

Ciclo di seminari: Metodi Computazionali per la Finanza

C.d.L.M. Finanza e Assicurazioni – a.a. 2018/2019

Lezione 1: Introduzione a Matlab

Informazioni generali

- Principali metodi numerici per il pricing di derivati (**Matlab**)
 - metodo binomiale
 - metodi alle differenze finite
 - metodo Monte Carlo
- **Orario seminari:**
martedì 11 – 14 (Lab. piano 1)

Matlab (*Matrix Laboratory*): linguaggio di programmazione ad alto livello per il calcolo scientifico (**linguaggio interprete**)

Ambiti applicativi principali

- Matematica Applicata (statistica)
- Ingegneria (teoria dei sistemi e del controllo)
- Economia (finanza)

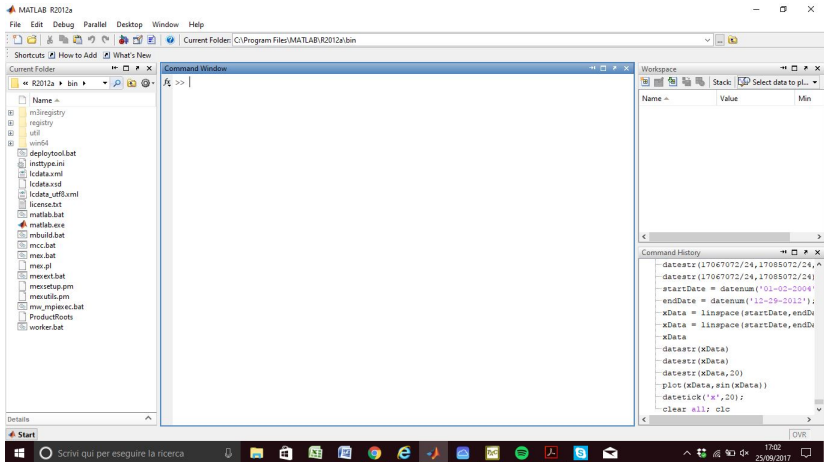
Variabili in Matlab

Struttura dati principale: **matrice** $m \times n$

- $m = n = 1$ (scalare) $a = 3;$
- $m = 1, n > 1$ (vettore riga) $b = [1, 2];$
- $m > 1, n = 1$ (vettore colonna) $c = [1; 2];$
- $m = n > 1$ (matrice quadrata) $A = [1 \ 3; 1 \ 2];$

dichiarazione automatica delle variabili

Workspace e Command window



Workspace e Command window

Matlab registra e mantiene tutti i cambiamenti alle variabili nello **workspace** e permette di digitare i comandi dalla **command window**

- **who** visualizza tutte le variabili
- **whos** visualizza tutte le variabili e le rispettive dimensioni
- **clear all** cancella tutte le variabili
- **clear a** cancella la variabile *a*
- **clc** pulisce la command window

Programmazione in Matlab

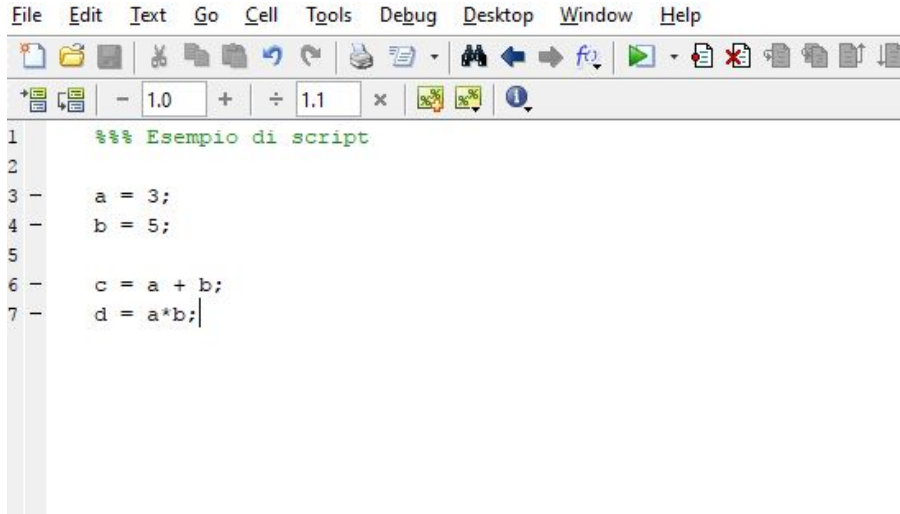
Script

- sequenze di istruzioni raccolte in un file.m
- per esecuzione, scrivere nome del file in Command Window
- no variabili input/output, visibilità variabili totale

function

- sequenze di istruzioni, in cui specificare variabili di ingresso e di uscita
- sintassi: *function[out1,out2] = nomefunction(in1,in2,in3)*
- variabili interne locali
- per esecuzione, invocare da linea di comando con opportuni parametri

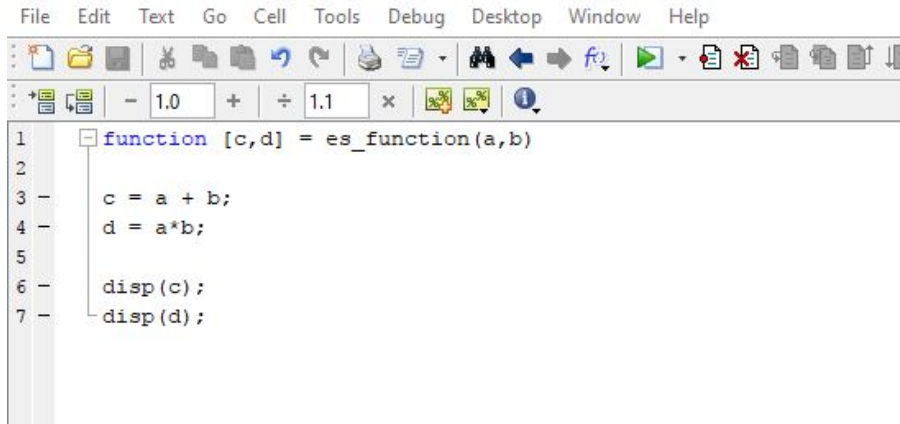
Un esempio di script



The image shows a screenshot of the MATLAB script editor. The window has a menu bar with 'File', 'Edit', 'Text', 'Go', 'Cell', 'Tools', 'Debug', 'Desktop', 'Window', and 'Help'. Below the menu bar is a toolbar with various icons for file operations, editing, and execution. The script editor area shows a script with the following content:

```
1      %%% Esempio di script
2
3      a = 3;
4      b = 5;
5
6      c = a + b;
7      d = a*b;
```


Un esempio di function



The image shows a screenshot of the MATLAB IDE. The top menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. Below the menu is a toolbar with various icons for file operations, editing, and execution. The command window shows a calculation: `- 1.0 + 1.1 x` resulting in `1.1`. The editor window displays a function definition for `es_function` with the following code:

```
1 function [c,d] = es_function(a,b)
2
3     c = a + b;
4     d = a*b;
5
6     disp(c);
7     disp(d);
```

Operazioni fondamentali

Operazioni tra matrici consentite se le dimensioni sono compatibili

- Somma e sottrazione: $A + B$, $A - B$
- Moltiplicazione (prodotto righe per colonne): $A * B$
- Elevamento a potenza: $A \wedge 2$
- Operazioni elemento per elemento: $A. * B$ $A. \wedge 2$
- Trasposizione: A'

Sottomatrici

Se A è una matrice $m \times n$

- estrazione dell'elemento in posizione i, j : $A(i, j)$
- estrazione dell' i -sima riga: $A(i, :)$
- estrazione dell' i -sima colonna: $A(:, j)$
- estrazione del vettore degli elementi dall' i -simo al k -simo della j -sima colonna: $A(i : k, j)$
- estrazione del vettore degli elementi dal j -simo all' h -simo della i -sima riga: $A(i, j : h)$
- estrazione di una sottomatrice: $A(i : k, j : h)$
- estrazione dell' i -sima diagonale: $\text{diag}(A, i)$

