Laboratorio Didattico di Matematica Computazionale

a.a. 2018-2019

Esercitazione 3

Dinamica complessa e frattali

1. Immagini in Matlab

In questa sezione vedremo come creare e rappresentare immagini in Matlab.

- Cominciamo con le immagini in bianco e nero. Provate a dare il comando seguente:

  `A=imread('cameraman.tif');`

  Che tipo di variabile è `A`? (Controllate la descrizione nella finestra Workspace, oppure usate il comando `whos A`)

  Visualizziamo l'immagine rappresentata da `A` con il comando

  `imshow(A)`

  In alternativa possiamo scrivere

  `imagesc(A)`

  `colormap gray`

  Che legame c'è tra la matrice `A` e l'immagine del cameraman?

- Gli elementi di A sono interi a 8 bit, ma un'immagine b/n si può anche rappresentare come una matrice di `double` compresi tra 0 e 1:

  `B=double(A)/255;`

  `imshow(B)`

- Definite le matrici seguenti e visualizzate le immagini associate. Che cosa è cambiato rispetto all'immagine di partenza e perché?

  `C=1-B;`

  `D=sqrt(B);`

  `E=B.^2;`

- Per salvare una matrice come immagine:

  `imwrite(C,'negativecameraman.tif')`

  Naturalmente si possono usare anche altri formati, quali PNG e JPEG; trovate le opzioni possibili nella pagina di help.

- Passiamo alle immagini a colori. Matlab rappresenta un'immagine a colori come un array di dimensione $m \times n \times 3$. Nel formato RGB, ciascuna "fetta" $m \times n$ corrisponde ai livelli di intensità di un colore primario (nell'ordine, rosso, verde e blu).

  Che cosa succede se diamo i comandi seguenti?

  `clear all`

```
A=imread('peppers.png');
imshow(A)
figure(2)
imshow(A(:,:,1))
figure(3)
imshow(A(:,:,2))
figure(4)
imshow(A(:,:,3))
```

Esercizio 1 *Scrivere uno script in Matlab che crei e rappresenti un'immagine a sfondo nero contenente un cerchio rosso, un rettangolo giallo e un quadrato verde.*

2. Segno di un numero complesso

Sia z un numero complesso, non immaginario puro. Si definisce

$$\mathrm{sign}(z) = \begin{cases} 1 & \text{se} \quad \Re(z) > 0 \\ -1 & \text{se} \quad \Re(z) < 0 \end{cases}$$

Per esempio, verifichiamo con Matlab che il segno di $z = -2 + 4i$ è -1:

```
z=-2+4*i;
s=sign(real(z))
```

Si dimostra che il segno di un numero complesso z è il limite della successione

$$x_1 = z,$$
$$x_{n+1} = \frac{x_n + x_n^{-1}}{2}, \qquad n \geq 1.$$

Esercizio 2 *Scrivere una function* `[s,m] = segno(z, maxiter, epsilon)` *che*

- *prende in input il numero complesso* `z`*, l'intero positivo* `maxiter` *e il numero reale positivo* `epsilon`*;*
- *calcoli gli elementi della successione* $\{x_n\}_{n\geq 1}$ *per* n *che va da* 1 *a* m *tale che* $m =$`maxiter` *oppure* $|x_m - x_{m-1}| <$`epsilon`*;*
- *restituisca in output* `s`*, che è l'ultimo elemento calcolato della successione, e* `m`*, il numero di elementi calcolati.*

Scrivere poi uno script che usi la function `segno` *per rispondere alle domande seguenti:*

- *Si scelga un numero complesso con parte reale piccola, ad esempio dell'ordine di* 10^{-2}*. Quante iterazioni servono perchè il metodo converga con* `epsilon`$= 10^{-12}$*?*
- *Che cosa succede se si applica l'iterazione ad un numero immaginario puro?*

Vogliamo ora disegnare i bacini di attrazione dell'iterazione in un rettangolo del piano complesso $[a, b] \times i[c, d]$, dove $[a, b]$ e $[c, d]$ sono intervalli della retta reale. Per far questo costruiamo un reticolo nel rettangolo, applichiamo l'iterazione a ciascun punto del reticolo e contiamo quante iterazioni sono necessarie per avere convergenza.

Esercizio 3 *Scrivere una function* `[x, y, iter] = bsegno(a, b, c, d, maxiter, epsilon)` *che:*

- *prenda in input gli estremi* `a`, `b`, `c`, `d` *degli intervalli* $[a, b]$ *e* $[c, d]$, *il numero massimo di iterazioni* `maxiter` *e il numero reale positivo* `epsilon`*;*
- *suddivida* $[a, b]$ *e* $[c, d]$ *in sottointervalli piccoli (ma non troppo!), per esempio* `x=linspace(a,b,100)` *e* `y=linspace(c,d,100)`*;*
- *costruisca la matrice* `iter` *di dimensione* `length(x)`×`length(y)` *tale che l'elemento* `(h,k)` *di* `iter` *sia il numero di elementi della successione generata dalla function* `segno`, *a partire da* `z = x(h) + 1i*y(k)`*;*
- *restituisca in output le variabili* `x`, `y`, `iter`.

Scrivere poi uno script che usi la function `bsegno` *per disegnare i bacini di attrazione per varie scelte dei valori in input. Si potrà cominciare per esempio con* `a=-5`, `b=5`, `c=-5`, `d=5`, `epsilon=1e-8`, `maxiter=20` *e usare il comando* `imagesc(iter')`. *(In questo caso vogliamo ottenere un'immagine a colori, perciò non useremo la colormap in toni di grigio). Sarà interessante modificare gli estremi degli intervalli per ingrandire delle zone vicine a* 1 *o* -1, *o vicino all'asse immaginario.*

3. Insieme di Mandelbrot

Sia $s \in \mathbb{C}$ e si definisca l'iterazione

$$\begin{aligned} z_1 &= s, \\ z_{n+1} &= z_n^2 + s, \qquad n \geq 1. \end{aligned}$$

L'*insieme di Mandelbrot* è definito come l'insieme dei punti s nel piano complesso tali che l'iterazione $\{z_n\}_{n\geq 1}$ definita a partire da s sia limitata.

Vogliamo disegnare l'insieme di Mandelbrot sul piano complesso. Per far questo costruiamo un reticolo in un rettangolo $[a, b] \times i[c, d]$, definiamo un'opportuna matrice W, e per ogni punto s del reticolo assegnamo all'elemento corrispondente di W il valore del K-esimo elemento della successione definita a partire da s, dove K è un indice prefissato.

Esercizio 4 *Scrivere una function* `W = mandel (a, b, c, d, K)` *che:*

- *prenda in input gli estremi* `a`, `b`, `c`, `d` *degli intervalli* $[a, b]$ *e* $[c, d]$, *e il numero intero positivo* `K`,
- *suddivida* $[a, b]$ *e* $[c, d]$ *in sottointervalli; come nell'esercizio precedente, siano* `x` *e* `y` *le variabili contenenti le discretizzazioni di* $[a, b]$ *e* $[c, d]$, *rispettivamente;*
- *costruisca la matrice* `W` *tale che* `W(h,k)` *sia il* `K`*-esimo elemento della successione* $\{z_k\}_{k\geq 1}$ *definita a partire da* s =`x(h)+1i*(y(k))`.

  Attenzione: per certi valori di s *la successione diverge rapidamente e il valore calcolato di* z_K *risulta essere* `NaN`. *Per evitare questo fenomeno è opportuno introdurre un controllo sul modulo di* z_n. *Se il modulo di* z_n *è più grande di un certo valore (per esempio* `1.e16`*) si interrompe l'iterazione e si assegna a* `W(h,k)` *l'ultimo valore calcolato.*

Scrivere uno script che applichi la function `mandel` *e disegni l'immagine ottenuta usando il comando* `imagesc(exp(-abs(W')))`*. La funzione* `exp` *permette di appiattire vicino a zero gli elementi grandi.*

Provare dapprima con `W=mandel(-2,0.6,-1,1,20);` *e scegliere poi altri valori in input per fare degli "zoom" in zone interessanti dell'insieme (per esempio vicino al bordo).*

4. Generalizziamo l'insieme di Mandelbrot

Sia $s \in \mathbb{C}$, sia p un intero maggiore di 1, e si definisca l'iterazione

$$\begin{aligned} z_1 &= s, \\ z_{n+1} &= z_n^p + s, \qquad n \geq 1. \end{aligned}$$

Come nella sezione precedente, vogliamo disegnare l'insieme dei punti s nel piano complesso tali che la successione associata è limitata.

Esercizio 5 *Scrivere una function* `W = mandelp (a, b, c, d, p, K)` *e uno script che disegni degli insiemi di Mandebrot in modo simile all'esercizio precedente, usando questa volta la successione generalizzata. Si provino diversi valori di p. Per esempio:*
`W=mandelp(-1.2,1.2,-1.2,1.2,8,20);`

5. Insiemi di Julia

Fissati $s, z_1 \in \mathbb{C}$, si definisca la successione

$$\begin{aligned} &z_1, \\ &z_{n+1} = z_n^2 + s, \qquad n \geq 1. \end{aligned}$$

Vogliamo disegnare i bacini di attrazione della successione al variare di z_1 scelto in un rettangolo del piano complesso (mentre s resta fisso).

Esercizio 6 *Scrivere una function* `Z = julia(a, b, c, d, s, K)` *che calcoli e disegni i bacini di attrazione della successione, nel modo seguente:*

- *si suddividano $[a, b]$ e $[c, d]$ in sottointervalli, ottenendo un reticolo come negli esercizi precedenti;*
- *per ciascun punto* `x(h)+1i*y(k)` *del reticolo si calcolino gli elementi della successione $\{z_n\}_{n=1,\ldots,K}$ con $z_1 =$* `x(h)+1i*y(k)` *e si definisca* `Z(h,k)` *come l'ultimo elemento calcolato (servirà anche qui un controllo sul modulo di z_n);*
- *si disegni la figura definita da* `Z`*.*

Che cosa ottenete scegliendo $a = c = -1.5$, $b = d = 1.5$, $s = 0.27334 - 0.00742i$, $K = 20$? E per $s = -0.82 - 0.2i$?

Scegliamo $s = -0.70176 - 0.3842i$, dapprima con $K = 20$ e poi con $K = 30$: notate una differenza?

Provate con altri valori dei parametri.