Esercizi Laboratorio Calcolo Numerico 1
- Settimana 2 -

Nota: per i comandi non esplicitamente introdotti nel video di spiegazione si può utilizzare `help` oppure `doc` dei comandi stessi

1. **MATRICI: DEFINIZIONE, OPERAZIONI**

   (a) Dati i vettori

   $$a_1 = \begin{bmatrix}100 & 200 & 300 & 400 & 500\end{bmatrix}, \qquad a_2 = \begin{bmatrix}9 & 7 & 5 & 3 & 1 & -1 & -3\end{bmatrix},$$
   $$a_3 = \begin{bmatrix}2 & 4 & 8 & 16 & 32\end{bmatrix}, \qquad a_4 = \begin{bmatrix}3 & 3 & 3 & 3 & 3 & 3 & 3\end{bmatrix},$$

   costruire (se definite) le matrici

   $$A = \begin{bmatrix}a_1\\ a_2\\ a_3\\ a_4\end{bmatrix} \qquad B = \begin{bmatrix}a_1^t & a_2^t & a_3^t & a_4^t\end{bmatrix} \qquad C = A * a_1 \qquad D = a_1 * A$$

   Determinare la dimensione di ciascuna matrice con il comando `size`

   (b) Scrivere una linea di comando Matlab che produca la seguente matrice come risultato del prodotto di due vettori opportunamente scelti

   $$M = \begin{bmatrix}1 & 2 & 3 & 4\\ 1 & 2 & 3 & 4\\ 1 & 2 & 3 & 4\\ 1 & 2 & 3 & 4\end{bmatrix}$$

   (c) Considerando la matrice M costruita al punto precedente:

   i. estrarre la terza colonna di M e immagazzinarla nel vettore b. Verificare le dimensioni di b

   ii. estrarre la prima e la terza colonna di M e immagazzinarle nella matrice C. Verificare le dimensioni di C

   iii. sommare la prima e la quarta riga di M e immagazzinarle nel vettore d. Verificare le dimensioni di d. È possibile calcolare $b + d$?

   (d) Creare la matrice B di dimensione 10×10 che abbia il valore 3 sulla diagonale principale, il valore -1 sulla prima sovra e sotto-diagonale e il valore 2 sulla seconda sovra e sotto-diagonale (usare il comando `diag`)

   (e) Calcolare la matrice inversa, B^{-1}, di B e verificare quanto vale in Matlab il prodotto $B * B^{-1}$

2. **OPERATORI RELAZIONALI**

   Dato il vettore $v = [-1, 5, 2, -10, 0, 2, 4, 7]$, scrivere le istruzioni Matlab che permettono di:

(a) contare quanti elementi di v sono maggiori di 0 e minori di 3

(b) contare quanti elementi di v sono minori o uguali di 1 oppure maggiori di 4

3. SINTASSI '.' di Matlab, FUNCTION HANDLE `@`

(a) Disegnare con una sola riga di comando la seguente funzione, nell'intervallo $I = [0, 5]$

$$f(x) = \begin{cases} x+2 & \text{per} \quad 0 \le x \le 1, \\ x & \text{per} \quad 1 < x \le 3, \\ 2 & \text{per} \quad x > 3. \end{cases}$$

A tale scopo ci si aiuti con i comandi relazionali

(b) Sia $p(x) = 2x^3 - 3$, per $x \in [-1, 3]$

i. definire il polinomio tramite i suoi coefficienti, trovarne le radici con il comando `roots`

ii. disegnare il polinomio su un grafico utilizzando i comandi `polyval` e `plot`, avendo definito `x=linspace(-1,3)`

iii. definire il polinomio con un function handle `f=@(x)...` e disegnarlo poi con il comando `plot(x,f(x))`

4. COMANDI FOR, IF

(a) Costruire il vettore x di N elementi con componente x_n data da

$$x_n = (-1)^{n+1}/(2n-1), \quad n = 1, \ldots, N$$

Utilizzare a tale scopo due approcci diversi: un ciclo `for` oppure un comando compatto di Matlab che faccia uso della notazione '.' Misurare il tempo impiegato a costruire x nei due casi considerando N grande e racchiudendo ogni porzione di codice tra i comandi `tic...toc`