Matrici notevoli

Se n , m sono interi positivi e v un vettore di lunghezza n

- matrice identità $n \times n$: `eye(n)`
- matrice di elementi tutti pari a 1 : `ones(n, m)`
- matrice di elementi tutti pari a 0 : `zeros(n, m)`
- matrice i cui elementi sono numeri pseudo-casuali, generati da una distribuzione uniforme in $[0, 1]$: `rand(n, m)`
- matrice i cui elementi sono numeri pseudo-casuali, generati da una distribuzione gaussiana standard: `randn(n, m)`
- matrice diagonale: `diag(v)`

► Esercizio 1

Manipolazione di vettori e matrici

Se A è una matrice $m \times n$ e v un vettore,

- media e varianza di una matrice: *mean*(A) e *var*(A)
- rango di una matrice: *rank*(A)
- somma, somma cumulata e prodotto degli elementi di una matrice: *sum*(A), *cumsum*(A) e *prod*(A)
- somma, somma cumulata e prodotto degli elementi di una matrice lungo le colonne: *sum*($A, 1$), *cumsum*($A, 1$) e *prod*($A, 1$)
- somma, somma cumulata e prodotto degli elementi di una matrice lungo le righe: *sum*($A, 2$), *cumsum*($A, 2$) e *prod*($A, 2$)
- creazione di una matrice $p \times q$ come copia della matrice A : *repmat*(A, p, q)

Manipolazione di vettori e matrici/2

Se A è una matrice $m \times n$ e v un vettore,

- max e min di una matrice: $\max(A)$ e $\min(A)$
- max e min di una matrice lungo la p -sima riga o (colonna):
 $[p, i] = \max(A)$ e $[p, i] = \min(A)$
- vettori equispaziati: $[a : s : b]$ o $\text{linspace}(a, b, N)$
- ricerca degli elementi non nulli di una matrice: $\text{find}(A)$
- ordinamento degli elementi di una matrice (in ordine crescente): $\text{sort}(A)$
- operatori di confronto: $>$, $<$, \leq , \geq , $==$, \sim
- operatori logici: $\&$, $|$, \sim

► Esercizio 2

► Esercizio 3

Costrutti sintattici fondamentali

Ciclo *for*: ripete un blocco di istruzioni un numero **determinato** di volte

sintassi:

```
for indice = inizio : passo : fine  
    blocco istruzioni  
end
```

NB: è possibile annidare più cicli *for*

Costrutti sintattici fondamentali

Ciclo *while*: ripete un blocco di istruzioni fino a quando una data condizione è verificata

sintassi:

```
while  condizione di persistenza del ciclo  
        blocco istruzioni  
end
```


Costrutti sintattici fondamentali

Istruzioni condizionali: controllano l'esecuzione di un determinato blocco di codice, a seconda del valore assunto da una certa espressione logica

sintassi:

```
if espressione 1
    blocco istruzioni 1
elseif espressione 2
    blocco istruzioni 2
else espressione 3
    blocco istruzioni 3
end
```

Funzioni matematiche

Matlab consente di operare su funzioni matematiche

sintassi: ($f = @(var_1, var_2, \dots) \text{functionexpression}$) (**anonymous functions**) oppure tramite m-file (operazioni element-wise)

Data la seguente funzione $f = 1 + x^2 + 3xy - 5y^2$, in Matlab

```
f = @(x,y) 1 + x.^2 + 3 * x * y - 5y.^2;
```

Funzioni matematiche

Matlab consente di operare su funzioni matematiche

sintassi: ($f = @(var_1, var_2, \dots) \text{functionexpression}$) (**anonymous functions**) oppure tramite m-file (operazioni element-wise)

Data la seguente funzione $f = 1 + x^2 + 3xy - 5y^2$, in Matlab

$$f = @(x, y) 1 + x.^2 + 3 * x * y - 5y.^2;$$

La grafica in Matlab

Matlab permette di creare grafici bidimensionali e tridimensionali

Sintassi (2-dim): *plot*(*x*, *y*, **opzioni**)

