Metodi numerici con elementi di Programmazione

A.A. 2013-2014

Introduzione al MatLab
III parte

Docente: Vittoria Bruni

Email: vittoria.bruni@sbai.uniroma1.it

Ufficio: Via A. Scarpa,
Pal. B, I piano, Stanza n. 16
Tel. 06 49766648

Ricevimento: Giovedì 14.00-15.00

Il materiale didattico è disponibile sul sito http://ingaero.uniroma1.it/ nella pagina dedicata al corso Metodi Numerici con elementi di Programmazione

Per consultazione: Getting Started with MatLab – The mathworks
www.mathworks.com

Alcune funzioni

sum(A)	se A è un vettore calcola la somma degli elementi. Se A è una matrice? (usare lo help)
prod(A)	se A è un vettore calcola il prodotto degli elementi. Se A è una matrice?
[m,n]=size(A)	assegna ad m ed n i valori delle dimensioni di A
D=det(A)	calcola il determinante di A
B=inv(A)	calcola la matrice inversa di A
A=zeros(m,n)	crea una matrice di dimensione m x n con tutti gli elementi nulli
A=ones(m,n)	crea una matrice di dimensione m x n con tutti gli elementi uguali a 1
D=eye(m,n)	la matrice D ha tutti 1 in posizione diagonale (se D è quadrata, D=eye(n) è la matrice identità)

Alcune funzioni

R=rand(m,n) matrice di dimensione mxn i cui elementi sono numeri casuali uniformemente distribuiti e compresi tra 0 ed 1

R= randn(m,n) matrice di dimensione mxn i cui elementi sono numeri casuali con distribuzione normale di media nulla e varianza 1

Esercizi

Si scrivano i comandi Matlab per ognuna delle seguenti operazioni

1. Creare un vettore riga x di 5 elementi equispaziati nell'intervallo [2,3]
2. Aggiungere 1 al secondo elemento di x
3. Creare un vettore riga y avente la stessa dimensione di x i cui elementi sono i numeri interi pari maggiori o uguali a 4
4. Creare una matrice A la cui prima riga sia il vettore x, la seconda riga sia un vettore i cui elementi sono tutti uguali a 1 e la terza riga sia uguale a y
5. Definire il vettore riga z i cui elementi siano uguali alla media degli elementi di ogni colonna di A
6. Definire il vettore colonna w i cui elementi siano uguali alla media degli elementi di ogni riga di A
7. E' possibile usare una sola istruzione per creare il vettore a tale che $a(n) = (-1)^n \pi^{2n} / (2n)!$, con $0 \leq n \leq 50$?

Esercizi

Si scrivano i comandi Matlab per ognuna delle seguenti operazioni

8. Dato il vettore t = [0:0.01:1], calcolare le seguenti funzioni:
   $e^{(t+t2)}$, $\cos^{-1}(t)$, $\tan^2(t)-1$

9. Si considerino le matrici A = [1 2; 4 -1] e B = [4 -2; -6 3]
   - Calcolare C=A+B, D = A-B, F=AB e G=BA
   -Definire la matrice H tale che $H(i,j)= B(i,j)+A(i,j)B(i,j)^{1/3}$
   -Se A e B non sono matrici singolari, calcolarne le matrici inverse
   -Sottrarre alla seconda riga di A la sua prima riga moltiplicata per 3

10. Dati i vettori x=[0 2 3]', y=[2 -3 4] e le matrici A = [0 8; -4 3; -2 5], B=[9 7 6; 8 5 -1], stabilire quali dei seguenti comandi possono essere eseguiti correttamente

x+y	x+y'	x*y	y*x
A*y	A'*x	B'.*A	y*A*B*x

Operatori relazionali

Si applicano a matrici o vettori aventi la stessa dimensione e sono operazioni elemento per elemento

C = A < B	minore
C = A <= B	minore o uguale
C = A > B	maggiore
C = A >= B	maggiore o uguale
C = A == B	uguale
C = A ~= B	diverso