(b) Creare uno script Matlab che scelga un intero tra 1 e 10 (comando `randi`), poi chieda all'utente di indovinare tale numero, confronti i due numeri e quindi stampi il messaggio 'Congratulazioni, hai indovinato!' oppure 'Sarai più fortunato la prossima volta' a seconda del risultato ottenuto

(c) Trovare tutti gli interi tra 1 e 10000 multipli di 37. Proporre almeno due differenti modi di risolvere questo problema (una versione che utilizza istruzioni `for, if` e una versione compatta)

(d) Costruire i termini della serie di Fibonacci definita come

$$s_0 = 1, \ s_1 = 1, \ s_j = s_{j-2} + s_{j-1}, \qquad j = 2, 3, \ldots$$

Verificare che per j abbastanza grande (ad esempio $j = 100$) vale che

$$\frac{s_{j+1}}{s_j} \to \frac{1+\sqrt{5}}{2} := \Phi \qquad \text{(rapporto aureo)}$$

Disegnare un grafico che mostri questo comportamento (Suggerimento: per disegnare il valore ϕ costruire un vettore di tanti elementi quanti quelli di s tutti di valore pari a ϕ stesso)

5. **ESERCIZIO di RICAPITOLAZIONE (impegnativo!)**

Creare uno script Matlab `bugs.m` che implementi la souzione del seguente problema.

Considerare una griglia nella quale in alcune posizioni vivono degli "insetti virtuali" (bugs). La griglia è data come una matrice A di dimensione $m \times n$, con valore 0 nelle celle vuote e 1 nelle celle (i, j) occupate dai bugs (si veda un esempio in Fig. 1 per $m = 30, n = 20$).

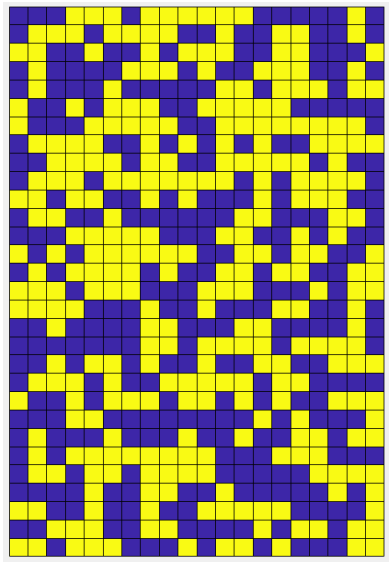

Figura 1: Posizione nella griglia (matrice di dimensione 30×20) dei bugs (in colore giallo) con la scelta di matrice di densità media.

Trovare il numero medio di bugs che circondano ogni bug (ovvero nelle 4 celle a nord, ovest, sud, est, rispettivamente). Creare le seguenti griglie di differente densità, per $m = 30, n = 20$:

```
A = rand(m,n) < 0.1; % sparsa  (pochi bugs)
A = rand(m,n) < 0.5; % media
A = rand(m,n) < 0.7; % densa (molti bugs)
```

Nota 1. Per ottenere dei valori ripetibili, si consiglia di premettere alla generazione della matrice A la seguente riga
`rng('default')`
che permette di fissare il *seed* del generatore di numeri casuali e quindi di ottenere sempre gli stessi numeri "random"

Nota 2. Si può calcolare il numero totale di bugs (elementi non nulli) della matrice A con il comando `nnz(A)`

Nota 3. Nello sviluppo del codice, può essere di aiuto considerare inizialmente una matrice di dimensioni più piccole e visualizzare la sua struttura con il comando `spy` oppure utilizzando le seguenti istruzioni (con cui è ottenuta anche la figura)

```
Ab=[A zeros(m,1)]; Ab=[Ab; zeros(1,n+1)];
figure, pcolor(Ab), set(gca,'YDir', 'reverse')
```