

Matematica Applicata

- MATLAB- Comandi base-

Laboratorio didattico

A.A 2011/2012

Stefano Vena

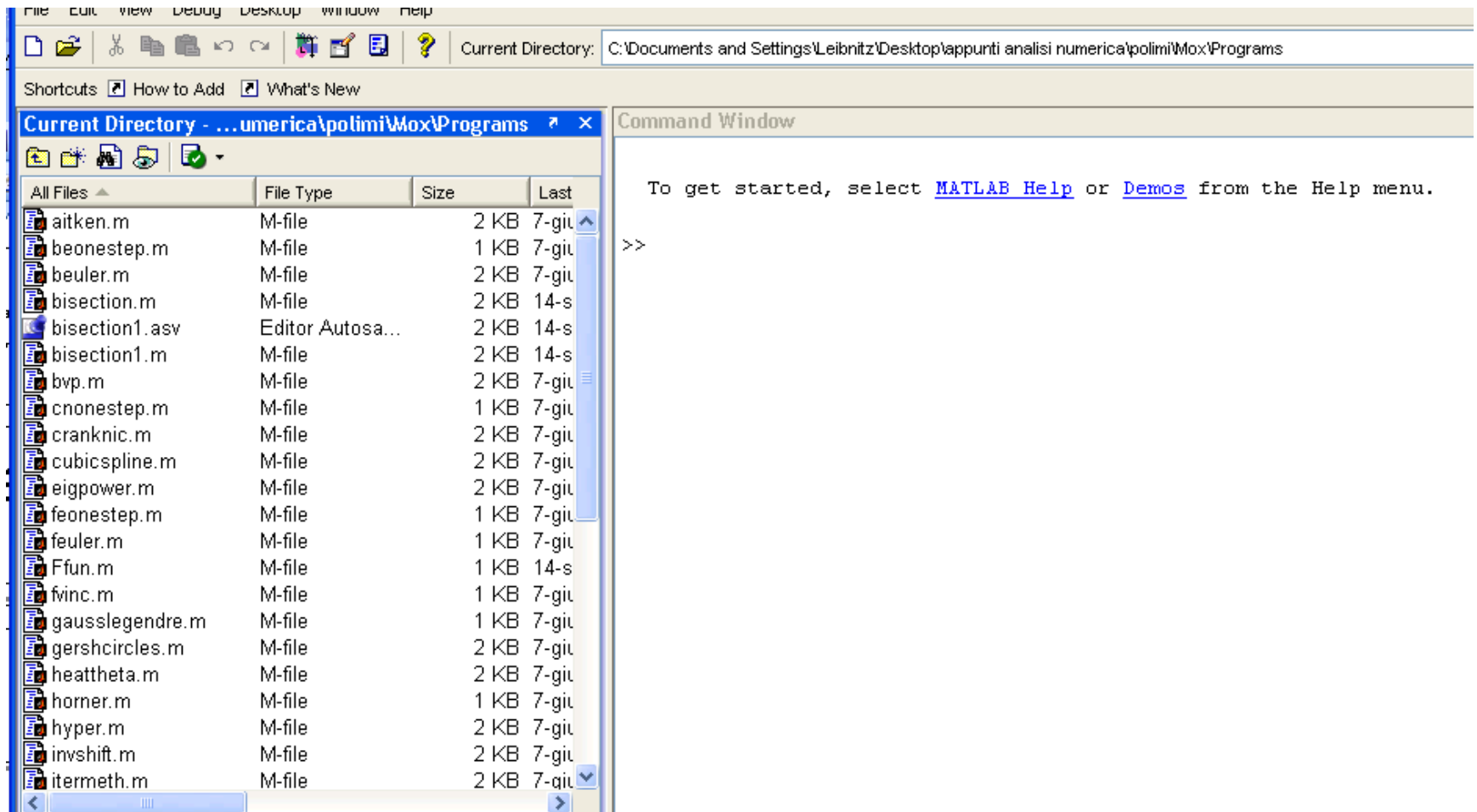
Materiale

- Matlab :
<http://www.mathworks.com/products/matlab>
- Octave :
<http://octave.sourceforge.net/>
- Materiale del Corso :
<https://github.com/wetfire2k/Matematica-Applicata>
- Libro:
Calcolo Scientifico, Esercizi e problemi risolti con
MATLAB e Octave, Alfio Quarteroni, Fausto Saleri.

