Tarea 3 - Problema 1 - Programación avanzada

Nicolás Aylwin

Buscamos los complejos $C = a + bi$ con $a, b \in [-2, 2]$ tal que se cumpla que en la siguiente iteración:

$$z_0 = 0$$
$$z_{n+1} = z^m{}_n + C$$

El módulo de z_n sea menor a 2 en cada iteración luego de 100 de estas.

Para lograr esto, se creó un pequeño programa; consta de dos funciones, siendo una de estas 'main'. La primera llamada 'itera', recibe un complejo (constante y por referencia, para tener cuidado de cambiarlo, y mejorar el rendimiento respectivamente) y un entero. Estos corresponden a C y m. En esta función se realiza la recurrencia en un ciclo 'while' con la condición sobre z_n. De romperse esta, se entrega un 1, dando a entender que con ese C y esa m la recurrencia diverge. Dentro del bucle también se suma un contador por cada iteración, que al llegar a 100 es evaluado por un 'if' para devolver un 0, dando a entender que con ese C y esa m convergió (o al menos no rompió la condición en 100 iteraciones).

En 'main' primero se le pide al usuario el valor de m, este es almacenado en una variable tipo 'int' y esta a su vez convertida a 'string' para crear un archivo de texto de nombre 'no_ divergen_ m.txt'. Además se crean un complejo 'c' y un recipiente de 'int' llamado 'res'. Luego se entra a dos ciclos 'for' anidados, para recorrer toda la grilla en que puede estar C.

Se crea un C, se introduce en 'itera' junto al m ingresado por el usuario, se recibe en 'res' y se estudia el valor de este. De ser cero se incluye la parte real y la imaginaria de C en el archivo .txt, luego se avanza un 1/10000 en la parte real y se repite, al recorrer todos los valores de la parte real se avanza lo mismo pero en la parte imaginaria y se repite el proceso.

Dado esto, el programa demora varios minutos en terminar de calcular, por esto se imprime a pantalla cada C a analizar, así darle a entender al usuario que si están pasando cosas y no se quedo pegado su computador simplemente.

Luego de generar los archivos para $m = \{1, 2, 3, 4\}$, se graficaron estos en formato .png con Gnuplot. Solo se incluyen el ejecutable, el archivo de texto que genera el ejecutable y los gráficos, pues cada .txt (excepto el de $m = 1$) pesan alrededor de 200mb.