

# Esame di Programmazione Informatica

14 settembre 2021

## Informazioni generali

- 2 esercizi di script MATLAB e 2 domande a risposta multipla.
- La durata dell'esame è di 2 ore.
- È consigliato l'uso di MATLAB nello svolgimento di tutti i punti.
- Le slide e gli script del corso sono consultabili liberamente.
- Il punteggio assegnato ad ogni esercizio ed alle domande a risposta multipla è indicato tra parentesi.

## Esercizio 1 (14/30)

Scrivere una funzione MATLAB che prenda come argomenti in ingresso:

- una matrice  $\mathbf{A}$ ,
- un intero  $n$ ,
- un valore reale  $\varepsilon$

e calcoli, mediante un ciclo `for`, la seguente matrice  $\mathbf{M}$  ottenuta moltiplicando  $\mathbf{A}$  per se stessa  $n$  volte:

$$\mathbf{M} = \underbrace{\mathbf{A} \times \mathbf{A} \times \cdots \times \mathbf{A}}_{n \text{ volte}} = \mathbf{A}^n$$

La funzione dovrà restituire come argomenti in uscita:

- la matrice  $\mathbf{M}$ ,
- un valore booleano (`logical`) che dovrà essere vero se tutti gli elementi  $m_{ij}$  di  $\mathbf{M}$  rispettano la condizione  $|m_{ij}| < \varepsilon$ , falso altrimenti.

Utilizzare la precedente funzione nel caso di  $\mathbf{A} = \begin{bmatrix} 0.25 & 1 \\ 1 & 0.25 \end{bmatrix}$ ,  $n = 100$  e  $\varepsilon = 10^{-3}$ .

*Soluzione:* inizializziamo la matrice  $\mathbf{M}$  con  $\mathbf{A}$  ed utilizziamo un ciclo `for` con  $n - 1$  iterazioni per calcolare il prodotto  $\mathbf{A}^n$ . Dopodichè utilizziamo le funzioni MATLAB `max()` e `abs()` per controllare se la matrice risultante  $\mathbf{M}$  rispetta la condizione  $|m_{ij}| < \varepsilon \forall i, j$ :

`prodotto_e_controllo.m`

```
function [M, controllo] = prodotto_e_controllo(A, n, epsilon)
    M = A ;
    for i = 1 : n-1
        M = M * A ;
    end
    controllo = max( abs(M(:)) ) < epsilon ;
end
```

Per utilizzare la funzione nel caso della matrice richiesta  $\mathbf{A} = \begin{bmatrix} 0.25 & 1 \\ 1 & 0.25 \end{bmatrix}$  con  $n = 100$  e  $\varepsilon = 10^{-3}$ , sarà sufficiente definire la matrice e richiamare direttamente la funzione:

```
A = [ 0.25 1 ; 1 0.25 ] ;
[An, condizione] = prodotto_e_controllo(A, 100, 1e-3)
```

ottenendo correttamente una `condizione` falsa (la matrice  $\mathbf{A}$  ha un autovalore  $> 1$ ).

## Esercizio 2 (14/30)

Date le due seguenti funzioni:

$$f(x) = \frac{1}{1 - \sin x}$$
$$g(x) = \sum_{i=0}^N (\sin x)^i$$

scrivere una funzione MATLAB che prenda come argomenti in ingresso:

- un vettore  $\mathbf{x}$  di valori di  $x$ ,
- un intero  $N$

e restituisca come argomenti in uscita i due vettori  $\mathbf{f}$  e  $\mathbf{g}$  delle due funzioni  $f(x)$  e  $g(x)$  calcolate per tutti i valori  $x$  del vettore  $\mathbf{x}$ , utilizzando il più possibile le operazioni vettoriali/elemento per elemento.

Si mostri poi come utilizzare la precedente funzione nel caso di un vettore  $\mathbf{x}$  di 1000 valori di  $x$  equidistanziati nell'intervallo  $[-\pi, \pi/4]$  ed  $N = 6$ . Si faccia quindi il plot dei due grafici nella medesima figura.

Cosa succede aumentando  $N$ , ad esempio utilizzando  $N = 10, 100, 1000$ ?

*Soluzione:* il vettore  $\mathbf{f} = f(\mathbf{x})$  si può calcolare immediatamente tramite operazioni vettoriali, mentre il vettore  $\mathbf{g} = g(\mathbf{x})$  necessita di un ciclo `for` sugli  $N + 1$  elementi della sommatoria, inizializzando opportunamente a 0 i valori del vettore  $\mathbf{g}$  di partenza:

fg.m

```
function [f, g] = fg(x, N)
    s = sin(x) ;
    f = 1 ./ ( 1 - s ) ;

    g = 0*x ;
    for i = 0 : N
        g = g + s .^ i ;
    end
end
```

Utilizzo nel caso di un vettore  $\mathbf{x}$  di 1000 valori di  $x$  equidistanziati nell'intervallo  $[-\pi, \pi/4]$  ed  $N = 6$  e relativi plot grafici:

```
x = linspace(-pi, pi/4, 1000) ;
N = 10 ;
[f, g] = fg(x, N) ;
plot(x, f) ;
hold on ;
plot(x, g) ;
```

Aumentando  $N$  le due curve tenderanno ad assomigliarsi, tranne nel punto  $\bar{x} = -\pi/2$  dove  $f(\bar{x}) = 1/2$ , mentre  $\sin(\bar{x}) = -1$  e quindi  $g(\bar{x})$  oscillerà indefinitamente tra 0 e 1.

## Domande a risposta multipla (5/30)

### Domanda 1 (2/30)

In MATLAB, la successione di Fibonacci definita da  $x_{n+1} = x_n + x_{n-1}$  (e dati due valori qualsiasi di partenza  $x_1$  e  $x_2$ ):

- non si può implementare direttamente;
- si può implementare solo in maniera sequenziale e non vettoriale (cioè sono costretto ad utilizzare esplicitamente un ciclo `for`);
- si può implementare anche in maniera vettoriale senza cicli `for`, basta calcolare i vettori dei valori  $x_n$  e  $x_{n-1}$  e sommarli;
- si può calcolare ma solo nel caso di valori  $x_i$  interi;

*Soluzione:* si può implementare solo in maniera sequenziale e non vettoriale: sono costretto ad utilizzare esplicitamente un ciclo `for`.

### Domanda 2 (3/30)

Data la seguente funzione MATLAB:

```
function y = f(x,y)
    if y > x
        y = sin(x) ;
    else
        y = cos(x) ;
    end
    y = tan(x) ;
end
```

che funzione matematica  $f(x, y)$  essa implementa?

- $\sin x$ ;
- $\cos x$ ;
- $\tan x$ ;
- una funzione definita a tratti:  $\sin x$  se  $x < y$  o  $\cos x$  altrimenti;

*Soluzione:*  $\tan x$ .