

ESERCITAZIONE 3

Sistemi di funzioni iterate e frattali

Per questa esercitazione verranno corretti gli **esercizi 3 e 7**. Create un file `.tar` o `.zip` contenente gli script che risolvono gli esercizi e le eventuali function ausiliarie, e caricatele sulla pagina di e-learning del corso.

1. Fern fractal

Il *fern fractal*, o *felce di Barnsley*, è un insieme frattale così chiamato perché prende la forma di una foglia di felce. Si ottiene disegnando una successione di punti del piano generati tramite un processo casuale che ora descriviamo.

Rappresenteremo un punto sul piano per mezzo del vettore $\mathbf{x} \in \mathbb{R}^2$ delle sue coordinate. La successione che definisce il *fern fractal* è generata applicando successivamente delle trasformazioni affini del tipo

$$\mathbf{x} \rightarrow A\mathbf{x} + \mathbf{b},$$

dove A è una matrice 2×2 e \mathbf{b} un vettore di dimensione 2. Nella costruzione del *fern fractal* si usano quattro diverse trasformazioni di questo tipo, ciascuna scelta con una certa probabilità. Cominciamo studiando la prima di queste trasformazioni, quella che viene scelta con maggiore probabilità. Essa è definita da

$$A_1 = \begin{bmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}.$$

L'effetto di questa trasformazione è di ruotare e "accorciare" il vettore \mathbf{x} , e poi aggiungere 1.6 alla seconda componente. L'applicazione ripetuta di questa trasformazione costruisce tante copie sempre più piccole delle foglie e del gambo della felce.

Esercizio 1 Scegliamo $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Scrivere uno script in Matlab che applichi 50 volte la trasformazione definita sopra e disegni sul piano tutti i punti ottenuti. (Suggerimento: anziché usare il comando `plot` ogni volta che si calcola un nuovo punto, è preferibile memorizzare dapprima tutti i punti calcolati in un array \mathbf{S} di dimensioni 2×50 , e poi disegnarli con il comando `plot(S(1,:), S(2,:), 'g.')`).

Le altre tre trasformazioni usate per il *fern fractal* sono definite da

$$\begin{aligned} A_2 &= \begin{bmatrix} 0.20 & -0.26 \\ 0.23 & 0.22 \end{bmatrix}, & \mathbf{b}_2 &= \begin{bmatrix} 0 \\ 1.6 \end{bmatrix} \\ A_3 &= \begin{bmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{bmatrix}, & \mathbf{b}_3 &= \begin{bmatrix} 0 \\ 0.44 \end{bmatrix} \\ A_4 &= \begin{bmatrix} 0 & 0 \\ 0 & 0.16 \end{bmatrix}, & \mathbf{b}_4 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned}$$

e hanno rispettivamente l'effetto di generare la foglia in basso a destra, la foglia in basso a sinistra, e il gambo della felce.

Esercizio 2 Ripetere l'Esercizio 1 per le altre tre trasformazioni appena definite.

Passiamo ora a costruire il *fern fractal*. Siano

$$p_1 = 0.85, \quad p_2 = 0.92, \quad p_3 = 0.99, \quad p_4 = 1.$$

Prendiamo $\mathbf{x}_1 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$ come punto iniziale. Per $k = 1, 2, \dots$,

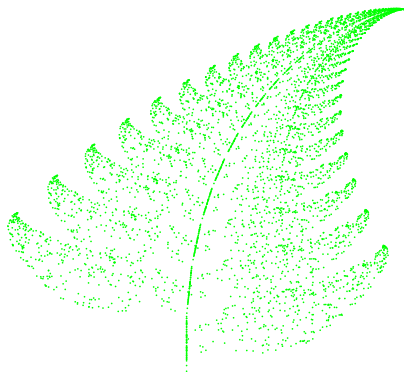
- scegliamo in modo casuale, con distribuzione uniforme, un numero reale r compreso tra 0 e 1;
- se $r \leq p_1$ poniamo $i = 1$, se $p_1 < r \leq p_2$ poniamo $i = 2$, se $p_2 < r \leq p_3$ poniamo $i = 3$, altrimenti si pone $i = 4$;
- definiamo $\mathbf{x}_{k+1} = A_i \mathbf{x}_k + \mathbf{b}_i$.

Esercizio 3 Scrivere una function `fern(m)` che prende in ingresso l'intero positivo m e disegna sul piano i punti rappresentati dai vettori $\{\mathbf{x}_k\}$ per $k = 1, \dots, m$.

