Version 13/11/19

UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

Arquitectura de Computadores

Trabajo OpenMP
Conjunto de Mandelbrot

Serafin.Benito@uclm.es

UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

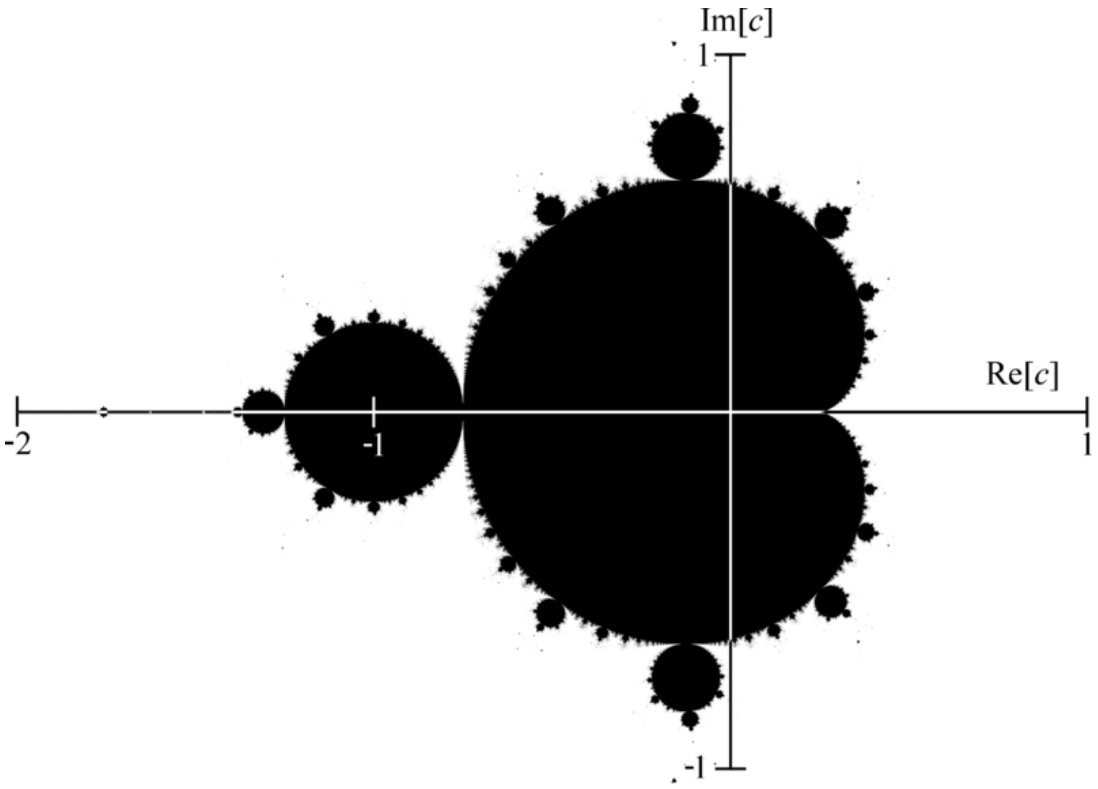
Contenido

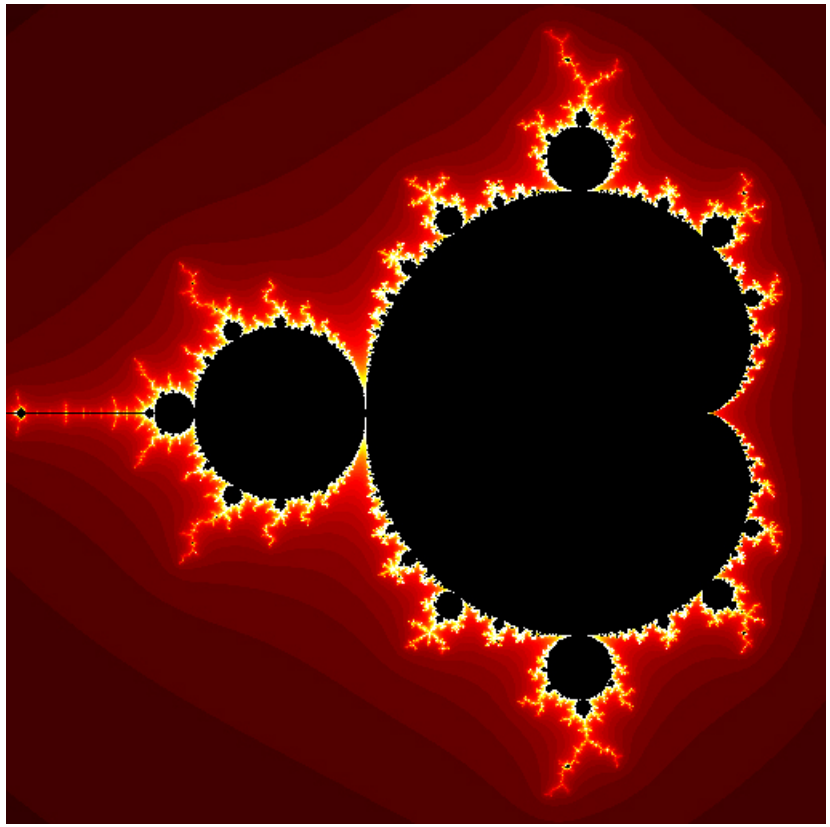
- Conjunto de Mandelbrot
- Programando Mandelbrot
- Prueba del programa
- Paralelización con OpenMP
- Tiempos de ejecución
- Entrega

Conjunto de Mandelbrot

Es el más conocido de los fractales

Se representa en el plano complejo: al número $x+i \cdot y$ corresponde el punto de coordenadas (x,y)





Definición del conjunto

- Llamemos M al conjunto de Mandelbrot
- Dado un número complejo *c*, se construye la sucesión definida por:

$$z_0 = 0$$

$$z_{n+1} = z_n{}^2 + c$$

- $c \in M$ si y solo si la sucesión está acotada. Ejs.:
  - $c = i \quad \rightarrow \quad 0, i, i-1, -i, i-1, -i \ldots$ acotada
  - $c = -1 \quad \rightarrow \quad 0, -1, 0, -1 \ldots$ acotada
  - $c = 1 \quad \rightarrow \quad 0, 1, 2, 5, 26, 677 \ldots$ no acotada
- Así pues, $i \in M$, $-1 \in M$ y $1 \notin M$

