`mandel`: Mandelbrot set

Secondo frammento LCS 2006/07

1 Introduzione

Il secondo frammento di progetto prevede la realizzazione di un programma C che calcola e disegna l'insieme di Mandelbrot [1] utilizzando più thread di esecuzione.

2 Definizione matematica

L'insieme di Mandelbrot é costituito dalla famiglia dei polinomi complessi $f_c : \mathcal{C} \to \mathcal{C}$

$$f_c(z) = z^2 + c \tag{1}$$

dove $c \in \mathcal{C}$ è un parametro complesso. Fissato c, si studia la sequenza

$$0, f_c(0), f_c(f_c(0)), f_c(f_c(f_c(0))) \dots \tag{2}$$

ovvero

$$0, f_c(0), f_c^2(0), f_c^3(0) \dots$$

ottenuta iterando f_c a partire da $z = 0$. Al variare di c, la sequenza può tendere all'infinito o rimanere confinata in un disco di raggio 2 del piano complesso centrato nell'origine.

L'insieme di Mandelbrot M é definito come l'insieme di tutti i punti c per cui la sequenza *non* tende all'infinito.

In Fig. 1, viene mostrata una visualizzazione dell'insieme (i punti in nero). M è un insieme compatto, e un punto c appartiene a M se e solo se $|f_c^n(0)| < 2$ per ogni $n \geq 0$. Se $|f_c^n(0)|$ supera 2 per qualche n, la sequenza diverge e c non appartiene a M.

3 Visualizzare Mandelbrot

L'algoritmo più semplice per visualizzare (una approssimazione de) l'insieme di Mandelbrot é l'*Escape Time Algorithm*. In questo algoritmo, dati A (l'area del piano complesso da visualizzare) ed r (una precisione fissata) si effettuano i seguenti passi:

1. si discretizza A in una griglia di punti a distanza uniforme (pixel)
2. per ogni punto (x, y)

[1] Una semplice forma frattale, vedi `http://en.wikipedia.org/wiki/Mandelbrot_set`

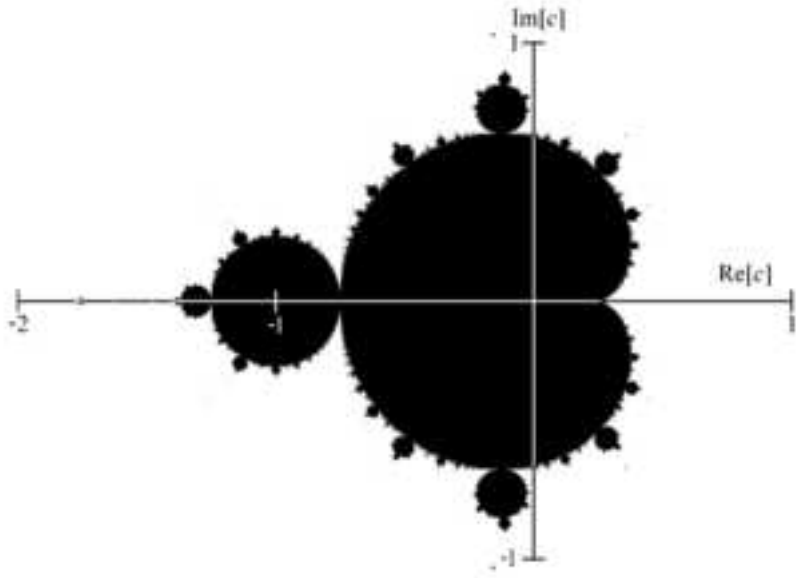

Figura 1: Una visualizzazione dell'insieme di Mandelbrot (i punti in nero).

(a) si calcolano r valori della sequenza $0, f_c(0), f_c^2(0), \ldots, f_c^{r-1}(0)$, dove $c = x + iy$

(b) se dopo r iterazioni $|f_c^r(0)| < 2$ si considera (x, y) appartenente a M e si assegna a (x, y) il colore nero.

(c) altrimenti, si assegna a (x, y) il colore j, dove j é il minimo per cui $|f_c^j(0)| \geq 2$.

Lo pseudocodice per il calcolo del colore di un singolo pixel è mostrato in Fig. 2. Nel codice, le coordinate del pixel (x, y) sono usate come valore iniziale per il calcolo. Il risultato di ogni iterazione è usato come punto d'inizio della successiva. Ad ogni iterazione si controlla se siamo usciti dal cerchio di raggio 2 (e quindi se l'insieme non appartiene a M), se questo è vero si assegna il numero di iterazioni come colore a (x, y) e si considera il prossimo punto, altrimenti si calcola la nuova iterazione. Quindi, rispetto a Eq. 1, $z = x + iy$, $c = x0 + iy0$, $z^2 = x^2 + i2xy - y^2$. Inoltre, r è `maxiteration`.

