Herramientas Computacionales

Taller 7 - Python - Método de Newton-Raphson

marzo de 2015

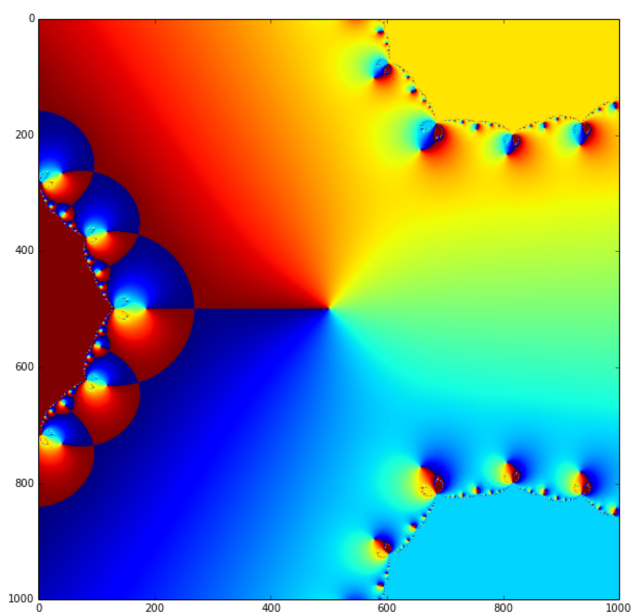
La solución de este taller debe ser presentada en un solo archivo comprimido de nombre `NombreApellido_HW7.tar.gz`, en el cual esté contenido el notebook de iPython con todo el código que solucione la tarea y las dos imágenes: `fractal_lr.png` y `fractal_hr.png`. Para cada punto se dará 1/3 del puntaje si el código corre, 1/3 si el código tiene sentido y 1/3 si el resultado es el apropiado.

1. $\boxed{\text{100 pt}}$ Un fractal es un objeto geométrico que repite su estructura básica a difierentes escalas. Una forma de generar fractales es usando el método de Newton-Raphson para hallar las raíces de una función $f(z)$. En este taller vamos a generar un fractal de éstos usando la función $f(z) = z^5 + z^2$.
Para generar el fractal se debe aplicar el método de Newton-Raphson para hallar las raíces de $f(z)$ utilizando como punto de partida, z_0, los números en el plano complejo.

   Método de Newton-Raphson:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}.$$

   La idea de este taller es producir el siguiente fractal:



   (a) $\boxed{\text{18 pt}}$ Escriba dos funciones, una que retorne el valor de la función $z^5 + z^2$ y otra que retorne el valor de su derivada.

(b) $\boxed{18 \text{ pt}}$ Escriba una función que implemente el método de Newton-Raphson para un punto inicial z_0 y que haga 50 iteraciones. Esta función debe usar las dos funciones del punto anterior. Tenga en cuenta para esta función que no se puede dividir entre $0.0 + 0.0j$.

(c) $\boxed{18 \text{ pt}}$ Utilice la función `np.vectorize(fun)` para generar una nueva función que sea capaz de implementar el método generalizado de Newton-Raphson sobre un arreglo. Procese el arreglo hecho en la parte anterior con la función vectorizada que implementa el método de Newton-Raphson generalizado. Tenga en cuenta que este proceso puede tardar ya que el algoritmo se aplica sobre 4×10^6 puntos. (**Opcional:** Si no utiliza la función `np.vectorize(fun)` puede hacer el procesamiento mediante ciclos. Tenga en cuenta que no usar esta función puede hacer más rápido el código y que se dará un bono de **10 puntos** al código más rápido de la sección.)

(d) $\boxed{18 \text{ pt}}$ Haga un arreglo bidimensional de tamaño 200×200, en donde cada elemento del arreglo sea un número complejo de la región cuadrada del plano complejo que va desde -1 a 1 en la parte real y de $-j$ a j en la parte imaginaria. El elemento $[0, 0]$ de este arreglo debe ser el número complejo $-1.0 + 1.0j$ y el elemento $[199, 199]$ debe ser el número complejo $1.0 - 1.0j$. Esto para que la orientación de los números en el arreglo sean como en el plano complejo usual.

(e) $\boxed{10 \text{ pt}}$ Finalmente, aplíquele la función `np.angle(num)` al arreglo que se generó después de aplicar el método de Newton-Raphson, visualice el resultado con `imshow()` y exporte la imagen a un archivo de nombre "fractal_lr.png".

(f) $\boxed{18 \text{ pt}}$ Repita los pasos d y e pero, ahora que sabe que el código funciona, haga un arreglo de 2000×2000. El procesamiento de este arreglo puede tomar más de un minuto. Exporte la imagen de este punto a un archivo de nombre "fractal_hr.png".

(g) $\boxed{15 \text{ pt}}$ **BONO** Para darse cuenta de que esta figura tiene estructura fractal, genere la misma figura, pero en la región cuadrada del plano complejo que vaya desde -0.8 hasta -0.7 en la parte real y desde $-0.05j$ hasta $0.05j$ en la parte compleja. El arreglo bidimensional de los puntos en el plano debe ser de tamaño 2000×2000 también. (Hacer esto es equivalente a hacer un zoom en esa pequeña región del plano complejo.)
Debe generarse la siguiente imagen: