Laboratorio computazionale numerico
Lezione 8

Federico Poloni <f.poloni@sns.it>

2009-12-09

1 Frattali di Newton

In questa lezione cercheremo di disegnare i frattali che si ottengono disegnando i bacini di attrazione del metodo di Newton (sul piano complesso) per un polinomio. Le immagini risultanti dovrebbero assomigliare a quella seguente. Notate

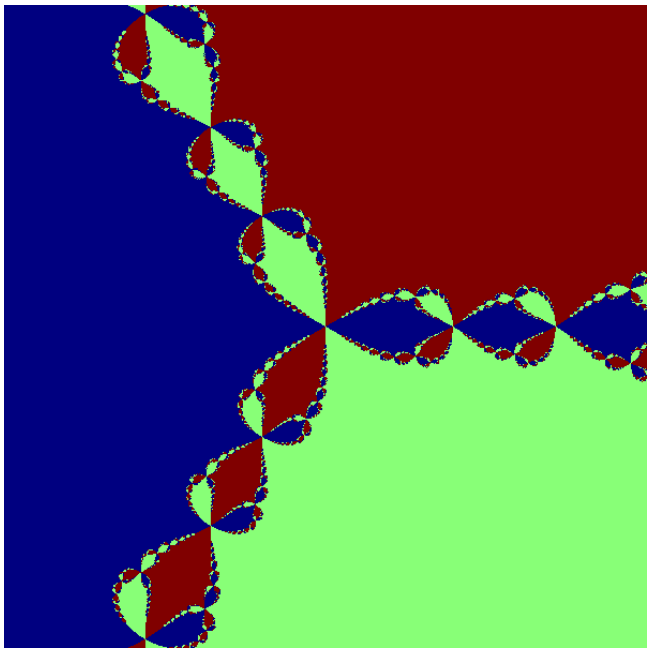

Figura 1: Disegno dei bacini di attrazione del metodo di Newton per il polinomio $x^3 + 1$. I tre colori diversi corrispondono ai punti del piano complesso a partire da cui Newton converge alle tre diverse radici.

la simmetria della figura: a seconda del quadrante del piano complesso da cui partiamo, si ha convergenza alla più vicina delle tre radici; però, nei punti che sono circa equidistanti da due delle tre radici, si ha un comportamento caotico.

Anche se sembra complicato, il problema può essere suddiviso in molti sottoproblemi ognuno abbastanza semplice.

1.1 Manipolazione di polinomi

Dato un polinomio, lo rappresentiamo come il vettore dei suoi coefficienti: ad esempio, a $x^3 - 1$ corrisponde il vettore [1;0;0;−1]. Notare che se il polinomio ha grado n, il vettore ha lunghezza $n + 1$.

Il metodo di Horner per valutare un polinomio corrisponde a fare i prodotti associandoli in questo modo: per esempio, per un polinomio di grado 4,

$$a(x) = (((a_4 * x + a_3) * x + a_2) * x + a_1) * x + a_0.$$

Esercizio 1. Scrivere una **function** y=horner(p,x) che prende un polinomio p (rappresentato come il vettore dei suoi coefficienti) e un numero x e calcola $p(x)$ con il metodo di Horner.

Esercizio 2. Scrivere una **function** dp=derivata(p) che prende un polinomio p (vettore di coefficienti) e restituisce la sua derivata (vettore di coefficienti).

1.2 Metodo di Newton

Il metodo di Newton è l'iterazione

$$x_{k+1} = x_k - \frac{p(x_k)}{p'(x(k)}.$$

Esercizio 3. Scrivere una **function** x=newton(p,x0) che esegue il metodo di Newton sul polinomio p partendo dal punto iniziale x0. Come criterio di arresto, si può usare quello di terminare se $|p(x)| \leq 10^{-12}$:

```
function x=newton(p,x0);
  x=x0;
  px=horner(p,x0);
  while(abs(px)>1E-12)
    %calcola il nuovo x e il nuovo px
  endwhile
endfunction
```

Testare il metodo di Newton sul polinomio $p(x) = x^3 + 1$. Quali sono le sue radici? Riuscite a trovare un valore iniziale per il metodo di Newton che lo faccia convergere ad ognuna di esse? Ricordate che il modo più semplice per inserire un numero complesso in Octave è 2+3∗I.

1.3 "Disegnare" una matrice

La funzione **imagesc** prende come parametro una matrice $m \times n$ A, e genera un'immagine $m \times n$ in cui il pixel i, j è colorato di un colore che varia su una scala da rosso a blu a seconda di quanto il valore di $A_{i,j}$ è grande/piccolo rispetto agli altri elementi della matrice. Per esempio, con

```
octave:1> imagesc(eye(100))
```

viene visualizzata un'immagine in cui la diagonale (elementi più grandi) è rossa, e tutti gli altri elementi (elementi più piccoli) sono blu. Provate anche **imagesc**(laplacian(10)) o **imagesc**(**rand**(100)).

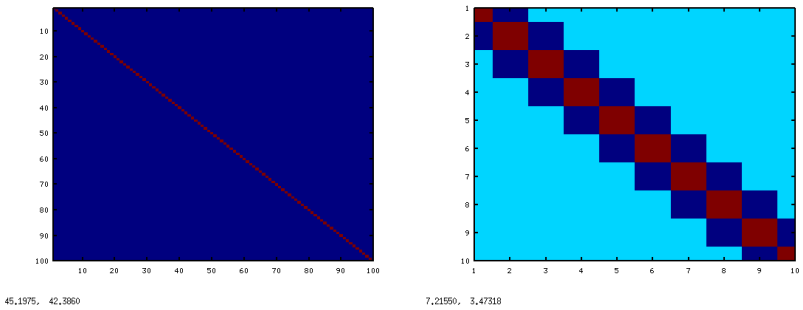

Figura 2: **imagesc**(**eye**(100)) e **imagesc**(laplacian(10))

1.4 Frattale di Newton

Per disegnare il frattale di Newton relativo al polinomio $p(x) = x^3 + 1$, abbiamo bisogno innanzitutto di una funzione che "decida" a quale valore c'è convergenza.

Esercizio 4. Scrivere una funzione **function** val=decidi(x) che restituisca 1, 2 o 3 a seconda se il numero complesso x è più vicino a -1, a $\frac{1}{2} + I\frac{\sqrt{3}}{2}$, o a $\frac{1}{2} - I\frac{\sqrt{3}}{2}$.

Esercizio 5. Scrivere una **function** img=newton() che non prende alcun argomento e restituisce una matrice 101×101 chiamata img calcolata in questo modo:

- genera 101 valori equispaziati nell'intervallo $[-2, 2]$ con l'istruzione range=−2:0.04:2.
- per ogni coppia (i, j):
  - calcola il punto z_0 del piano complesso z0=range(i)+I∗range(j);
  - esegue il metodo di Newton per il polinomio $x^3 + 1 = 0$ partendo dal punto z0;
  - utilizzando la funzione decidi(), scrive 1, 2 o 3 in img(i,j) a seconda della radice del polinomio a cui si ha convergenza a partire dal valore iniziale z0.[1].
- restituisce la matrice img

La funzione qui scritta sarà probabilmente abbastanza lenta (potrebbe metterci un mezzo minuto...) e restituirà una matrice img che potrete poi visualizzare a schermo con l'istruzione **imagesc**(img). Assomiglia alla figura 1?

Esercizio 6. Generate l'immagine corrispondente per il metodo di Newton su altri polinomi. Non sapete le radici? Potete farle calcolare a Octave: la funzione **roots**(p) calcola le radici di un polinomio (rappresentato come vettore di coefficienti): per esempio,

```
octave:2> roots([1 -1 -1 ])    %radici di x^2-x-1=0
ans =

  -0.61803
   1.61803
```

[1] potete dare per scontato che il metodo di Newton converga per tutti i valori iniziali nel nostro range.

Se volete, potete riscrivere le funzioni scritte finora in modo che il polinomio p non sia fissato ma sia uno degli argomenti.

1.5 Se vi state annoiando...

Il *frattale di Julia* relativo al numero complesso c è definito come l'insieme dei punti z_0 per cui la successione definita da $z_{k+1} = z_k^2 + c$ non diverge.

Potete generare un interessante diagramma del *frattale di Julia* relativo al numero complesso c con la seguente funzione **function** img=julia(c). La funzione:

- genera 101 valori equispaziati nell'intervallo $[-2, 2]$ con l'istruzione range=−2:0.04:2.
- per ogni coppia (i, j):
  - calcola il punto z_0 del piano complesso z0=range(i)+I∗range(j);
  - applica per 10 volte la funzione $f(z) = z^2 + c$ a partire dal punto z_0
  - scrive in img(i,j) l'*arcotangente* del modulo del numero complesso z_{10} così calcolato. Difatti i numeri hanno variazioni molto grosse (da 0 a 10^{300}...), e disegnarli così come sono non produrrebbe un risultato interessante.
- restituisce la matrice img

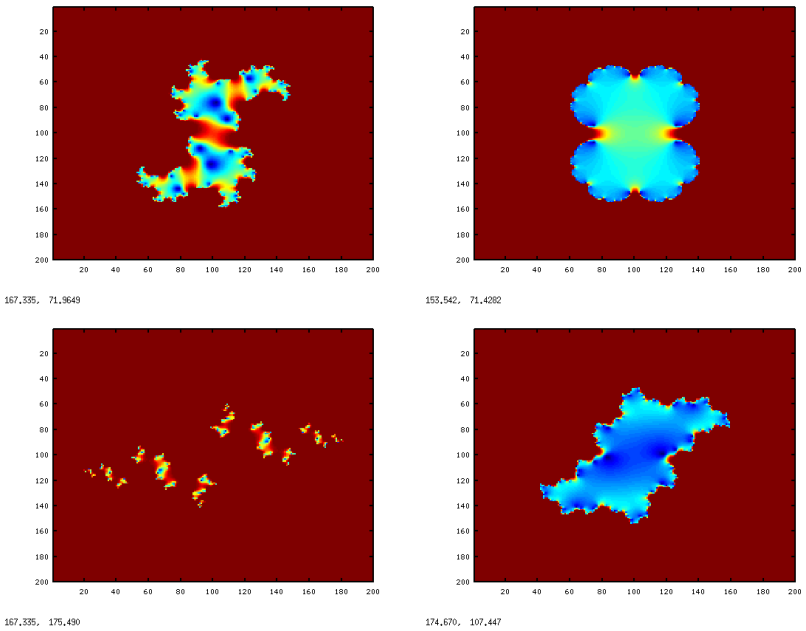

Figura 3: Alcuni frattali di Julia

1.6 Versione vettorizzata

Il programma newton.m è abbastanza lento; difatti non abbiamo dedicato alcuna attenzione all'ottimizzazione (sono questioni tecniche e noiose — non è lavoro per un matematico). Una versione più veloce, che potreste usare se volete generare delle immagini più grosse, per esempio da usare come sfondo, è pubblicata sul sito del corso.

Potete scaricare versioni più veloci dei programmi che disegnano il frattale di Newton e quello di Julia rispettivamente da `http://poisson.dm.unipi.it/%7epoloni/dida/lcn09/lezione%208/disegna_newton.m` e `http://poisson.dm.unipi.it/%7epoloni/dida/lcn09/lezione%208/julia.m`.