4 Visualizzare con più thread

Nel nostro esercizio utilizzeremo k thread (k parametrico) per calcolare in parallelo le varie righe di pixel dell'area di visualizzazione scelta.

Il costo della visualizzazione delle righe è molto variabile (dipende dal numero di iterazioni), useremo quindi un semplice schema con un produttore e più consumatori per bilanciare il carico fra tutti i thread attivi.

Il nostro programma (vedi Fig. 3) sarà costituito da un thread *main/dispatcher*, da k thread *worker* e da un thread *visualizer*.

Funzionamento del dispatcher:

1. calcola il numero di $N \times M$ pixel nell'area in base alle specifiche dell'utente;

2. inizializza la matrice `color[N][M]` dei colori;

```
Per ogni pixel:
{
  x = x0 = x co-ordinate of pixel
  y = y0 = y co-ordinate of pixel
  x2 = x*x
  y2 = y*y
  iteration = 0
  maxiteration = 1000

  while ( x2 + y2 < (2*2)
          AND  iteration < maxiteration )
  {
    y = 2*x*y + y0
    x = x2 - y2 + x0
    x2 = x*x
    y2 = y*y
    iteration = iteration + 1
  }

  if ( iteration == maxiteration )
    colour = black
  else
    colour = iteration
}
```

Figura 2: Pseudocodice per il calcolo del colore di un pixel.

3. inserisce gli indici di tutte le righe da calcolare $(0 \dots N-1)$ in una `coda` condivisa fra i vari thread *worker*
4. attiva k thread worker e un thread visualizer ne attende la terminazione
5. termina il processo

Ogni thread worker, esegue un ciclo in cui

- finchè la `coda` non è vuota
  1. estrae l'indice r di una nuova riga da calcolare dalla `coda`
  2. calcola il colore di ogni pixel (r, j) in r e lo riporta in `color[r][j]`

Infine il thread visualizer gestisce la visualizzazione dei colori (vedi Fig. 4).

5 Il comando `mandel`

Il visualizzatore parallelo può essere invocato da shell con il comando

```
mandel xm ym xa ya resolution [-t nthread]
```

