

Risoluzione di sistemi lineari

- ▶ Il comando `\`.

```
A=[1 2 3; -3 2 0; 1 1 -1]
```

```
b=[1 2 -1]'
```

```
x= A\b
```

```
A*x
```

Usa il metodo di Gauss

- ▶ Il comando `inv`.

```
inv(A)*b
```

Più costoso.

Esempio di matrice mal condizionata

La matrice di Hilbert: $a_{i,j} = \frac{1}{i+j-1}$.

```
A=hilb(10)
```

```
b=sum(A)'
```

```
x=A \ b
```

```
cond(A)
```

Determinante, norma, autovalori...

```
A=[1 2 3; -1 0 1; -2 -3 4]
```

```
det(A)
```

```
norm(A)
```

```
norm(A,1)
```

```
norm(A, inf)
```

```
eig(A)
```

Esercizio: Calcolare il raggio spettrale della matrice A.

Esercizio

Risolvere il sistema lineare

$$\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix} \mathbf{x} = 0.04 \begin{bmatrix} -2 \\ -2 \\ -2 \\ -2 \end{bmatrix}$$

Questo problema è il caso $N = 5$ di una famiglia di sistemi lineari di dimensione $(N - 1) \times (N - 1)$:

$$N = 5 \quad h = 1/N \quad 0.04 = h^2$$

Posso fare una **funzione** che dato N risolva il corrispondente sistema lineare.

Funzioni

Una funzione è scritta in un file con estensione `.m`, ad esempio `nome.m`.

- ▶ La prima riga del file deve essere

```
function [out1,out2,...,outn]=nome(in1,in2,...,inm)
```

`out1, ..., outn` sono le variabili in uscita, i risultati.

`in1, ..., inm` son le variabili in ingresso, gli argomenti.

La funzione si deve chiamare come il file dove è scritta.

- ▶ Una funzione viene chiamata dopo il prompt di Matlab ma bisogna dare (fra parentesi tonde) i suoi argomenti.
- ▶ Quando si invoca una funzione Matlab crea un'area di lavoro locale.

Le istruzioni contenute in una funzione non possono riferirsi a variabili dichiarate nell'area di lavoro di base a meno che non rientrino fra i parametri in ingresso.

Tutte le variabili usate in una funzione vanno perse a fine esecuzione a meno che non siano tra i parametri in uscita.

```
function [x]=esempio(N)
h=1/N;
b= -2*h^2*ones(N-1,1);
B=diag(ones(N-2,1),1);
A=2*eye(N-1)-B-B';
x=A\b;
```

Esercizio

$$\begin{cases} -u''(x) = f(x) & x \in (0, 1) \\ u(0) = u(1) = 0 \end{cases}$$

$$h = \frac{1}{N} \quad x_i = i * h \quad i = 0, \dots, N$$

$$u''(x_i) \approx \frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1}))}{h^2}$$

Scrivere una funzione di Matlab che approssimi la soluzione del problema

$$\begin{cases} -u''(x) = f(x) & x \in (0, 1) \\ u(0) = u(1) = 0 \end{cases}$$

usando questo metodo.

Le variabili in input sono N ed f . Come dare f ?

Possiamo usare una stringa \rightsquigarrow eval

```
function [x,u]=corda(N,f)
h=1/N;
x=[h:h:1-h];
B=diag(ones(1,N-2),1);
A=2*eye(N-1)-B-B';
b=h^2*eval(f); % f stringa
u=A\b';
u=[0 u' 0];
x=[0 x 1];
```

```
>> [t,y]=corda(10,'-x.^2')
```

Oppure una funzione di Matlab \rightsquigarrow feval

```
function [x,u]=corda(N,f)
h=1/N;
x=[h:h:1-h];
B=diag(ones(1,N-2),1);
A=2*eye(N-1)-B-B';
b=h^2*feval(f,x); % f funzione
u=A\b';
u=[0 u' 0];
x=[0 x 1];
```

```
function y=dataf(x)
y=-x.^2;
```

```
>> [t,y]=corda(10,@dataf)
```