

Algebra lineare con Matlab:

- come creare matrici e vettori;
- prodotti matrice-vettore;
- soluzione di sistemi lineari (e verifica delle soluzioni con `norm`);
- autovettori e autovalori (con verifica);
- altri calcoli: determinanti, ranghi. . . ;
- funzioni più complesse per creare matrici (`zeros` e cicli `for`. . .);
- alcuni calcoli simbolici con `syms`;
- derivate, integrali simbolici.

Creare matrici e vettori

```
>> A = [1 2 3; 4 5 6]
```

```
A =
```

```
    1  2  3
```

```
    4  5  6
```

```
>> v = [7 8 9]
```

```
v =
```

```
    7  8  9
```

```
>> w = [7;8;9]
```

```
w =
```

```
    7
```

```
    8
```

```
    9
```

Prodotti matrice-vettore

```
>> A = [1 2; 3 4]
A =
     1 2
     3 4
>> B = [1 0 1; 0 1 0]
B =
     1 0 1
     0 1 0
>> A*B
ans =
     1 2 1
     3 4 3
>> B*A
Error using *
Inner matrix dimensions must agree.
```

Prodotti matrice-vettore

```
>> v = [7 8 9]
```

```
v =
```

```
7 8 9
```

```
>> w = [7;8;9]
```

```
w =
```

```
7
```

```
8
```

```
9
```

```
>> v*w, w*v
```

```
ans =
```

```
194
```

```
ans =
```

```
49 56 63
```

```
56 64 72
```

```
63 72 81
```

Sistemi lineari

```
>> A = [1 2; 2 5]
```

```
A =  
    1 2  
    2 5
```

```
>> b = [-1; 2]
```

```
b =  
   -1  
    2
```

```
>> x = A \ b
```

```
x =  
   -9  
    4
```

```
>> A*x - b
```

```
ans =  
    0  
    0
```

Sistemi lineari

```
>> A = [1 2; 3 4];  
>> b = [-1; 2];  
>> x = A \ b  
x =  
    4.0000  
   -2.5000  
>> A*x - b  
ans =  
    1.0e-15 *  
    0.4441  
         0
```

Le operazioni sono approssimate a 16 cifre significative—è normale trovare errori di circa una parte su 10^{16} .

Sistemi lineari

```
>> A = [1 2; 3 4]
A =
     1 2
     3 4
>> b = [-1; 2] * 12345
b =
    -12345
     24690
>> x = A \ b
x =
  1.0e+04 *
     4.9380
    -3.0862
>> A*x - b
ans =
  1.0e-10 *
     0.0728
```

Sistemi lineari

```
>> A = [1 1; 1 1+10^(-14)]
A =
    1.0000    1.0000
    1.0000    1.0000
>> b = A*[1;1]
b =
    2.0000
    2.0000
>> x = A \ b
x =
    1.0222
    0.9778
```

Gli errori possono essere anche molto maggiori — c'è tutta una branca della matematica che studia cosa succede con queste approssimazioni.

Autovalori e autovettori

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

```
>> [V,D] = eig(A)
```

```
V =
```

```
-0.2320 -0.7858 0.4082  
-0.5253 -0.0868 -0.8165  
-0.8187 0.6123 0.4082
```

```
D =
```

```
16.1168 0 0  
0 -1.1168 0  
0 0 -0.0000
```

```
>> v = V(1:3, 1)
```

```
v =
```

```
-0.2320  
-0.5253  
-0.8187
```

```
>> lambda = D(1,1)
```

```
lambda =
```

Autovalori e autovettori

```
>> lambda = D(1,1)
lambda =
    16.1168
>> A*v - lambda*v
ans =
    1.0e-14 *
    0.3109
   -0.1776
   -0.1776
>> v = V(1:3, 2);
>> lambda = D(2, 2);
>> A*v - lambda*v
ans =
    1.0e-14 *
    0.0111
   -0.1901
   -0.3775
```

Altre operazioni di algebra lineare

```
v =  
    0  
    1  
    2  
>> norm(v)  
ans =  
    2.2361  
>> A = [0 1 2; 3 4 5; 6 7 8]  
>> det(A)  
ans =  
    0  
>> rank(A)  
ans =  
    2
```

Costruire matrici

```
>> A = zeros(2,3)
A =
    0 0 0
    0 0 0
>> B = ones(4,1); % matrice / vettore di tutti uni
>> C = zeros(4, 4);
>> R = rand(5, 6); % entrate casuali in [0,1]
>> for i = 1:3
    C(i, i+1) = 1;
end
>> C
C =
    0 1 0 0
    0 0 1 0
    0 0 0 1
    0 0 0 0
```

Calcoli simbolici

Serve aver installato il *symbolic math toolbox* (va usato di nuovo il Matlab installer se non l'avete installato in precedenza)

```
>> syms x y % dichiara x, y come variabili
```

```
>> A = [1+x 3; 1-x y]
```

```
A =
```

```
[ x + 1, 3]
```

```
[ 1 - x, y]
```

```
>> lambda = eig(A)
```

```
lambda =
```

```
 x/2 + y/2 - (x^2 - 2*x*y - 10*x + y^2 - 2*y + 13)^(1/2)/2 + 1
```

```
 x/2 + y/2 + (x^2 - 2*x*y - 10*x + y^2 - 2*y + 13)^(1/2)/2 + 1
```

```
>> solve(lambda(1)==0, x)
```

```
ans =
```

```
-(y - 3)/(y + 3)
```

Calcoli simbolici

```
>> syms x y
>> sols = solve([x+y-1, x-y])
sols =
  struct with fields:

    x: [1x1 sym]
    y: [1x1 sym]
>> sols.x
ans =
1/2
>> sols.y
ans =
1/2
```

Calcoli simbolici

```
>> sols = solve(A*[x;x] - [2;3]); sols.x
```

```
ans =
```

```
- 6^(1/2) - 2
```

```
6^(1/2) - 2
```

```
sols = solve(A*[x;y] - [2;1]); sols.x
```

```
ans =
```

```
3/2 - (3*root(z^4 + 6*z^3 + z^2 - 12*z + 7, z, 1))/2 - root(z
```

```
3/2 - (3*root(z^4 + 6*z^3 + z^2 - 12*z + 7, z, 2))/2 - root(z
```

```
3/2 - (3*root(z^4 + 6*z^3 + z^2 - 12*z + 7, z, 3))/2 - root(z
```

```
3/2 - (3*root(z^4 + 6*z^3 + z^2 - 12*z + 7, z, 4))/2 - root(z
```

Qualche volta neppure Matlab riesce a trovare una forma esplicita della soluzione...

Calcoli simbolici

```
>> diff(x^3+sin(x))
ans =
cos(x) + 3*x^2
>> diff(x^3+sin(y), y)
ans =
cos(y)
>> int(x^3)
ans =
x^4/4
>> int(x^3, 1,2)
ans =
15/4
>> int(exp(x)/x^2*sin(x)) % non trova una formula chiusa
ans =
int((exp(x)*sin(x))/x^2, x)
```