

Progetto di programmazione avanzata e parallela

Parte 1 di 2 - Linguaggio C

Lo scopo di questo progetto è la realizzazione di un programma in grado di disegnare il frattale di Mandelbrot come una immagine a colori in formato netpbm a scala di grigi.

Definizione dell'insieme di Mandelbrot

Dato $c \in \mathbb{C}$, sia $\{z_n\}_{n \in \mathbb{N}}$ la successione di numeri complessi definita come:

$$\begin{cases} z_0 = 0 \\ z_n = z_{n-1}^2 + c \quad \text{per } n > 0 \end{cases}$$

L'*insieme di Mandelbrot* è l'insieme di tutti i punti nel piano complesso (i.e., tutti i $c \in \mathbb{C}$) tali per cui la successione non va all'infinito (i.e., è limitata). Visualizzando i punti contenuti e non nell'insieme usando colori diversi si ottiene la ben nota figura frattale.

Dato che non è possibile stabilire facilmente in ogni caso se la successione $\{z_n\}_{n \in \mathbb{N}}$ rimane limitata per un certo $c \in \mathbb{C}$ fissato, si esegue invece questa procedura:

- Si sceglie un raggio r tale per cui se un elemento della successione supera r in modulo si considera la successione non limitata. Scegliremo sempre 2 in quanto si è sicuri che se la successione supera 2 questa divergerà all'infinito.
- Si sceglie un numero massimo di iterazioni M .
- A partire da un punto $c \in \mathbb{C}$ si itera la funzione $f(z) = z^2 + c$ a partire da c (i.e., si calcola $f(c), f(f(c)), \dots$). Se $|f^n(c)| \geq r$ allora si considera il punto c esterno all'insieme di Mandelbrot e considera il minimo n tale per cui $|f^n(c)| \geq r$ come una indicazione della "velocità" con cui c si diverge.
- Se si compiono M iterazioni e $f^M(c)$ non ha mai superato il raggio r come modulo allora si considera il punto c interno all'insieme di Mandelbrot.

Funzionalità del programma

- Il programma deve accettare *dalla linea di comando* come argomenti (in questo ordine):
 - Il nome del file da creare;
 - Il numero massimo di iterazioni M da compiere;
 - La risoluzione verticale n_{rows} dell'immagine. La risoluzione orizzontale n_{cols} sarà sempre $1.5 \times n_{\text{rows}}$.
- Il programma dovrà calcolare l'insieme dei punti appartenenti all'insieme di Mandelbrot (usando M iterazioni e raggio $r = 2$) nella parte del piano compresa tra -2 e 1 per la parte reale e $-i$ e i per la parte complessa. Per fare questo userà una griglia equispaziata di punti di n_{rows} righe e n_{cols} colonne. Per ogni punto $c \in \mathbb{C}$ che non appartengono all'insieme di

Mandelbrot sarà necessario memorizzare il minimo n per il quale $|f^n(c)| \geq r$.

- Nel file indicato il programma deve salvare, in formato pgm binario (link alla documentazione: <https://netpbm.sourceforge.net/doc/pgm.html>) un'immagine dell'insieme di Mandelbrot in scala di grigi con le seguenti caratteristiche:
 - L'immagine deve avere altezza n_{rows} pixel e larghezza n_{cols} pixels.
 - I punti all'interno dell'insieme di Mandelbrot devono avere colore bianco (255).
 - I punti all'esterno devono avere colore $\lfloor 255 \times \frac{\log(n)}{\log(M)} \rfloor$ dove n è, per il punto c corrispondente a quel pixel, definito come il minimo valore per il quale $|f^n(c)| \geq r$.

Un esempio di output è dato in figura. In essa si ha un valore di $n_{\text{rows}} = 1000$ (e quindi $n_{\text{cols}} = 1500$) e $M = 1000$.

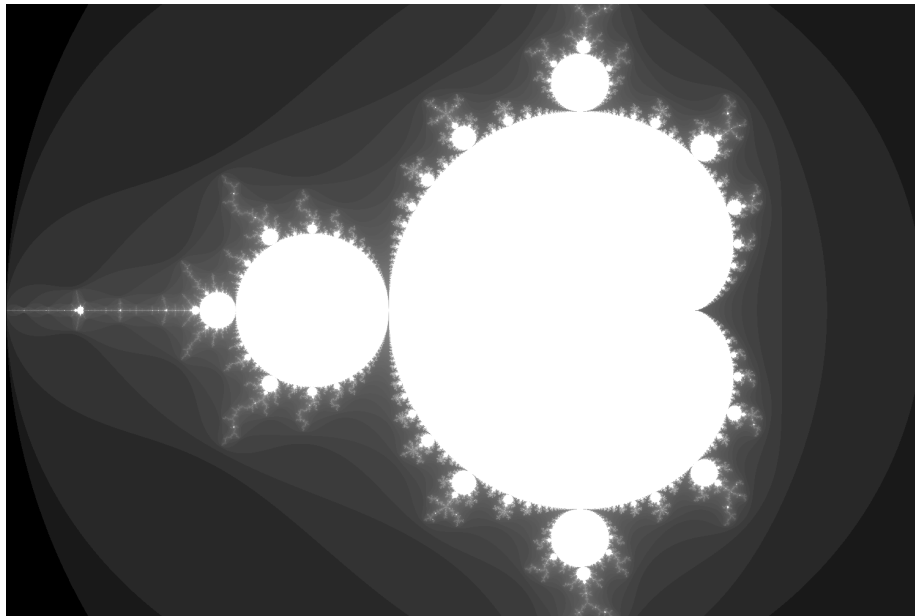


Figure 1: Frattale di Mandelbrot

Requisiti

- Il programma deve essere strutturato in più file, almeno un file `main.c`, due file `mandelbrot.h` e `mandelbrot.c` per quanto riguarda il calcolo del frattale e due file `pgm.h` e `pgm.c` per il salvataggio dell'immagine nel formato corretto.
- Il programma deve poter essere compilato tramite un makefile.

- È possibile utilizzare `complex.h` come libreria per gestire i numeri complessi oppure è possibile gestirsi una propria implementazione.
- Il calcolo dell'appartenenza dei punti all'insieme di Mandelbrot deve essere eseguito in parallelo con OpenMP.
- Il salvataggio dell'immagine deve usare `mmap`.
- Il programma deve correttamente rilasciare le risorse allocate.
- In caso di errore (e.g., file non apribile) il programma deve segnalare l'errore e uscire normalmente con un codice di errore diverso da 0.
- Ogni funzione rilevante (i.e., non di 2-3 righe) deve essere adeguatamente commentata spiegando che compito svolge, che input richiede e che output genera.
- Ogni file `.c` e `.h` deve contenere all'inizio come commento il proprio nome, cognome e numero di matricola.

Note importanti

- Questo progetto è relativo solamente alla parte in C del corso. Per sostenere la parte di progetto del corso è necessario consegnare sia la parte in C (parte 1) che la parte in Python (parte 2).
- Sebbene appaia ovvio, quando viene chiesto di inserire Nome, Cognome e Matricola, dovete inserire il vostro nome, cognome e numero di matricola a non, letteralmente, il testo "Nome", "Cognome" e "Matricola".
- Le istruzioni di consegna saranno fornite con la parte 2 del progetto.