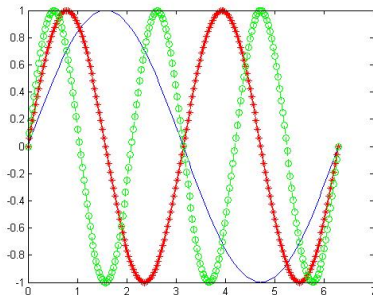
- *x* : vettore da rappresentare in ascissa
- *y* : vettore da rappresentare in ordinata (funzione)
- *opzioni*: linestyle, marker, linecolor, etc...

Anche più grafici nella stessa figura: **hold on** e **subplot**

► Esercizio 5

La grafica in Matlab – Esempio 2D

```
x = 0 : pi/100 : 2 * pi;  
y1 = sin(x);  
y2 = sin(2 * x);  
y3 = sin(3 * x);  
plot(x, y1);  
hold on;  
plot(x, y2, 'r *');  
plot(x, y3, 'go -');
```



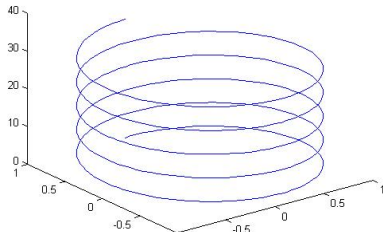
La grafica in Matlab

Nel caso tridimensionale, è possibile tracciare curve o superfici

Sintassi (curve): `plot3(x,y,z)`

- x : vettore da rappresentare in ascissa
- y : vettore da rappresentare in ordinata (funzione)
- z : vettore da rappresentare in quota

```
x = 0 : pi/50 : 10 * pi;  
plot3(sin(x), cos(x), x)
```



La grafica in Matlab

Sintassi (superfici): $[X, Y] = \text{meshgrid}(x, y)$; $\text{mesh}(X, Y, Z)$
oppure $\text{surf}(X, Y, Z)$ **oppure** $\text{contour}(X, Y, Z, n)$

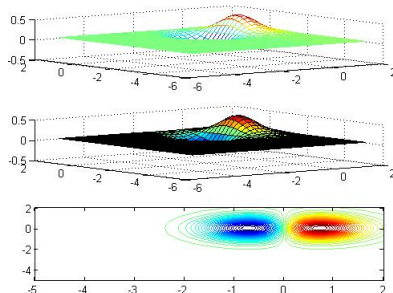
- x, y : vettori delle variabili della funzione
- Z : funzione da graficare
- *meshgrid*: griglia di base che individua tutti i nodi
- *mesh*: scheletro della superficie
- *surf*: superficie
- *contour*: n curve di livello

La grafica in Matlab – Esempi 3D

```
x = -5 : 0.2 : 2;  
y = -5 : 0.2 : 2;  
[X, Y] = meshgrid(x, y);  
Z = X.*exp(-(X.^2 + Y.^2));  
subplot(3,1,1);  
mesh(X, Y, Z);  
subplot(3,1,2);  
surf(X, Y, Z);  
subplot(3,1,3);  
contour(X, Y, Z, 50);
```

► Esercizio 6

► Esercizio 7



Esercizio 1

Immettere la matrice:

$$A = \begin{pmatrix} 1 & 0 & 2 & 3 \\ 0 & -1 & 2 & -2 \\ 1 & 2 & 1 & -3 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{e il vettore: } \underline{b} = (1, 4, 1, -2)'$$

- ❶ calcolare A^T , $\det(A)$, A^{-1} (se esiste);
- ❷ calcolare $A * \underline{b}$, $\underline{b}^T * A$;
- ❸ calcolare la somma delle componenti del vettore \underline{b} ;
- ❹ costruire una matrice che ha 3 colonne tutte uguali al vettore \underline{b} ;
- ❺ costruire una matrice che ha 5 righe tutte uguali al vettore \underline{b} ;
- ❻ costruire una matrice tridiagonale che ha sulla diagonale principale i valori da $-m$ ad m , essendo m la lunghezza del vettore \underline{b} , e sulla prima sopradiagonale e sulla prima sottodiagonale vettori unitari.