La matrice di output C ha la stessa dimensione di A e B

C(i,j) = 1 se la relazione è verificata,

C(i,j) = 0 se la relazione non è verificata

Operatori relazionali

Esempio:

```
>> A=[1 2 3; 4 5 6; 7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
>> B = [0 2 4; 3 5 6; 3 4 9]
B =
     0     2     4
     3     5     6
     3     4     9
>> A==B
ans =
     0     1     0
     0     1     1
     0     0     1
```

Verificare anche gli altri operatori

Operatori logici

Si applicano a matrici o vettori aventi la stessa dimensione e sono operazioni elemento per elemento

C = A & B and

C = A | B or

C = A ~ B not

C = xor(A, B) or esclusivo

all(v) restituisce 1 se tutti gli elementi del vettore v sono diversi da zero, 0 altrimenti. Se v è una matrice?

any(v) restituisce 1 se almeno un elemento del vettore v è diverso da zero, 0 altrimenti. Se v è una matrice?

Operatori logici

Esempio: Siano x=[0 5 3 7] ed y=[0 2 8 7]

```
>> x>y
ans =
     0     1     0     0
>> x>4
ans =
     0     1     0     1
>> m=(x>y)&(x>4)
m =
     0     1     0     0
>> p= x|y
p =
     0     1     1     1
```

Nota: qualsiasi intero ≠0 vale vero

Esercizi

1. Generare una matrice A di dimensione 30x30 di numeri casuali con media nulla e varianza 1:

   a) usando gli operatori relazionali ma non le funzioni predefinite in matlab, determinare il vettore v contenente il valore assoluto degli elementi di A
   b) determinare il vettore w

   $w(k) = 0$ se $|A(k)| \leq 1.2$

   $w(k) = A(k)$ se $|A(k)| > 1.2$

Alcune funzioni utili

fliplr(v): inverte l'ordine degli indici di un vettore riga se v è una matrice, inverte l'ordine degli indici di colonna

flipud(v): inverte l'ordine degli indici di un vettore colonna se v è una matrice, inverte l'ordine degli indici di riga

Esempio:

```
>> v = [3 1 –4 0];
>> a = fliplr(v)
a =
     0    -4     1     3
>> b = flipud(v')
b =
     3
     1
    -4
     0
```

Alcune funzioni utili

Esempio:

```
>> A = [3 1 –4; 10 0 3; -1 2 0]
A =
     3     1    -4
    10     0     3
    -1     2     0
>> a = fliplr(A)
a =
    -4     1     3
     3     0    10
     0     2    -1
>> b = flipud(A)
b =
    -1     2     0
    10     0     3
     3     1    -4
```

Alcune funzioni utili

diff(v) calcola la differenza tra componenti successive di un vettore [v(2)-v(1), v(3)-v(2), …., v(end)-v(end-1)]. La dimensione del vettore risultante è length(v)-1

Se v è una matrice, la matrice risultante contiene le differenze successive tra le righe di v. La dimensione della matrice di output è (m-1)xn se la dimensione di v è mxn.

diff(v,k) calcola le differenze di ordine k tra le componenti di v.

Esempio:

```
>> v = [9 1 7 4 3 0 9 -1];
>>diff(v)
ans =
    -8     6    -3    -1    -3     9   -10
>>diff(v,2)
ans =
    14    -9     2    -2    12   -19
>>diff(v,3)
ans =
   -23    11    -4    14   -31
```

Alcune funzioni utili

Esempio:

```
>> A = [9 1 7; 4 3 0 ;9 –1 5]
A =
     9     1     7
     4     3     0
     9    -1     5

>> diff(A)
ans =
    -5     2    -7
     5    -4     5

>> diff(A,2)
ans =
    10    -6    12
```

Alcune funzioni utili

indici = find(condizione su v)

restituisce un vettore costituito dagli indici corrispondenti alle componenti di v che verificano la condizione.

