# Geophysical Data Modelling

**Università degli Studi di Napoli Federico II**

**Prof. Maurizio Fedi**

# INDICE

## 1.Introduzione e primi argomenti

**The command window:** finestra dell'interfaccia di ML dove è possibile digitare i comandi. Questi comandi vengono immessi da un **prompt** (>>) e poi eseguite.

**Workspace:** spazio di lavoro in cui compaiono i dati.

**Current directory:** La finestra *Current Directory* permette, come si può intuire, di esplorare il contenuto delle cartelle sul proprio hard disk. Da questa finestra è possibile aprire direttamente file compatibili con MATLAB con un semplice doppio click.

**Command history:** Nella finestra *Command History* sono elencati tutti i comandi digitati di recente, divisi per ora e data. È possibile rilanciare direttamente da Command History, un comando digitato in Command Windows in precedenza, semplicemente con un doppio click.

```
>> date

ans =

  22-Mar-2011
```

### File e cartelle

ML fornisce un modo per organizzare i dati e l'analisi dei dati prodotti utilizzando nomi significativi e prevedibili e dando la possibilità di progettare cartelle che aiutino a tenere traccia delle cose.

### Comandi per la navigazione nelle cartelle

>> Pwd : display current folder: visualizzare la directory corrente

>> cd : change directory - cambia la directory corrente

>> dir: visualizza il contenuto della directory corrente

Con ML è possibile eseguire semplici calcoli aritmetici, in cui valgono le regole classiche dell'aritmetica (segni, parentesi...)

```
a=3.5;
b=4.1;
c=a+b
   7.600
```

Ma anche espressioni più complicate

$$c = \sqrt{a^2 + b^2} \text{ with } a = 3 \text{ and } b = 4$$

```
a=3;
b=4;
c = sqrt(a^2 + b^2);
c=
   5
```

dove sqrt= square root: radice quadrata

Un'altra espressione complicata:

$$c = \sin\frac{n\pi(x - x_0)}{L} \text{ with } n = 2, x = 3, x_0 = 1, L = 5$$

```
n=2;
x=3;
x0=1;
L=5
c=sin(n*pi*(x-x0)/L);
c=
   0.5878
```

dove pi: pigreco

*: moltiplicazione

/: divisione

sin: funzione seno

## Matlab script

Quando si devono eseguire più comandi di ML, conviene inserirli in un file con estensione m. Ciò comporta una serie di vantaggi: velocizzazione delle attività "ripetitive", controllo della correttezza dei file prima della loro esecuzione, oltre che la possibilità di documentare il lavoro svolto dato che il file resta in memoria. Tuttavia, nasconde gli errori (se su 300 righe una riga è sbagliata potrei non notarlo).

**Esempio**

```
% example of simple algebra,
% c=a+b with a=3.5 and b=4.1
a=3.5;
b=4.1;
c=a+b;
c
   7600
```

Con %: simbolo da inserire per i commenti (non eseguibili).

c è senza ; perchè ne voglio conoscere il risultato.

## Esecuzione di uno script

Tasto destro - evaluate selection.

## Vettori e matrici

$$\mathbf{r} = \begin{bmatrix} 2 & 4 & 6 \end{bmatrix} \text{ and } \mathbf{c} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix}^T \text{ and } \mathbf{M} = \begin{bmatrix} 1 & 2 & 4 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Un vettore viene inserito racchiudendo fra parentesi quadre i valori degli elementi, separati da uno spazio bianco o da una virgola:

```
>> myvec=[1 2 3 4] oppure
```

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

Il risultato di questo inserimento `e un vettore riga; per creare un vettore colonna `e sufficiente utilizzare il ";" come separatore fra gli elementi o il "'" dopo la parentesi che contiene gli elementi. Provando le due differenti modalità di inserimento, si ottiene:

```
>> myvec=[1 2 3 4]
```

```
>> myvec = 1 2 3 4
```

oppure

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

```
>> myvec=

        1
        2
        3
        4
```

Oppure

```
>> myvec=[1 2 3 4]'
```

```
>> myvec=

       1
       2
       3
       4
```

Per definire una matrice vengono combinati i caratteri che consentono di inserire vettori riga e colonna: le righe vengono inserite separando gli elementi con uno spazio o una virgola mentre il passaggio ad una nuova riga viene ottenuto inserendo un “;”.

