

Informe Tarea 4

Alumno: Lucas Suárez

Profesor: Elwin Van't Wout

1. El conjunto de Mandelbrot está contenido dentro de la circunferencia de radio 2. Por lo tanto, los puntos c se escogerán dentro del cuadrado de lado 4 con centro en el origen $(0, 0)$. Un punto c está pertenece al conjunto de Mandelbrot si y sólo si la sucesión $z_{n+1} = z_n^2 + c$ y $z_0 = 0$ cumple que $\|z_n\| \leq 2$ para todo $n \geq 0$. Evidentemente, como no podemos verificar computacionalmente esa propiedad para todo n , tenemos que utilizar alguna cota. En el caso de la implementación que realicé se consideró 10000 iteraciones.
2. En pyOpenCL los números se pueden generar aleatoriamente usando `np.random.uniform`. En C++ generar los números paralelamente puede causar que se generen números iguales en cada hilo debido a que la generación de estos números no es realmente aleatoria y depende del momento en que se ejecute la instrucción.
3. Para programar el algoritmo lo que se hizo fue calcular en cada hilo la iteración z_n , si en alguna iteración $\|z_n\| > 2$ entonces el valor de c asociado a ese hilo no pertenece al conjunto de Mandelbrot, por lo tanto se almacenaba `device_labels[global_id] = 1`. Así `device_labels` es un vector de dimensión igual a K con entrada igual a 1 si el i -ésimo vector n está en el Mandelbrot y 0 en otro caso.

Luego, para hacer la reducción, si `local_id == 0` el hilo 0 de ese grupo sumaba la cantidad de 1's asociado a su grupo. Es decir, el hilo 0 de cada grupo se encargaba de calcular la cantidad de puntos que NO pertenecían al Mandelbrot. Esto se almacena en el arreglo `device_group_sums` que tiene dimensión igual a la cantidad de grupos.

Finalmente, ese arreglo se copia al host en el numpy array `host_group_sums`, se hace `np.sum` y se obtiene la cantidad de puntos que no pertenecen al Mandelbrot, después es directo sacar el complemento para obtener la proporción de puntos dentro del fractal. Para calcular la superficie se multiplica la proporción obtenida antes por el área del rectángulo de donde se escogen los puntos, es decir, la proporción por 16.

El único if-else utilizado en cada iteración es el que verifica si es que $\|z_n\| > 2$, en cuyo caso se terminan las iteraciones pues se alcanza la condición de término del while-loop. La eficiencia no es muy alta porque tiene que utilizar variables auxiliares (u y v) para realizar esta tarea.