar todas las raíces.

En el caso de Newton-Raphson se elegía siempre la misma semilla y el proceso de división aseguraba obtener siempre distintas raíces. Para la bisección se usó la otra función exótica, que se encarga de trozar el intervalo específico de interés $([-1,1])$ y encontrar los nuevos intervalos en los cuales realizar la bisección.

Cabe destacar que Durand-Kerner por si solo ya calcula todas las raíces, así que se usó así simplemente.

Esta clase es importada al archivo “problema1.cc” donde se realiza lo pedido. Se ingresa el grado del polinomio de Legendre a través de un argumento en main y se entrega el resultado en pantalla especificando cada método usado.

Se realiza una “limpieza” de raíces si alguna de estas es muy cercana a cero ($< 10^{-12}$), y se lleva a cero directamente. Además para el método de Durand-Kerner se generan raíces complejas pero se presenta solo su parte real (puesto la imaginaria claramente es solo ruido).

Para compilar solo hace falta hacer make.