Esempio: ind = find(v == 0) Restituisce le posizioni delle componenti nulle di v

Se si omette la condizione, cioè ind = find(v), ind contiene le posizioni delle componenti di v diverse da zero

Esempio:

```
>> V = [6   1   0.3   -5   10   -4   8   3];
>> ind = find(V <= 0);
>> ind
ind =
     4      6
>> V(ind)
ans =
    -5     -4
```

Esercizi

Dato x = [7 6 1 2 0 -1 4 3 -2 0], scrivere i comandi Matlab necessari per eseguire le seguenti operazioni

1. Porre a zero gli elementi negativi di x
2. Generare il vettore y contenente gli elementi di x più grandi di 3
3. Aggiungere 3 agli elementi pari di x
4. Detto m il valore della media degli elementi di x:
   - porre gli elementi di x inferiori a m uguali a zero
   - porre gli elementi di x superiori alla media uguali alla loro differenza con m

Esercizi

Usando opportunamente le funzioni Matlab find e diff, scrivere la sequenza di istruzioni necessaria per individuare e stampare opportunamente gli estremi dell'intervallo di separazione della radice delle seguenti equazioni non lineari:

- $f(\lambda) = e^{\lambda} + 0.435\,(e^{\lambda} - 1) / \lambda - 1.564$ in $I=[0.05,0.15]$
- $f(x) = \log(x+1)+(x+2)^{1/2}-1$ in $I=[-0.9,0.5]$
- $f(x) = x^3 - 10x^2 + 5$ in $I=[0.6,0.8]$
- $f(x) = (x-1)^2-3\log(x)$ in $I=[0.5,2]$
- $f(x) = \sin(x)+x-1$ in $I=[0,\pi]$
- $f(x) = \log(x)+x^2-x$ in $I=[0.7,2.3]$

I valori degli estremi devono assegnati al vettore intervallo. Il messaggio di stampa deve distinguere i due estremi

Esempio

```
>> x=(0.6:.02:.8)';
>> f=@(x)[x.^3-10*x.^2+5];
>> y=f(x);
>> pos = find(abs(diff(sign(y))));
>> intervallo(1) = x(pos);
>> intervallo(2) = x(pos+1);
>> fprintf('l''estremo inferiore e'' %6.5f\nl''estremo
superiore e'' %6.5f\n',intervallo)
l'estremo inferiore e' 0.72000
l'estremo superiore e' 0.74000
```

Esempio

Infatti

```
>> [(1:length(x))' x y sign(y)]
ans =
    1.0000    0.6000    1.6160    1.0000
    2.0000    0.6200    1.3943    1.0000
    3.0000    0.6400    1.1661    1.0000
    4.0000    0.6600    0.9315    1.0000
    5.0000    0.6800    0.6904    1.0000
    6.0000    0.7000    0.4430    1.0000
    7.0000    0.7200    0.1892    1.0000
    8.0000    0.7400   -0.0708   -1.0000
    9.0000    0.7600   -0.3370   -1.0000
   10.0000    0.7800   -0.6094   -1.0000
   11.0000    0.8000   -0.8880   -1.0000
```

Esempio

```
>> [(1:length(x)-1)'  diff(sign(y))]
ans =
      1      0
      2      0
      3      0
      4      0
      5      0
      6      0
      7     -2
      8      0
      9      0
     10      0
>> pos=find(abs(diff(sign(y))))
pos =
      7
```

equivalente a

```
>> pos=find((diff(sign(y)))~=0)
pos =
     7
```

Esercizio

Rappresentare graficamente la funzione $f(x; y) = x\, e^{-x^2-y^2}$ sul dominio $D = [-2, 2]x[-2, 2]$.

Soluzione

- Definire una griglia sul dominio D (matrice di punti):

- Definire la funzione f(x) e valutarla nei punti della griglia:

- Tracciare il grafico di z

Grafici

- Definire una griglia sul dominio D:

```
>> [x,y]=meshgrid(-2:.1:2,-2:.1:2);
```

La funzione [x,y]=meshgrid(xvett,yvett) genera una griglia di punti le cui coordinate sono contenute nelle matrici x e y. I valori degli elementi di x e y sono dati dai vettori di punti equidistanti xvett e yvett

Grafici

```
x =

   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000


y =

   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000
   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000
   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000
   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000
         0         0         0         0         0         0         0         0         0
    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000
    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000
```

Grafici

- Definire f(x) e valutarla nei punti della griglia:

```
>> f='x.*exp(-x.^2-y.^2)';
>> z=eval(f);
```