$$\mathbf{r} = \begin{bmatrix} 2 & 4 & 6 \end{bmatrix} \text{ and } \mathbf{c} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix}^T \text{ and } \mathbf{M} = \begin{bmatrix} 1 & 2 & 4 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

si scrive:

```
r = [2, 4, 6];
c = [1, 2, 3]’;
M =[ [1, 4, 7]', [2, 5, 8]', [4, 6, 9]’];
```

oppure

```
M = [1; 4; 7], [2; 5; 8;], [4; 6; 9;]
```

## Operatore di trasposizione

L’operazione di trasposizione di una matrice o di un vettore viene eseguita utilizzando il carattere “’” (“apice”). Ad esempio:

1
2
3
4
diventa [ 1, 2, 3, 4 ] (e viceversa)

La notazione matematica standard per la trasposizione è $a^T$, quella di ML è a’.

## Moltiplicazione tra vettori

(m,n)x(m,n) la moltiplicazione non è possibile

(m,n)x(n,m) la moltiplicazione è possibile

Fissati due numeri interi positivi **m** e **n**, una **MATRICE** è una tabella di **m**x**n** elementi, ordinati in **m** linee orizzontali (dette **righe** della matrice) e **n** linee verticali (dette **colonne** della matrice). Ogni elemento è associato a una coppia ordinata di indici, i cui valori indicano la riga e la colonna in cui è posto l'elemento in questione.

**m** e **n** si dicono rispettivamente la dimensione per riga e per colonna di una matrice.

a) Vettore di dimensioni 3-1 (3 righe, 1 colonna)

b) Vettore di dimensioni 3-1 (3 righe, 1 colonna)

a) e b) non possono essere moltiplicati perchè le loro dimensioni non sono speculari: uno dei due vettori deve avere dimensioni 1-3 (1 riga, 3 colonne). Esempio:

a) 3-1

b) 1,3

M= 3,3: matrice derivante da due vettori

```
a = [1, 3, 5];

b = [3, 4, 5]'

c=a*b

c

   40
```

Se le dimensioni non fossero speculari:

```
a = [1, 3, 5];

b = [3, 4, 5]
```

```
c=a*b
c
```

Incorrect dimensions for matrix multiplication.

$$\mathbf{a} = \begin{bmatrix}1\\3\\5\end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix}2\\4\\6\end{bmatrix} \text{ and } \mathbf{M} = \begin{bmatrix}1 & 0 & 2\\0 & 1 & 0\\2 & 0 & 1\end{bmatrix} \text{ and } \mathbf{N} = \begin{bmatrix}1 & 0 & -1\\0 & 2 & 0\\-1 & 0 & 3\end{bmatrix}$$

```
a = [1, 3, 5];
c = [3, 4, 5]';
M =[ [1, 0, 2]', [0, 1, 0]', [2, 0, 1]'];
   N =[ [1, 0,-1]', [0, 2, 0]', [-1,0, 3]'];
```

**Prodotto scalare (prodotto interno o dot product)**

Poichè non si possono moltiplicare due colonne (perchè sono due matrici di uguali dimensioni, cioè 1x3), posso trasporre la prima rispetto alla seconda, cioè posso "girare" la prima e rendere così possibile l'operazione. In questo consiste il dot product.

$$s = \mathbf{a}^{\mathrm{T}}\mathbf{b} = \begin{bmatrix}1\\3\\5\end{bmatrix}^{T} \begin{bmatrix}2\\4\\6\end{bmatrix} = \begin{bmatrix}1 & 3 & 5\end{bmatrix}\begin{bmatrix}2\\4\\6\end{bmatrix} = 2\times1 + 3\times4 + 5\times6 = 44$$

```
s= a'*b
```

**Prodotto vettoriale (prodotto esterno o tensore)**

$$\mathbf{T} = \mathbf{a}\mathbf{b}^{\mathrm{T}} = \begin{bmatrix}1\\3\\5\end{bmatrix}\begin{bmatrix}2\\4\\6\end{bmatrix}^{T} = \begin{bmatrix}2\times1 & 4\times1 & 6\times1\\2\times3 & 4\times3 & 6\times3\\2\times5 & 4\times5 & 6\times5\end{bmatrix} = \begin{bmatrix}2 & 4 & 6\\6 & 12 & 18\\10 & 20 & 30\end{bmatrix}$$

```
T= a*b';
```