

FINANZA COMPUTAZIONALE 2022-2023

Appunti di Federico De Angelis Scorsone

INTRODUZIONE

Appunti A.A. 2022-2023; esame pratico in laboratorio + orale

Scritto = 3 esercizi uno per ogni modulo.

Parte orale chiede tutto il libro.

Modulo 1 = tecniche di programmazione per la matematica e la matematica finanziaria

- > i principali schemi di programmazione;
- > richiami algebra lineare;
- > richiami di matematica finanziaria;

Modulo 2 = selezione di portafogli azionari

- > teoria della probabilità;
- > statistica univariata e multivariata;
- > ottimizzazione a singolo obiettivo e multiobiettivo;

Modulo 3 = pricing di contratti derivati

- > elementi di teoria della probabilità (quella riferita ai processi stocastici);
 - > metodo montecarlo per la simulazione dei processi;
 - > principali modelli a tempo discreto (binomiale) e continuo (black-scholes) per la valutazione delle opzioni. Per le call europee i tre metodi (esercizio che capita?).
-

MODULO 1: TECNICHE DI PROGRAMMAZIONE

Introduzione modulo 1

- a) alcuni elementi di base su matlab tra cui come si definiscono le variabili; cos'è un workspace; come si definiscono gli scalari, i vettori e le matrici; algebra lineare, operazioni speciali ed operazioni elemento per elemento ed alcune funzioni predefinite;
- b) .m file script e function; principali costrutti di programmazione (if and, for and , while and);
- c) grafica matlab
- d) esercizi quotidiani



MATLAB usato nel calcolo scientifico. E' importante saper leggere i modelli e tradurli in linguaggi di programmazione. MATLAB è un linguaggio orientato alle matrici (MATrix LABoratory). Una delle potenzialità è proprio la grafica. Inoltre è ottimizzato rispetto a Python.

L'unità di base in MATLAB è l'array → è un concetto più generale della matrice. Le matrici ce le immaginiamo rettangolari, quadrate, ma potrebbe essere anche essere un cubo. Le matrici con più di due dimensioni si chiamano tensori.

L' array è un concetto generale → è un'allocazione di memoria di elementi ordinati.

Le proprietà di MATLAB: ha l'interprete ed è un vero e proprio linguaggio di programmazione. Ha ottime potenzialità grafiche. MATLAB parla con diversi pacchetti applicativi.

Assieme a MATLAB ci sono i toolbox → cioè delle librerie aggiuntive.

[spiegazione dell'ambiente]

Workspace è la memoria. Se dobbiamo caricare i prezzi li carichiamo nel workspace (es. una matrice prezzo con un po' di prezzo). Stiamo lavorando nel current folder. Utile mettere la propria current folder. I diversi modi di usare l'help. Commenti li scriviamo con %.

Matlab è case sensitive, quindi caps lock incide nel codice.

[esempi semplici]

Variabili e numeri dopo la virgola *comando 'format'*

```
>> format short %è quello di default
>> format short e %4 righe dopo la virgola ed esponenziale
>> format long %il numero tutto
>> format long e %lungo esponenziale
>> format short g %pare approssimato
>> format long g %pare approssimato
>> format long %quello delle currency euro e cents 2 cifre dopo virg
```

ci sono una serie di variabili predefinite:

nome variabile	significato
pi	3.14
eps	la precisione del calcolo
realmax	Circa la rappresentazione dei numeri macchina in valore assoluto nell'intervallo [realmin,realmax]
realmin	Circa la rappresentazione dei numeri macchina in valore assoluto nell'intervallo [realmin,realmax]
NaN	Not a number. Occupa spazio ma non è un vettore. Utile per pre-allocare.



quando abbiamo definito una variabile nel workspace, proprio nel workspace possiamo vedere diverse cose, come la *size* (numero righe e colonne), il *valore* assegnato, i *byte* e la *class* (tipo di variabile). In modalità standard usa la precisione doppia (double-precision variables) che usa 8 byte. Le *class* sono le classi degli oggetti presenti in MATLAB. Una classe ha caratteristiche uniche per tutti gli oggetti appartenenti a quella classe.

I byte sono importanti in quanto indicano la memoria allocata. Potremmo avere errori se la memoria è molto grande e supera la nostra capacità.

I nomi delle variabili hanno una lunghezza massima di 31 caratteri alfanumerici. Il primo non può essere un numero. I nomi delle variabili non possono avere caratteri speciali.

Per fare esempio di questo tema possiamo usare il comando `rand` assegnando ad una variabile con un numero molto alto. Il comando '`rand`' crea una matrice NxN dove le N sono di 1000000 in questo caso.

```
>> cc = rand(1000000)
```

La nostra variabile si chiama 'cc'.

inoltre con il comando `clear` puliamo la workspace mentre con il comando `clc` puliamo gli output dell'interprete.

Principali operazioni aritmetiche

addizione '+'

sottrazione '-'

moltiplicazione '*'

divisione '/'

elevazione a potenza '^'

MATLAB fa i calcoli nel seguente ordine:

- 1) prima le operazioni nelle parentesi;
- 2) poi le potenze;
- 3) poi moltiplicazioni e divisioni da sinistra verso destra;
- 4) poi addizioni e moltiplicazioni da sinistra verso destra;

Importante: le parentesi da usare per i calcoli sono solo le parentesi tonde ! Quelle quadrate e graffe le interpreta per altro.

Il comando `...` splitta una istruzione in più pezzi (ossia i tre puntini).

```
>> 2+...  
2  
ans =
```



L'elemento cardine di MATLAB è l'array. L'array è una successione ordinata di memoria che contiene una collezione di elementi dello stesso tipo. Un array può essere composto da 6 tipi di dati: char, double, sparse, cell, uint8, struct.

Con il prompt dei comandi (quello dove appare >>) possiamo usare MATLAB come una vera e propria calcolatrice!

Definizione di vettori o matrici

Noi i vettori li abbiamo visti o come vettori colonna o vettori riga. Si costruiscono in maniera diversa.

Un vettore riga di $n=4$ elementi si può altrimenti definire come una matrice 1×4 .

In aggiunta un vettore colonna di $n=4$ elementi si definisce come una matrice 4×1 .

Come si definiscono in MATLAB ?

```
>> a = [1,2,3,4]
>> a = 1 2 3 4

>> b = [1;2;3;4]
>> b = 1
      2
      3
      4
```

è importante usare le parentesi quadrate. Il vettore riga si può creare anche in modi differenti. Quello colonna invece necessita del punto e virgola. Nell'esempio sotto abbiamo due modalità diverse per descrivere un vettore riga.

```
>> a = [1 2 3 4]
oppure
>> a = [1:4]
```

Da tenere a mente il metodo con i due punti perché sarà utile nel percorso didattico.

Matrici

A questo punto abbiamo gli elementi per poter definire una matrice.

```
>> c = [1 2 3; 4 5 6; 7 8 9]
c =

     1     2     3
     4     5     6
     7     8     9
```