Sono le stesse istruzioni usare per definire funzioni di una sola variabile
L'istruzione è equivalente a

```
>> f=@(x,y)[x.*exp(-x.^2-y.^2)];
>> z=f(x,y);
```

oppure

```
>> z=x.*exp(-x.^2-y.^2);
```

Grafici

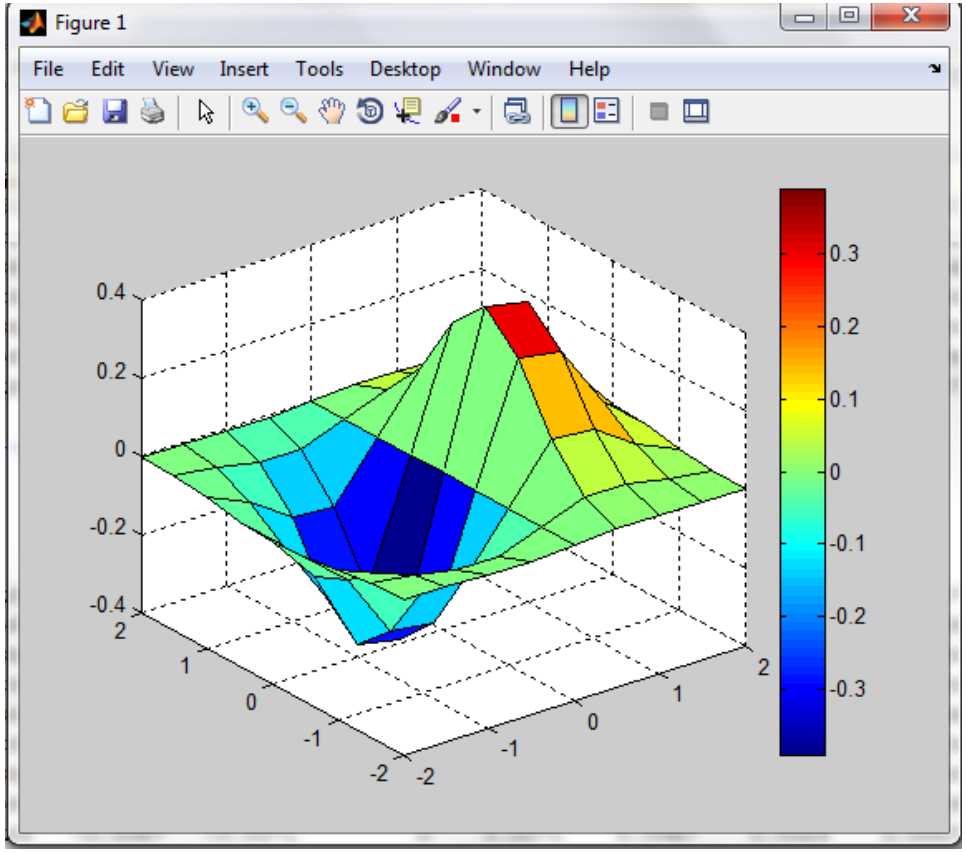
z =

-0.0007	-0.0029	-0.0067	-0.0071	0	0.0071	0.0067	0.0029	0.0007
-0.0039	-0.0167	-0.0388	-0.0410	0	0.0410	0.0388	0.0167	0.0039
-0.0135	-0.0582	-0.1353	-0.1433	0	0.1433	0.1353	0.0582	0.0135
-0.0285	-0.1231	-0.2865	-0.3033	0	0.3033	0.2865	0.1231	0.0285
-0.0366	-0.1581	-0.3679	-0.3894	0	0.3894	0.3679	0.1581	0.0366
-0.0285	-0.1231	-0.2865	-0.3033	0	0.3033	0.2865	0.1231	0.0285
-0.0135	-0.0582	-0.1353	-0.1433	0	0.1433	0.1353	0.0582	0.0135
-0.0039	-0.0167	-0.0388	-0.0410	0	0.0410	0.0388	0.0167	0.0039
-0.0007	-0.0029	-0.0067	-0.0071	0	0.0071	0.0067	0.0029	0.0007