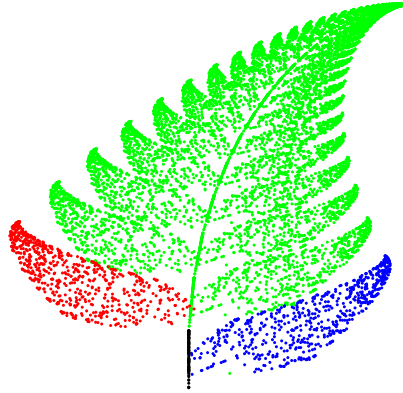
Suggerimenti:

- si potrà usare la funzione `rand` per generare il numero casuale r ,
- è consigliabile, come suggerito in precedenza, memorizzare tutti gli elementi della successione in un array e visualizzarli alla fine con un unico comando `plot`,
- si potrà usare il comando `axis off` per rimuovere gli assi cartesiani dalla figura.

Qui sotto è raffigurata l'immagine ottenuta per $m = 5000$.



Esercizio 4 Per capire meglio come si crea il fern fractal, modificate la function `fern` per creare una nuova function `ferncolor(m)` che disegna i punti calcolati in colori diversi a seconda delle trasformazioni usate (per esempio, i verde i punti ottenuti dalla prima trasformazione, in rosso quelli ottenuti dalla seconda, in blu quelli ottenuti dalla terza, in nero quelli ottenuti dalla quarta). Dovreste ottenere una figura simile a questa:



Esercizio 5 Che cosa succede modificando gli elementi di A_i e \mathbf{b}_i ?

Provate ad usare i parametri suggeriti alla pagina

<https://www.dcnicholls.com/byzantium/ferns/fractal.html>
per disegnare felci diverse.

1 bis. Curva di Lévy (facoltativo)

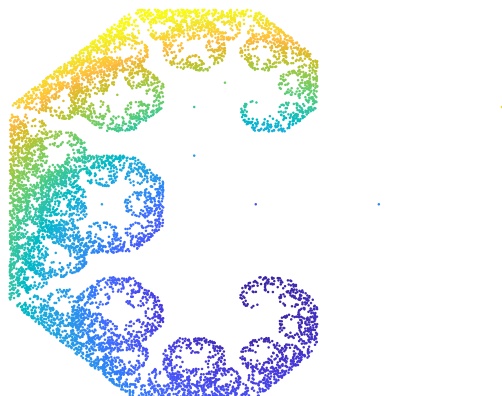
La curva di Lévy (cercatela su Wikipedia) si può costruire usando le seguenti trasformazioni complesse:

$$f_1(z) = \frac{(1-i)z}{2},$$

$$f_2(z) = 1 + \frac{(1+i)(z-1)}{2}.$$

In questo caso rappresentiamo i punti del piano come numeri complessi z . La scelta tra f_1 ed f_2 ad ogni passo dell'iterazione avviene in modo casuale, con uguale probabilità.

Esercizio 6 Scrivere una function `levy(m)` che prenda in ingresso un intero positivo m e disegni la curva di Lévy ottenuta effettuando m iterazioni. Potete usare il comando `scatter` per disegnare i punti usando un'opportuna colormap e ottenere un'immagine simile a questa:



2. Triangolo di Sierpinski

In questa sezione generalizziamo l'iterazione che genera il *fern fractal* per costruire il triangolo di Sierpinski.

Sia n un intero maggiore di 1, siano A_1, \dots, A_n matrici reali 2×2 assegnate e siano $\mathbf{b}_1, \dots, \mathbf{b}_n$ vettori reali di dimensione 2 assegnati. Consideriamo la matrice C , di dimensioni $2n \times 3$, così definita:

$$C = \begin{bmatrix} A_1 & \mathbf{b}_1 \\ A_2 & \mathbf{b}_2 \\ \vdots & \vdots \\ A_n & \mathbf{b}_n \end{bmatrix}.$$

Esercizio 7 Scrivere una function `fract(C,m)` che disegni i punti sul piano generati nel modo seguente:

- si costruisce il vettore `dets` di dimensione n , tale che la j -esima componente di `dets` contenga $|\det(A_j)|$ (suggerimento: usare il comando `det`),
- si modifica il vettore `dets` come segue:
`dets = max(dets, max(dets)/(25*n));`
`dets=dets/sum(dets);`
- si crea il vettore `p` di dimensione n con componenti
`0, dets(1), dets(1)+dets(2), \dots, dets(1)+\dots+dets(n-1)`
- si costruisce una successione di vettori $\{\mathbf{x}_k\}_{k=1,2,\dots}$ in questo modo: si parte da \mathbf{x}_1 scelto in modo casuale, dopodiché per ciascun $k = 1, 2, \dots$ si sceglie un numero casuale r con distribuzione di probabilità uniforme in $[0, 1]$, si pone $i = \text{sum}(\mathbf{p} < \mathbf{r})$ e si definisce

$$\mathbf{x}_{k+1} = A_i \mathbf{x}_k + \mathbf{b}_i,$$

- per $k = 21, 22, \dots, 20 + m$ si disegna sul piano il punto le cui coordinate sono gli elementi del vettore \mathbf{x}_k .

Scrivere quindi uno script che applichi la function `fract(C,m)` alle matrici seguenti (con opportune scelte di `m`):

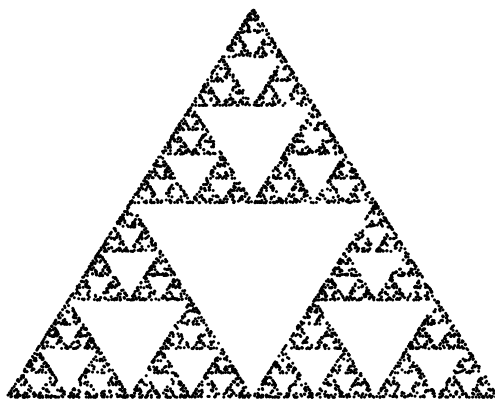
$$S = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 \\ 0.5 & 0 & 0.25 \\ 0 & 0.5 & 0.5 \end{bmatrix},$$

e

$$K = \begin{bmatrix} t & 0 & 0 \\ 0 & t & 0 \\ s & -x & t \\ x & s & 0 \\ s & x & 0.5 \\ -x & s & x \\ t & 0 & t_2 \\ 0 & t & 0 \end{bmatrix},$$

dove $x = \frac{1}{3} \sin(\pi/3)$, $s = 1/6$, $t = 1/3$, $t_2 = 2/3$.

La function `fract(C,m)` applicata alla matrice S disegna il triangolo di Sierpinski (figura qui sotto). Che cosa si ottiene usando la matrice K ?



3. The Chaos Game (facoltativo)

Consideriamo tre punti nel piano (uno rosso, uno verde, uno blu) posti ai vertici di un triangolo equilatero. Poi scegliamo un quarto punto (nero) in una posizione qualsiasi.

Il gioco consiste nel far muovere il punto nero secondo le regole seguenti. Ad ogni turno, lanciamo un dado a sei facce. Se il risultato è 1 o 2, muoviamo il punto nero in direzione del punto rosso, spostandolo di una lunghezza pari a metà della distanza che separava i due punti all'inizio del turno. Se il risultato è 3 o 4, muoviamo il punto nero verso il punto verde, percorrendo sempre metà della distanza tra i due punti. Se il risultato è 5 o 6, muoviamo il punto nero verso il punto blu, sempre rispettando la regola di coprire metà della distanza.

Esercizio 8 Scrivere una function `chaos(m)` che simuli il gioco descritto. Come al solito, m è un intero positivo. La function dovrà disegnare nel piano i punti generati dal gioco dal turno 21 al turno $20 + m$. Che cosa appare?

Che cosa succede se si colorano i punti a seconda della mossa che li ha generati (cioè si disegnano in rosso i punti generati da una mossa verso il vertice rosso, in verde i punti generati da una mossa verso il vertice verde, in blu i punti generati da una mossa verso il vertice blu)?

Che cosa succede se si modificano i vertici del triangolo (che quindi non sarà più necessariamente equilatero)?

Esercizio 9 Implementare una variante del gioco in cui si parte dai vertici di un esagono, anziché un triangolo, e ciascuno dei sei possibili esiti del lancio del dado muove il punto in direzione di un vertice diverso. In questa variante, ad ogni turno di gioco, il punto viene spostato di $2/3$ della distanza dal vertice prescelto.

Osservazione. In questa esercitazione abbiamo creato dei frattali a partire da trasformazioni lineari o affini. Questo procedimento si può estendere anche a trasformazioni non lineari: è il caso, per esempio, delle *fractal flames* proposte da Scott Draves. Trovate informazioni in materia anche su Wikipedia: https://en.wikipedia.org/wiki/Fractal_flame e una descrizione dettagliata è disponibile qui: <https://flam3.com/flame-draves.pdf>