Primi passi con Matlab

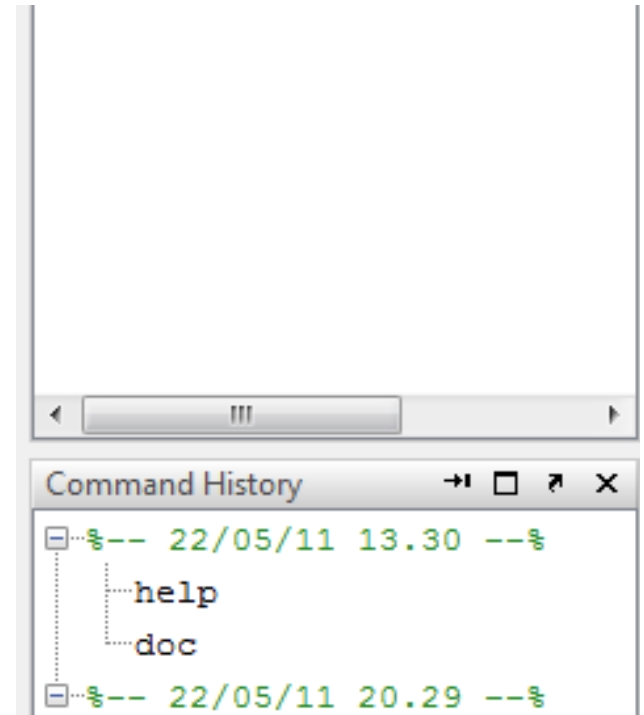
- Avviare Matlab o Octave
- Online help:
 - `help` keyword (e.s. `help cos`).
 - `doc` keyword
- Usare comandi di sistema
 - `mkdir 'MatApp'` crea una directory
 - `cd 'MatApp'` cambia directory
 - `pwd` stampa la directory corrente
 - `ls` lista files nella directory