Programando Mandelbrot

- No se conoce un método tal que, $\forall c$, nos diga si $c \in M$ o si $c \notin M$
  - Las representaciones del conjunto de Mandelbrot son todas aproximadas
- Se sabe que $c \in M \quad ssi \quad \forall n \in \mathbb{N} \quad |z_n| \leq 2$
- El algoritmo, a partir de *c*, va obteniendo los términos de su sucesión hasta un máximo de términos:
  - Si encuentra un término $z_n \, tal \, que \, |z_n| > 2, \quad c \notin M$
  - Si no, considera $c \in M$ aunque podría no ser (quizá continuando la sucesión haya un término con módulo >2)

Archivo de entrada

- Ejecución:

  <archivo ejecutable> <archivo de entrada>

- El archivo de entrada es un archivo de texto plano con el siguiente formato:
  - 1.ª línea: número de imágenes que quiero generar
  - Una línea adicional por cada imagen. Formatos de línea posibles:
    - 1 <menor abscisa> <mayor abscisa> <menor ordenada> <mayor ordenada> <nombre archivo imagen>
    - 2 <abscisa centro del cuadrado > <ordenada del centro> <lado del cuadrado> <nombre archivo imagen>
    - El nombre del archivo imagen no debe tener más de 15 caracteres ni punto