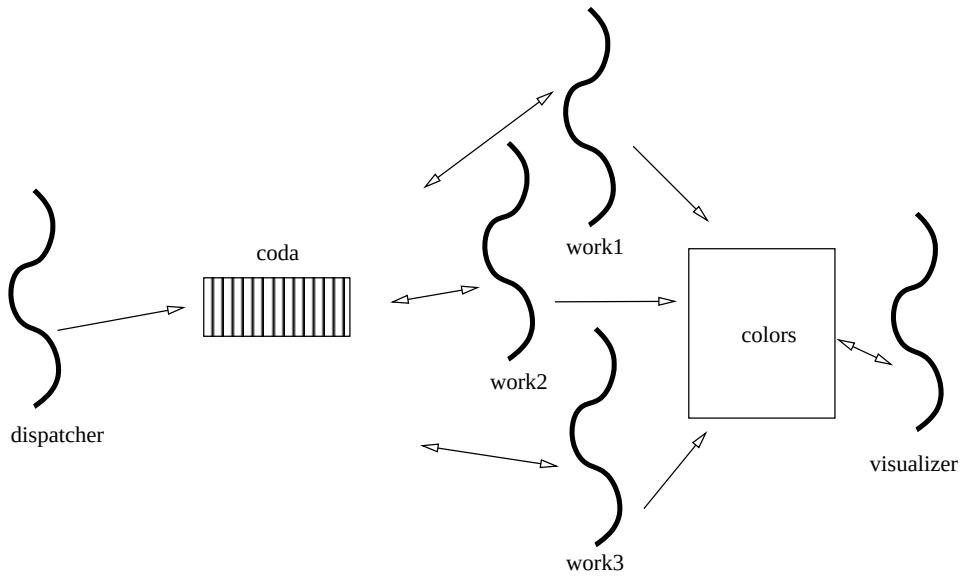

Figura 3: Visualizzazzione parallela con 3 thread worker.

```
 0  0  0  0  0  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  4  4  5  6 14  8 18  4  3  3  3  2  2  2  2  1  1  1  1  1  1  1  1
 0  0  0  0  0  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  4  5  7 10 11  7  5  4  3  3  3  2  2  2  2  1  1  1  1  1  1  1
 0  0  0  0  0  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  4  5  8 10 43 55  8  6  4  4  3  3  3  2  2  2  2  1  1  1  1  1  1
 0  0  0  0  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  5  5  7 11 89  *  *  16 10  5  4  4  3  3  3  2  2  2  2  1  1  1  1  1
 0  0  0  0  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  5  5  6  7 20  *  *  *  24  7  6  5  4  4  4  3  3  2  2  2  2  1  1  1  1
 0  0  0  0  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  4  6  7  7  7  8  9 11  *  *  *  17  9  8  6  5  5  7  4  3  2  2  2  2  2  1  1  1
 0  0  0  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  4  4  4  5  8 44 15 10 26 18  *  *  *  * 675 75 47 37  7  7 14  6  4  3  2  2  2  2  1  1  1
 0  0  1  2  2  2  2  2  2  2  2  2  2  2  3  3  4  4  4  4  4  5  6  8 19  *  23  *  *  *  *  *  *  *  *  *  27 12 92 26  9  4  3  2  2  2  2  2  1  1
 0  0  2  2  2  2  2  2  2  2  2  2  3  3  4  4  4  4  4  4  5  5  7  9 14 63  *  *  *  *  *  *  *  *  *  *  *  *  *  21  6  4  3  3  2  2  2  2  2  1
 0  1  2  2  2  2  2  2  2  2  3  3  4  5  4  4  4  4  5  5  5  7 38 56  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  10  6  5  4  3  2  2  2  2  2  1
 0  1  2  2  2  2  2  3  3  3  4  5  8  7  6  6  6  6  6  6  6  7 12 55  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *105  8  9  4  3  2  2  2  2  2  1
 0  2  2  2  2  3  3  3  3  4  4  5  9 10  9  8 20  8  7  7  7 10  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  10  4  3  2  2  2  2  2  2
 0  2  2  3  3  3  3  3  4  4  4  5  7 10 40 14 19 56 41 10  9 13  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  12  6  4  3  3  2  2  2  2
 0  2  3  3  3  3  3  4  4  4  5  6  8 11128  *  *  *  *  46 12 16  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  6  4  3  3  2  2  2  2
 0  3  3  3  3  3  4  4  4  5  7  8 10  *  *  *  *  *  *  *  38 33  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  12  5  4  3  3  2  2  2  2
 0  3  3  3  3  5  5  5  6  7 11 12 14  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  7  5  4  3  3  2  2  2  2
 0  4  5  6  8  6  7  7  8 10 14  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  10  6  4  4  3  3  2  2  2  2
 0  4  5  7 26  8  9  8  9 12 25  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  8  6  4  4  3  3  2  2  2  2
 0  3  4  3  4  5  5  5  6  7 10 17 17  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  6  4  4  3  3  2  2  2  2
 0  3  3  3  3  3  4  4  5  6 11  8 10 42  *  *  *  *  *  *  *  53  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  50  5  4  3  3  2  2  2  2
 0  2  3  3  3  3  3  4  4  4  5  6 11 13 98  *  *  *  *  *  *  14 20  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  30  6  4  3  3  2  2  2  2
 0  2  2  3  3  3  3  3  4  4  4  6  7 10 53 20 46 38 80 14 10 13  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  12  6  4  3  3  2  2  2  2
 0  2  2  2  3  3  3  3  3  4  4  5  7 19 13  9 13 11 24  8  8 10 32  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  45  7  4  3  3  2  2  2  2
 0  1  2  2  2  2  3  3  3  3  4  5  7  7  6  6  9  7  6  6  7  8 15  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  21 13  4  3  2  2  2  2  1
 0  1  2  2  2  2  2  2  2  3  3  4  6  5  5  5  4  5  5  5  6  7 52  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  14  7  5  4  3  2  2  2  2  1
 0  0  2  2  2  2  2  2  2  2  2  2  3  4  4  4  4  4  4  5  5  5  8 10 17  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  15  6  4  3  3  2  2  2  2  1
 0  0  1  2  2  2  2  2  2  2  2  2  2  3  3  3  4  4  4  4  5  5  6  8131  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  16  *  42 23  4  3  2  2  2  2  2  1  1
 0  0  1  2  2  2  2  2  2  2  2  2  2  2  3  3  3  4  4  4  5  5 10 26 20 11 94  *  *  *  *  *  *  *  *  *  45 14  9 23 11  8  4  3  2  2  2  2  2  1  1
 0  0  1  1  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  4  5  5 36 18  9  8 11 10 12 44  *  *  15 11  9  8  6  6  7  5  3  2  2  2  2  2  1  1  1
 0  0  0  1  1  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  5  5  6  6  7 12  *  *  *  *  8  6  5  5  4  4  3  3  2  2  2  2  2  1  1  1  1
 0  0  0  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  4  5  5  7 11  *  *  *  80  8  5  4  4  4  3  3  2  2  2  2  2  2  1  1  1  1
 0  0  0  0  0  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  4  4  4  5 11 12 43 49 10  7  4  4  3  3  3  3  2  2  2  2  1  1  1  1  1
 0  0  0  0  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  4  4  4  6  7 12 12  7  5  4  3  3  3  3  2  2  2  2  1  1  1  1  1  1
```

Figura 4: Esempio di visualizzazione intera, $*$ indica i punti appartenenti all'insieme di Mandelbrot.