Matlab: L' ambiente

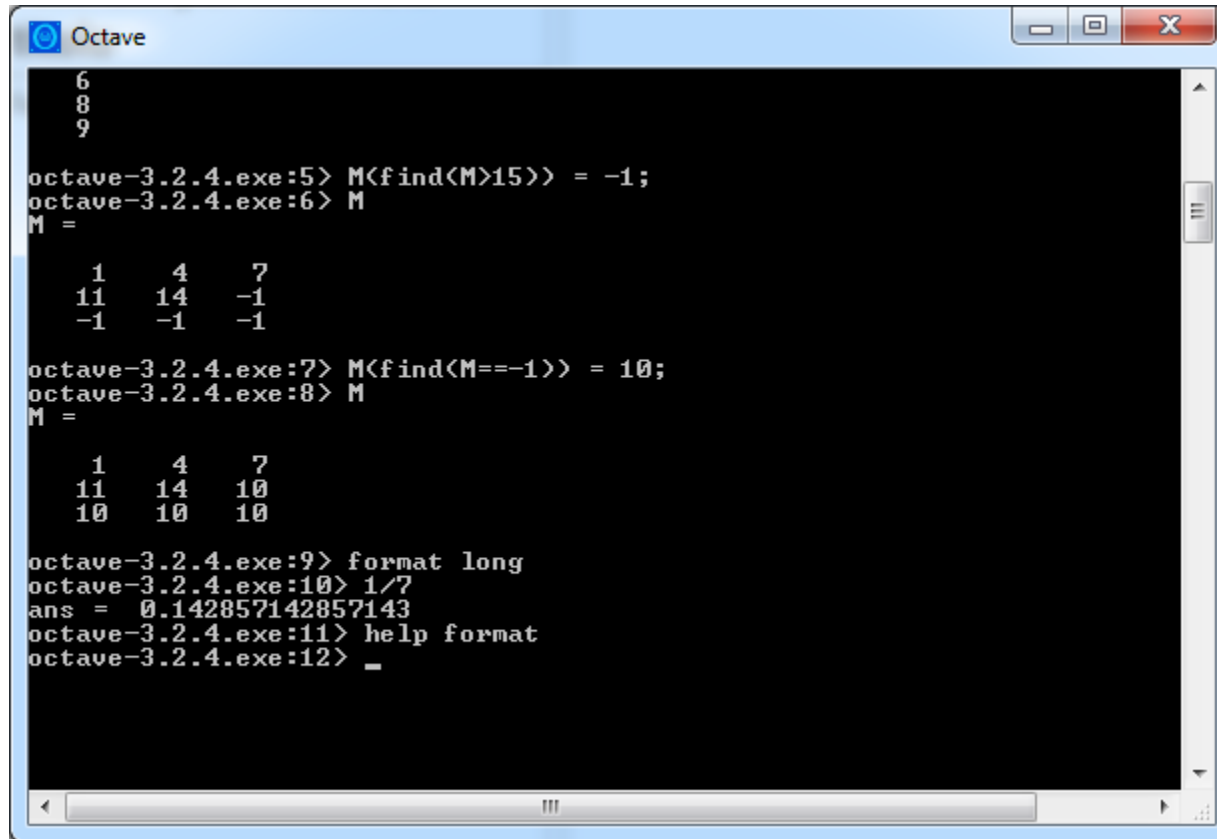


Workspace e Command History

- `ans`, la risposta; in questo caso contiene un numero Double
- Il numero è troncato alla IV cifra decimale
- Arrotondato alla quarta cifra decimale.
- Il comando `format long` aumenta il numero di cifre.
- In realtà la macchina processa fino a 16 cifre decimali.
- La command history contiene la storia degli ultimi comandi eseguiti nell'ambiente



Octave



```
Octave
6
8
9
octave-3.2.4.exe:5> M(find(M>15)) = -1;
octave-3.2.4.exe:6> M
M =
    1    4    7
   11   14   -1
   -1   -1   -1

octave-3.2.4.exe:7> M(find(M== -1)) = 10;
octave-3.2.4.exe:8> M
M =
    1    4    7
   11   14   10
   10   10   10

octave-3.2.4.exe:9> format long
octave-3.2.4.exe:10> 1/7
ans = 0.142857142857143
octave-3.2.4.exe:11> help format
octave-3.2.4.exe:12> _
```

- Garantisce una buona compatibilità con i comandi di matlab.
- E' minimalista.

Primi passi

- Variabili vengono assegnate senza dichiarare il tipo.

```
a=42  
s= 'test'
```

- Operatori di base (+ - * / \ ^)

```
5/2    ans = 2.5000
```

ans è una variabile di sistema

- funzioni (help elfun)

```
sqrt(3), sin(pi), cos(0)
```

pi è una variabile di sistema

- display & clear variabili

```
disp(ans), disp('ciao')
```

display il valore di ans , “ciao”

```
who, whos
```

mostra tutte le variabili definite

```
clear a
```

elimina la variabile a

```
clear
```

elimina tutte le variabili

- Frecche su/giù richiamano gli ultimi comandi

Primi passi

- **Variabili & Operatori**

`a=5*(2/3)+3^2` viene mostrato il risultato
`a=2/4 + 4\2;` non viene mostrato il risultato
`a` viene mostrato il valore di `a`

- **Funzioni elementari**

`abs(-1), sqrt(2)`
`tan(0), cos(0), acos(1) ...`
`exp(2), log(1), log10(1) ...`

- **Arrotondamento**

`round(2.3), round(2.5)` 2 3
`floor(5.7), floor(-1.2)` 5 -2 al valore più piccolo
`ceil(1.1), ceil(-2.7)` 2 -2 al valore più grande
`fix(1.7), fix(-2,7)` 1 -2 allo zero

- **Numeri complessi**

`(2+3i) * (1i)` -3+2i
`norm(1+1i)` 1.4142

Vettori

- **Vettore riga e Vettore Colonna:**

$[1, 2, 3] \rightarrow (1 \ 2 \ 3)$ alternativa: $[1 \ 2 \ 3]$
 $[1; 2; 3] \rightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

- **Inizializzare un vettore:** `[start : end]` o `[start : step : end]`

`[1:5]` $(1 \ 2 \ 3 \ 4 \ 5)$
`[1:3:10]` $(1 \ 4 \ 7 \ 10)$
`[0.5: -0.5 : -0.6]` $(0.5 \ 0 \ -0.5)$

- **Vettori lineari e log-spaced con N elementi**

`v=linspace(0,1,4)` $(0 \ 1/3 \ 2/3 \ 1)$
`w=logspace(1,4,4)` $(10 \ 100 \ 1000 \ 10000)$