Programa secuencial

A. Abrir archivo de entrada y leer número de imágenes
B. Para cada imagen que queremos obtener
   B1. Leer tipo de imagen (1 = rectangular, 2 = cuadrada)
   B2. Se calcula, en pixels, la anchura y altura de la imagen
   B3. Se abre y configura el archivo de imagen `ppm`
   B4. Por cada pixel se calcula el punto (x,y) y se llama a la función **mandel_val** que determina si puede pertenecer o no al conjunto de Mandelbrot
   B5. Se calcula el color del pixel correspondiente al número $x + y \cdot i$ y se imprime en el archivo de imagen
   B6. Se imprime el tiempo empleado en obtener la imagen

Función mandel_val

- Lista de parámetros: (x,y,max_iter)
- A partir de $c = x + i \cdot y$, va obteniendo los términos de su sucesión (p_real + i·p_imag) hasta un máximo de max_iter-1 términos:
  - Si encuentra un término cuyo módulo es >2, sale del bucle y devuelve el índice, j, de ese término en la sucesión
    - El color del pixel dependerá del número de iteraciones j
  - Si no, devuelve -1 (el número $x + i \cdot y$ se dibujará en blanco, como perteneciente al conjunto de Mandelbrot)

Prueba del programa

- Cread archivo de entrada. Por ejemplo, con el siguiente contenido:

```
2
2 -0.2 .8 .05 cuadrado
1 -2 1 -1 1 rectangulo
```

  Se crearán 2 imágenes (1.ª línea):
  - La primera cuadrada (2 inicial), en el archivo cuadrado.ppm con centro en el punto (-0.2, 0.8) y lado del cuadrado 0.05
  - La segunda rectangular (1 inicial), en el archivo rectangulo.ppm, desde la abscisa -2 a la abscisa 1 y desde la ordenada -1 a la ordenada 1
- Compilad y ejecutad el programa como se indica en los comentarios iniciales del propio programa
  - Jugad a cambiar el archivo de entrada probando a explorar zonas concretas
    - Cuanto más pequeña es el área explorada mayor ***zoom***
    - Las zonas más interesantes son las de los bordes

Paralelización con OpenMP

- Elaborad el programa **`mandelbrot_paralelo.c`** resultado de paralelizar con OpenMP el programa secuencial `mandelbrot_secuencial.c`
- Los cálculos de las sucesiones correspondientes a los números complejos de la zona por explorar son independientes →

  → se pueden **realizar en paralelo**

Paralelización con OpenMP

- Al paralelizar, el orden en que se generan los resultados es diferente al de la ejecución secuencial
- Pero **los píxeles deben imprimirse en** el archivo de imagen **en** el mismo **orden** en que lo hacía el algoritmo **secuencial**
- Sugerencia: guardar los resultados generados en una matriz y, cuando esté completa, tomar los resultados de la matriz para imprimir los píxeles
  - Los puntos B4 y B5 del algoritmo, ahora se deben hacer por separado

Tiempos de ejecución

- Usando el mismo archivo de entrada ejecutad el programa secuencial y el paralelo (sin usar la cláusula `schedule`)
- Comprobad que producen las mismas imágenes
- Comparad los tiempos para cada imagen
- Probad a usar la cláusula `schedule` como se indica en la diapositiva siguiente
- Dejad en el programa paralelo la cláusula `schedule` que dé mejores tiempos

Cláusula schedule

- Estudiad el efecto de usar la cláusula schedule de la directiva for:
  - Medid el tiempo de generar cada imagen usando unas veces la cláusula schedule(static,tt) y, otras, la cláusula schedule(dynamic,tt) probando con diferentes valores de tt
  - Comparad los tiempos para cada imagen con los del programa paralelo sin usar la cláusula schedule y con los del programa secuencial
  - Incluid un **comentario al final del código con los resultados obtenidos más relevantes y una explicación razonable de los mismos**
    - Especificad: modelo procesador usado y número de núcleos

Entrega

Grupos de 2 o 1 estudiante.

Comentario con nombres y apellidos de los estudiantes

Un estudiante por grupo subirá al espacio virtual de la asignatura el archivo **`mandelbrot_paralelo.c`**
antes de tres semanas
a contar desde el comienzo de la clase

Y recordad que ¡**copiar es una mala idea!**
Cruzaremos todas vuestras entregas.