Esercizio 2

Con riferimento alla matrice A ed al vettore \underline{b} dell'Esercizio 1:

- 1 Calcolare: i totali di riga e di colonna della matrice A ; la media delle colonne di A e la varianza di \underline{b} ; il minimo tra la seconda riga di A e \underline{b} ; il massimo di ogni riga di A e trovarne la posizione nelle colonne
- 2 Costruire la matrice A_1 ottenuta cancellando la terza riga della matrice A e la matrice A_2 ottenuta premettendo ad A_1 una colonna di zeri ed aggiungendo alla fine una riga di 1
- 3 Utilizzando gli appositi comandi di confronto, dire se il terzo elemento del vettore \underline{b} è positivo e contare gli elementi nulli della matrice A_2
- 4 Trovare per quali indici il vettore \underline{b} ha valori compresi tra 0 e 3

◀ Go Back

Esercizio 3

Sia $t = (1, 4, 1, -2, 0, -1)$.

- 1 Costruire un vettore con i soli elementi di t di posto dispari.
- 2 Costruire un vettore con i soli elementi di t compresi tra 0 e 3.
- 3 Ordinare t in ordine crescente.
- 4 Permutare gli elementi di t dall'ultimo al primo.
- 5 Scambiare il secondo e il quinto elemento di t .

◀ Go Back

Esercizio 4

- 1 Costruire una matrice M con 25 righe e 5 colonne, avente al posto i, j l'elemento $a_{ij} = (i + j)/(ij)$
- 2 Costruire il vettore b composto dalla quarta colonna di A . Fare la somma cumulata delle componenti del vettore b . Costruire un vettore che abbia le stesse componenti di b , ma solo quelle corrispondenti ad una somma cumulata minore o uguale a 5.
- 3 Continuare ad estrarre a caso un numero tra 1 e 6 (usare il comando `unidrnd`) finchè la somma dell'ultima estrazione e di tutte le precedenti non supera 50

Esercizio 5

Siano date le seguenti funzioni:

$$f(x) = x^4 - 10.5x^3 + 39x^2 - 59.5x + 30;$$

$$g(x) = x^4 - 8x^3 + 22x^2 - 24x + 9.$$

- 1 costruire le funzioni f e g
- 2 fare il grafico delle funzioni f e g utilizzando il comando *plot*, rappresentando $f(x)$ nell'intervallo $[1, 4]$ e $g(x)$ nell'intervallo $[0, 4]$;
- 3 fare il grafico delle funzioni f e g utilizzando il comando *fplot*;
- 4 All'interno di un'unica finestra (*figure*), fare il grafico delle funzioni f e g nell'intervallo $[0, 4]$.

Esercizio 6

Sia $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ definita da

$$f(\mathbf{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2, \quad \mathbf{x} := [x_1, x_2].$$

Fare il grafico di f per $\mathbf{x} \in [0, 3] \times [-1, 4]$ con l'utilizzo dei comandi *mesh*, *surf*, *plot3*; fare il grafico delle curve di livello con *contour*. Costruire più grafici sulla stessa figura con *subplot*

◀ Go Back

Esercizio 7

Si consideri la funzione

$$f(\mathbf{x}) = \frac{1}{2\pi \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

(funzione di densità di una normale bivariata).

- 1 costruire la funzione f
- 2 per i valori dei parametri

$$\boldsymbol{\mu} = \begin{pmatrix} -1 \\ 1.3 \end{pmatrix} \text{ e } \Sigma = \begin{pmatrix} 1 & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 2 \end{pmatrix},$$

fare il grafico della funzione, per $\mathbf{x} \in [-5, 3] \times [-4, 6]$, con *mesh*, *surf*, *plot3* e fare il grafico delle curve di livello con *contour*

- 3 misurare il tempo d'esecuzione della procedura implementativa