- **Imposta o visualizza un elemento**

`v(4)` 1 elemento #4
`w(3)=99` 10 100 99 10000 cambia l'elemento #3

- **Trasposizione Vettore**

$[1, 2, 3]' \Rightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

- **Numero di elementi**

`length(v)`

Vettori

$$\mathbf{v} = [1, 4, 9, 16, 25]$$

- Indirizzo elementi del vettore

$$\begin{array}{ll} \mathbf{v}([1, 3, 5]) & (1 \ 9 \ 25) \quad \text{elementi 1,3,5} \\ \mathbf{v}([2:4]) & (4 \ 9 \ 16) \quad \text{elementi 2,3,4} \end{array}$$

- Imposta elementi di un vettore

$$\begin{array}{ll} \mathbf{v}([1, 3, 5]) = 0 & (0 \ 4 \ 0 \ 16 \ 0) \quad \text{imposta elementi 1,3,5 a 0} \\ \mathbf{v}([1, 3, 5]) = [1 \ 2 \ 3] & (1 \ 4 \ 2 \ 16 \ 3) \quad \text{imposta el. 1,3,5 a [1 2 3]} \\ \mathbf{v}([1, 3, 5]) = [1 \ 2] & \text{ERROR} \quad \text{le dimensioni non coincidono} \end{array}$$

- Cerca e sostituisci elementi

$$\begin{array}{ll} \mathbf{p} = \text{find}(\mathbf{v} > 8) & (3 \ 4 \ 5) \quad \text{indici elementi con valori } > 8 \\ \mathbf{v}(\mathbf{p}) & (9 \ 16 \ 25) \quad \text{elementi } > 8 \\ \mathbf{v}(\mathbf{p}) = 0 & (1 \ 4 \ 0 \ 0 \ 0) \quad \text{imposta elementi } > 8 \text{ a } 0 \end{array}$$

- Aritmetica con i vettori:

$$\begin{array}{ll} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 4 & 5 & 6 \end{bmatrix} & (5 \ 7 \ 9) \\ \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} + 4 & (5 \ 6 \ 7) \\ \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} * 2 & (2 \ 4 \ 6) \end{array}$$

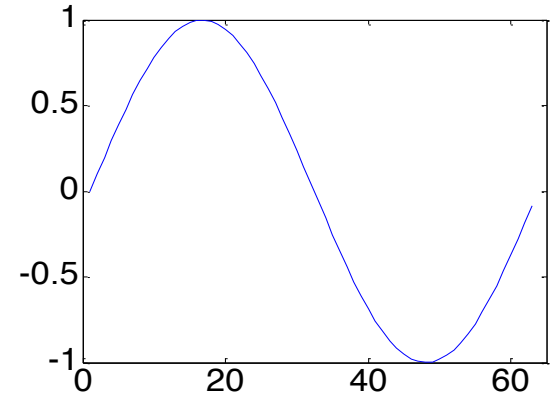
- Operazioni "Element-wise":

$$\begin{array}{ll} \mathbf{v}.*\mathbf{v}, \quad 1./\mathbf{w}, \quad \mathbf{v}.^{[1 \ 2]}.*[3 \ 4] & (3 \ 8) \\ [1 \ 2 \ 3] .^2 & (1 \ 4 \ 9) \end{array}$$

Plot di vettori

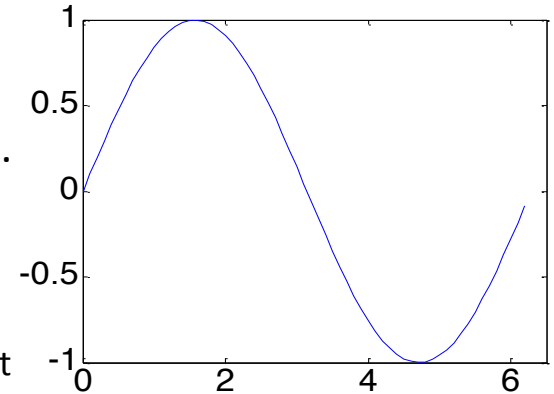
- **y-plots**

```
y=sin([0:0.1:2*pi])  
plot(y)
```



