Comandi che agiscono su un vettore

Comando	Azione
a = sum(x)	genera lo scalare $a = \Sigma_i x_i$
a = prod(x)	genera lo scalare $a = \Pi_i x_i$
a = max(x)	genera lo scalare $a = \max_i x_i$
a = min(x)	genera lo scalare $a = \min_i x_i$
a = norm(x)	genera lo scalare $a = \|\|x\|\|_2$
a = norm(x,1)	genera lo scalare $a = \|\|x\|\|_1$
a = norm(x,inf)	genera lo scalare $a = \|\|x\|\|_\infty$
A = diag(x)	genera la matrice $A = (a_{ij})_{i,j=1,\dots,n}$ con $a_{ii}=x_i$

NORME DI VETTORI

Un'applicazione $\mathbb{R}^n \to \mathbb{R}^+$ chiamata **norma**, indicata usualmente con $\|x\|$, quando verifica le seguenti condizioni

1. $\|\mathbf{x}\| \geq 0$, per ogni $\mathbf{x} \in \mathbb{R}^n$
2. $\|\mathbf{x}\| = 0$ se e solo se $\mathbf{x} = 0$
3. $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$, per ogni $\mathbf{x} \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$
4. Per ogni $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^n$, vale la disuguaglianza triangolare $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\|+\|\mathbf{y}\|$

Esempi: norma p, per $1 \leq p < \infty$

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

$$\|\mathbf{x}\|_\infty = \max_{1\leq i\leq n} |x_i|$$

Esercizio: $\mathbf{x} = [1, -2]$. calcolare le norme

$\|\mathbf{x}\|_1 =$ 3 $\qquad \|\mathbf{x}\|_2 = \sqrt{5} \qquad \|\mathbf{x}\|_\infty =$ 2

NORME DI MATRICI

Norme indotte

1. $\|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$
2. $\|A\|_2 = \sqrt{\rho(A^T A)}$
3. $\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$

Norma di Frobenius

$A \in \mathbb{R}^{n \times m}$

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2}$$

Comandi che agiscono su matrici

Comando	Azione
`a = norm(A)`	genera lo scalare $a = \|\|A\|\|_2$
`a = norm(A,1)`	genera lo scalare $a = \|\|A\|\|_1$
`a = norm(A,inf)`	genera lo scalare $a = \|\|A\|\|_\infty$
`x = sum(A)`	genera il vettore riga $(x_j)_{j=1,\ldots,n}$ con $x_j = \Sigma_i\, a_{ij}$
`x = max(A)`	genera il vettore riga $(x_j)_{j=1,\ldots,n}$ con $x_j = \max_i a_{ij}$
`x = min(A)`	genera il vettore riga $(x_j)_{j=1,\ldots,n}$ con $x_j = \min_i a_{ij}$
`x = diag(A)`	genera il vettore colonna $(x_i)_{i=1,\ldots,n}$ con $x_i=a_{ii}$
`B = abs(A)`	genera la matrice $B=(b_{ij})_{i,j=1,\ldots,n}$ con $b_{ij}= \| a_{ij}\|$
`B = tril(A)`	genera la matrice $B=(b_{ij})_{i,j=1,\ldots,n}$ con $b_{ij}= a_{ij}$ per $i=1,\ldots,n,\ 1\leq j < i$
`B = triu(A)`	genera la matrice $B=(b_{ij})_{i,j=1,\ldots,n}$ con $b_{ij}= a_{ij}$ per $i=1,\ldots,n,\ i \leq j < n$

- sum(*A*): Calcola la somma degli elementi di ogni colonna dell'array *A* e restituisce un vettore riga che contiene le somme risultanti;

```
>> X=[0 1 2
      3 4 5];
>> sum(X)
ans=
      [3 5 7]
>> sum(X,1)
ans=
      [3 5 7]
>> sum(X,2)
ans=
      [ 3
     12]
```

- prod(*A*): Calcola il prodotto degli elementi di ogni colonna dell'array *A* e restituisce un vettore riga che contiene i prodotti risultanti;
- sort(*A*): Dispone gli elementi delle colonne dell'array *A* in ordine crescente e restituisce un array della stessa dimensione di *A*.

