



### Assegnazioni

variabile  $a=3$  (con output) oppure  $a=7.3$ ; (senza output)  
 espressione  $b=(a+1)*(2-a^2)$   
 vettore riga con 3 elementi  $v=[1 \ 1.7 \ -12.78]$  oppure  $v=[1,1.7,-12.78]$   
 vettore riga con incrementi unitari  $v=j:k$  equivale a  $v=[j, j+1, \dots, k]$  ad esempio  $v=1:15$   
 vettore riga con incrementi arbitrari  $v=j:d:k$  equivale a  $v=[j, j+d, \dots, j+m*d]$  ad es.  $v=.4:.2:1$   
 $v=\text{linspace}(x1,x2 [ ,n])$  vettore riga con  $n$  elementi equispaziati da  $x1$  a  $x2$   
 $v=\text{logspace}(x1,x2 [ ,n])$  vettore riga con  $n$  elementi equispaziati in logaritmo, da  $10^{x1}$  a  $10^{x2}$   
 matrice  $M=[1,4,4;2,-7,6]$

### Operatori

Trasposta complesso coniugato:  $'$   
 $M(:,1)$  è la prima colonna della matrice  $M$   
 $M(:,\$)$  è l'ultima colonna della matrice  $M$   
 $M(1:3,1)$  corrisponde alle prime 3 righe e prima colonna della matrice  $M$   
 $M(3:6,3:7)$  alcune righe e alcune colonne della matrice  $M$   
 elevamento a potenza:  $^$   
 operatori algebrici:  $+$ ,  $-$ ,  $*$ ,  $/$   
 divisione di un numero per un vettore (rispettare gli spazi):  $1 ./ v$

### Valori predefiniti

pi-greco:  $\%pi$   
 unità immaginaria:  $\%i$   
 costante di Neper:  $\%e$   
 infinito:  $\%inf$

### Comandi e funzioni interne

uscita da scilab: `exit`  
 aiuto sul comando  $x$ : `help x`  
 editor di testi: `scipad`  
 resto della divisione intera: `modulo(x,y)`  
 radice quadrata di  $x$ : `sqrt(x)`  
 logaritmo naturale di  $x$ : `log(x)`  
 logaritmo decimale di  $x$ : `log10(x)`  
 valore assoluto di  $x$ : `abs(x)`  
 funzioni goniometriche: `sin(x)`, `cos(x)`, `tan(x)`, `cotg(x)`  
 funzioni goniometriche inverse: `asin(x)`, `acos(x)`, `atan(x)`  
 parte reale e immaginaria di  $x$ : `real(x)`, `imag(x)`  
 dimensioni della matrice  $m$ : `size(m)`  
 massimo elemento del vettore  $v$ : `max(v)`  
 disegno 2D: `plot2d(x,y)`, `plot2d(x,y,-1)`, `plot2d()`, `plot2d(x,y,style=[-9,-8,-7,-6,-5,-4,-3,-2,-1,0])`  
 griglia su disegno: `xgrid`  
 etichette su disegno: `xtitle`  
 controllo dei limiti degli assi: `isoview(xmin,xmax,ymin,ymax)`  
 impostazioni sul disegno attuale: `xset`, `xset('default')`  
 istogramma in 20 classi: `histplot(20,rand(1,10000,'normal'))`  
 numero casuale tra 0 e 1: `rand` oppure `rand()`  
 elenco delle variabili definite: `who`  
 elimina la variabile  $x$ : `clear x`  
 directory attuale: `pwd`  
 contenuto della directory attuale: `dir`  
 cambio directory: `cd c:\...`  
 commento in un file di testo: `//`

### Calcolo simbolico

monomio unitario di grado 1: `x=poly(0,'x')`  
 polinomio: `x=poly(0,'x');y=2*x^2-x+6`  
 zeri di un polinomio: `x=poly(0,'x');y=x^2+x-2;roots(y)`  
 Diagrammi di Bode, Nyquist, Luogo delle radici: `bode()`, `nyquist()`, `evans()`

### Funzioni esterne

esempio di definizione da linea di comando:  
`--> function [x]=segno(y)`  
`--> if y>0 then, x=1; else, x=-1; end`  
`--> endfunction`  
 oppure copiando le stesse righe nel file di testo sorgente.sci e richiamando: `exec('sorgente.sci')`

```
if modulo(num,2) == 0
    disp('The Number is
    Even');
elseif modulo(num,2) ~= 0
    disp('The Number is
    Odd');
else
    disp('Invalid Number');
end
```

```
i=2
for j = 1:3,
    if i == j then
        a(i,j) = 2;
    elseif abs(i-j) == 1
    then
        a(i,j) = -1;
    else a(i,j) = 0;
    end,
end
```

```
mat=rand(1,10,'normal');
binary=zeros(size(mat));
count = 1;
while(count<=length(mat))
    if mat(count) >= 0
        binary(count)=1;
    else
        binary(count)=-1;
    end
    count =count+1;
end
```

```
while %t
    n=round(10*rand(1,1))
    select n
    case 0 then
        disp(0)
    case 1 then
        disp(1)
    else
        break
    end
end
```