dove (vedi Fig. 5)

- `(xm,ym)` è l'angolo in alto a sinistra

$$-2 \leq xm \leq 1, -1 \leq ym \leq 1$$

- `(xa,ya)` è l'angolo in baso a destra

$$-2 \leq xa \leq 1, -1 \leq ya \leq 1, xm < xa, ym > ya$$

- `resolution` è il numero di pixel in ciascuna riga (quelli sulla colonna sono ricavati in modo da avere un campionamento uniforme)
- `-t nthread` è il numero dei thread worker da attivare (opzionale default 3)

I parametri individuano univocamente l'area da visualizzare (in grigio in Fig. 5) e tale area viene campionata con $resolution_x$ =`resolution` pixel in orizzontale e

$$resolution_y = \frac{(ym - ya) * resolution_x}{(xa - xm)}$$

in verticale in modo da ottenere una griglia uniforme. Il numero di terazioni massimo (`maxiter`) è fissato a 1000. Inoltre, le ccordinate sull'asse x e sull"asse y sono campinate a distanza

$$d = \frac{xa - xm}{resolution_x} = \frac{ym - ya}{resolution_y}$$

I colori vengono stampati come interi sullo standard output (utilizzando 3 posizioni per ogni cifra) per i pixel che non appartengono all'insieme e `*` per quelli che appartengono all'insieme. Ad esempio, Fig. 4 mostra l'output atteso per

```
mandel -2 1 1 -1 50
```

6 Opzionale: altre visualizazzioni

È possibile implementare altri formati di visualizzazione, oltre a quello intero.

All'interno del kit è disponibile un script Perl/TK che visualizza un'immagine a colori leggendola dallo standard input. Il formato atteso e':

```
nrighe
ncolonne
0 1 #ffaagg
...
```

in cui `nrighe` ed `ncolonne` danno rispettivamente il numero della righe e delle colonne nel canvas creato dallo script e le linee successive (terminate da `\n`) forniscono coordinate e colore di tutti i punti (anche in ordine sparso, i punti che non compaiono nell'elenco manterranno il colore dello sfondo).

Altrimenti, è possibile utilizzare ASCII art[2] o qualsiasi altro formato.

Il comando `mandel` deve essere esteso opportunamente con opzioni di visualizzazione della forma `-visualization_name`

```
mandel xm ym xa ya resolution [-t nthread] [-color | -ascii | ...]
```

la visualizzazione di default deve rimanere quella intera.

[2]vedi: `http://en.wikipedia.org/wiki/ASCII_art`

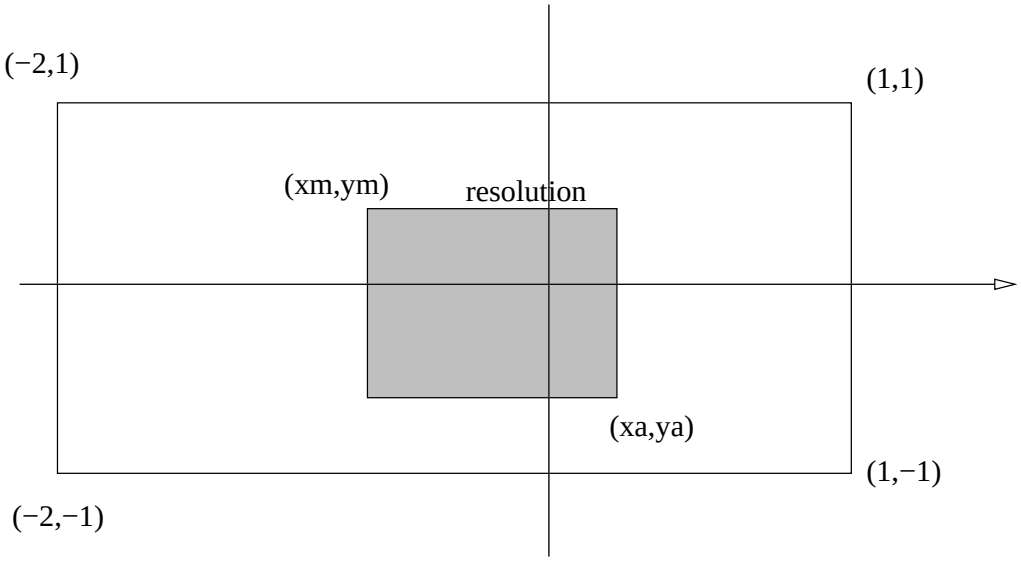

Figura 5: Significato dei parametri del comando `mandel`.

7 Opzionale: documentazione doxygen

La documentazione del codice prodotto può essere organizzata secondo il formato doxygen. In questo caso è richiesto di consegnare anche il file di configurazione ed estendere il Makefile con una opportuna regola (target `docu`).