Operazioni su array: Operatori algebrici

- E' possibile effettuare tutte le usuali operazioni elementari ben definite tra matrici,vettori e scalari:
  - prodotto righe per colonne;
  - prodotto di uno scalare per un array;
  - somma di array.
  - a*B,B*a
    se a è uno scalare e B un array di qualsiasi dimensione, fornisce l'array C avente le stesse dimensioni di B e ottenuto moltiplicando ogni elemento di B per a: $c_{ij} = a \cdot b_{ij}$
- Un discorso analogo vale per l'operazione B/a
- A*B Se il numero di colonne di A è uguale al numero di righe di B, fornisce il prodotto righe per colonne C = AB
- A^m Se A è una matrice quadrata e m un intero, fornisce il prodotto righe per colonne di A per se stessa m volte

Operazioni elemento per elemento

- Ognuno degli operatori *, /,^ può essere usato applicandolo anche "elemento per elemento" ad array delle stesse dimensioni usando la cosiddetta sintassi del punto

- Se $\odot$ è uno degli operatori sopra citati e A e B sono due array con uguali dimensioni, con la sintassi

$$A\ .\!\odot B$$

si crea un terzo array, con le stesse dimensioni di A e di B, il cui elemento di posizione ij è dato dall'espressione $a_{ij} \odot b_{ij}$.

Operazioni in MatLab

Simbolo	Operazione	Forma	Esempio
’	Trasposizione	A’	[1 2]’=[1 2]
+	Somma array-scalare	A + b	[5,2] + 1 = [6,3]
-	Sottrazione array-scalare	A - b	[5,2] - 1 = [4,1]
+	Somma di array	A + B	[5,2] + [-1,2] = [4,4]
-	Sottrazione di array	A - B	[5,2] - [-1,2] = [6,0]
.*	Moltiplicazione di array	A .* B	[2,3] .* [3,3] = [6,9]
./	Divisione a destra o diretta di array	A ./ B	[2,6] ./ [3,7] = [2/3,6/7]
.\	Divisione a sinistra o inversa di array	A .\ B	[2,7] .\ [3,4] = [2\3,7\4]
.^	Elevamento a potenza di array	A .^ B	[2,3] .^ 2 = [2^2,3^2] 2 .^ [2,3] = [2^2,2^3] [2,3] .^ [3,5] = [2^3,3^5]

Date le seguenti matrici:

```
>> A=[2 3;4 5];
>> B=[1 0;0 1];
```

sono ben definite le operazioni

```
>> C = A*B, D = A.*B
C =
        2 3
        4 5
D =
        2 0
        0 5
```

- Certe operazioni (come la somma e la sottrazione) sono in realtà già applicate elemento per elemento.

- Anche l'operatore "^" può essere applicato elemento per elemento ad un array eventualmente rettangolare usando ancora la sintassi del punto:

```
>> P = A.^2
P =
        4  9
        16 25
```

si genera una matrice con le stesse dimensioni di A il cui elemento di posizione ij è dato dall'espressione a_{ij}^m;

```
>> B=[1 2 3; 1 2 3]
>> B^2
??? Error using ==> mpower
Matrix must be square.
```

Funzioni dell'algebra lineare

d = det(A)	Calcola il determinante di una matrice quadrata *A*;
B = inv(A)	Calcola la matrice inversa di *A*, se *A* è matrice quadrata non singolare;
H = hilb(n)	Costruisce la matrice di Hilbert di ordine n, dove H(i,j) = 1/(i + j – 1);
V = vander(v)	Costruisce la matrice di Vandermonde di ordine n = length(v), dove V(i,j)= v(i)^(n-j)
[M,D] = eig(A)	Calcola gli autovalori ed autovettori di una matrice quadrata.
A\b	Calcola la soluzione del sistema lineare $Ax = b$ con il metodo di eliminazione di Gauss;

det(A)

```
>> A = [1 2;3 4];
>> det(A)
ans=
        -2
```

inv(A)

```
>> A = [1 2;3 4];
>> inv(A)
ans=
        -2    1
        1,5 -0,5
```

```
>> B = [0 0;0 1];
>> inv(B)

Warning:
Matrix is singular to working precision

ans=
        Inf Inf
        Inf Inf
```

H=hilb(n) $\quad H(i,j) = 1/(i + j - 1);$

$$H = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} \end{pmatrix}$$

Calcolo dei determinanti principali della matrice di Hilbert di ordine 4

```
>> format rat
>> A = hilb(4)
A =
        1     1/2 1/3 1/4
        1/2 1/3 1/4 1/5
        1/3 1/4 1/5 1/6
        1/4 1/5 1/6 1/7

>> m1 = A(1,1), dm1 = det(m1)
m1 =
        1
dm1 =
        1
```

```
>> m2 = A(1:2,1:2),dm2 = det(m2)
m2 =
        1     1/2
        1/2   1/3
dm2 =
        1/12

>> m3 = A(1:3,1:3),dm3 = det(m3)
m3 =
        1     1/2   1/3
        1/2   1/3   1/4
        1/3   1/4   1/5
dm3 =
        1/2160

>> det(A)
ans =
        1/6048000
```