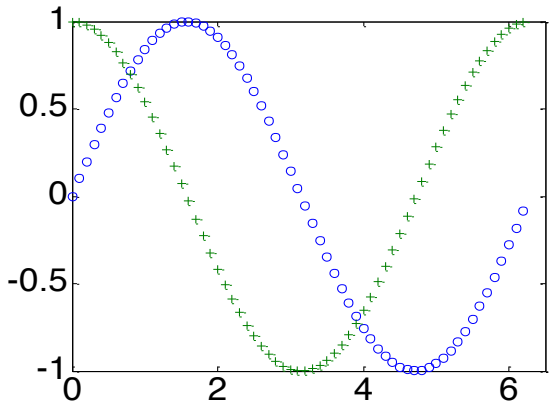
- **xy-plots**

```
x=[0:0.1:2*pi]  
y=sin(x)  
plot(x,y)      y potrebbe essere anche una matrice.
```



- **Formati di plot**

```
y1=sin(x)  
y2=cos(x)  
plot(x,y1,'o',x,y2,'+')  guarda help plot
```



- **Plot speciali**

```
semilogx, semilogy, loglog  guarda help
```

- **Axes, labels**

```
axis, xlabel, ylabel      guarda help
```

Matrici

- Inizializzare una matrice:

```
[1, 2, 3; 4, 5, 6; 7, 8, 9]
```

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

- Farlo come con i vettori

```
M=[1:3:9; 11:3:19; 21:3:29]
```

```
1 4 7
11 14 17
21 24 27
```

- Accesso agli elementi della matrice: `M(row, col)`

```
M(2, 3)
```

```
17
```

accesso elemento singolo

```
M(1, :)
```

```
1
```

```
4
```

```
7
```

accesso vettore riga

```
M(:, 2)
```

```
4
```

accesso vettore colonna

```
14
```

```
24
```

```
M(2:3, 1:2)
```

- Matrici speciali

```
zeros(2)
```

```
ones(2)
```

```
ones(2, 3)
```

```
eye(2)
```

```
0 0
0 0
```

```
1 1
1 1
```

```
1 1 1
1 1 1
```

```
1 0
0 1
```

- Trova e sostituisci elementi della matrice con proprietà comuni

```
find (M>15)
```

```
M(find(M>15)) = -1
```

Matrici

- Trasposta : A'
- Inversa: $\text{inv}(A)$
- Determinante: $\text{det}(A)$
- Matrice diagonale : $A = \text{diag}(v)$
- Dimensione: $\text{size}(A)$

- Somma : $A+B$ Stesso ordine di grandezza
- Prodotto: $A*B$ il numero di colonne di A e il numero di righe in B devono essere uguali
- Prodotto elemento per elemento: $A .* B$

- Divisione left/right :
 $A/B = ((B')^{-1} * A')'$ il calcolo è fatto senza calcolare $A^{-1} \Rightarrow$ molto efficiente
 $A \setminus B = A^{-1} * B$

Comparazioni ed esecuzione condizionale

- **Operatori relazionali**

`<`, `>`, `<=`, `>=` minore, maggiore, minore uguale, maggiore uguale
`==`, `~=` uguale, non uguale.

- **Operatori Logici**

`&`, `|`, `~` and, or, not

`(2>5) & (1>0)` `~(1>1)`
`(0>0) | (1>=1)` `~(10<=9) & ~(1>0)`

- **Simple if: `if (condition) ; ... ; end`**

```
x = 6;  
if (rem(x,3)==0) rem ritorna il resto della divisione  
    disp('il numero è divisibile per 3')  
end
```

- **if-else: `if (condition) ; ... ; else ... ; end`**

```
if (rem(x,2)==0)  
    disp('il numero è pari')  
else  
    disp('il numero è dispari')  
end
```

Cicli

- Ciclo for: effettua un ciclo tra tutti gli elementi di un vettore

for variable=vector; ... ; **end**

```
fib([1 2])=1;           %calcola i primi 10 numeri della serie di Fibonacci
for i=3:10
    fib(i)=fib(i-1)+fib(i-2)
end
```

- Ciclo while: cicla fintanto (while) condition è vera

while (condition); ... ; **end**

```
fib([1 2])=1;           %calcola i primi 10 numeri della serie di Fibonacci
i=3;
while (i<=10)
    fib(i)=fib(i-1)+fib(i-2)
    i=i+1;
end
```