Grafici

- Tracciare il grafico di z

```
>> surf(x,y,z);
>> colorbar
```

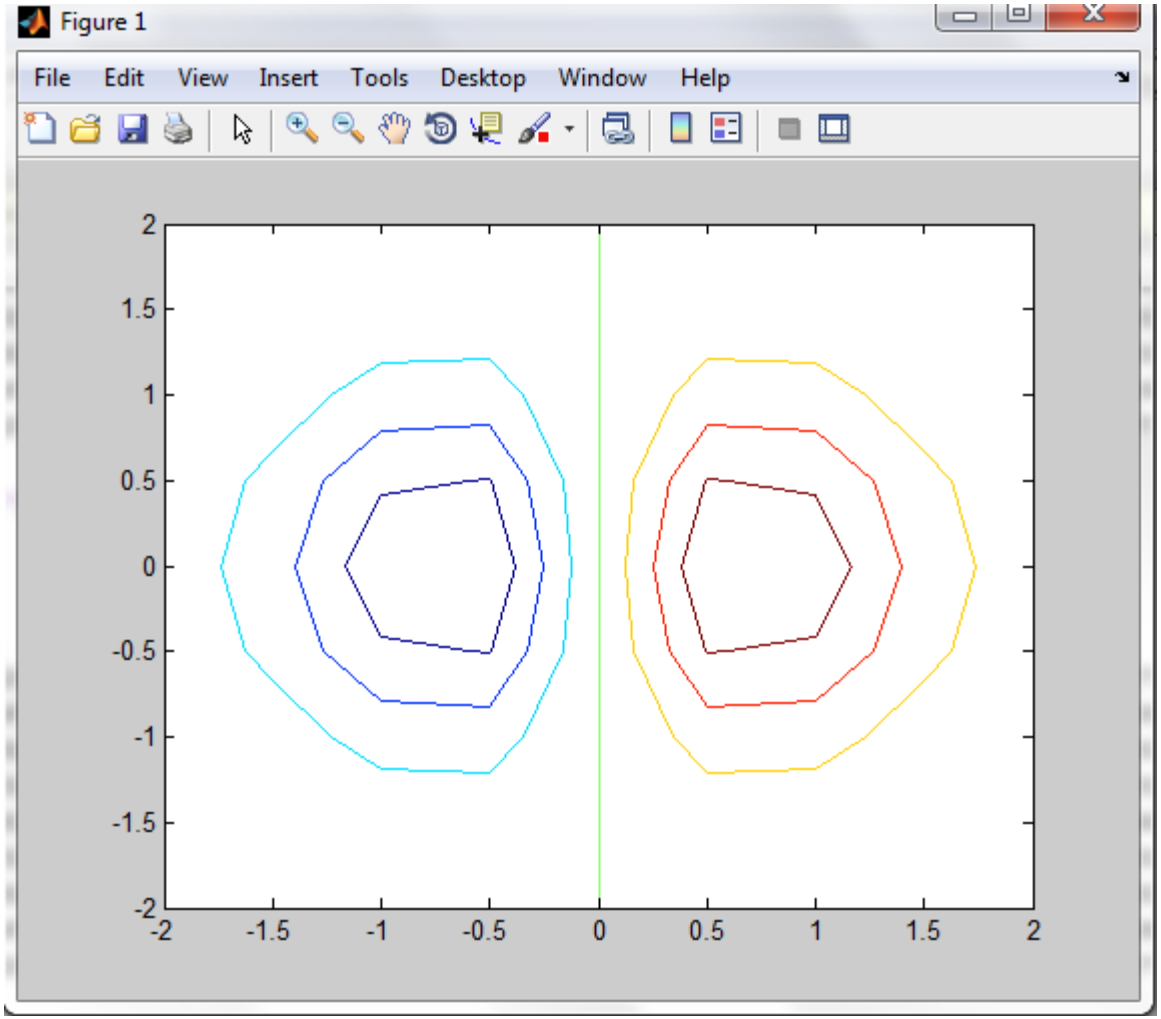


Il comando surf(x,y,z) disegna la superficie z=f(x,y) valutata nei punti della griglia le cui coordinate sono date dagli elementi corrispondenti delle matrici x e y

Colorbar mostra la barra dei colori che distinguono i valori assunti da z

Grafici

Il comando contour(x,y,z) disegna le linee di livello, cioè le curve dei punti in cui la superficie assume un valore fissato costante

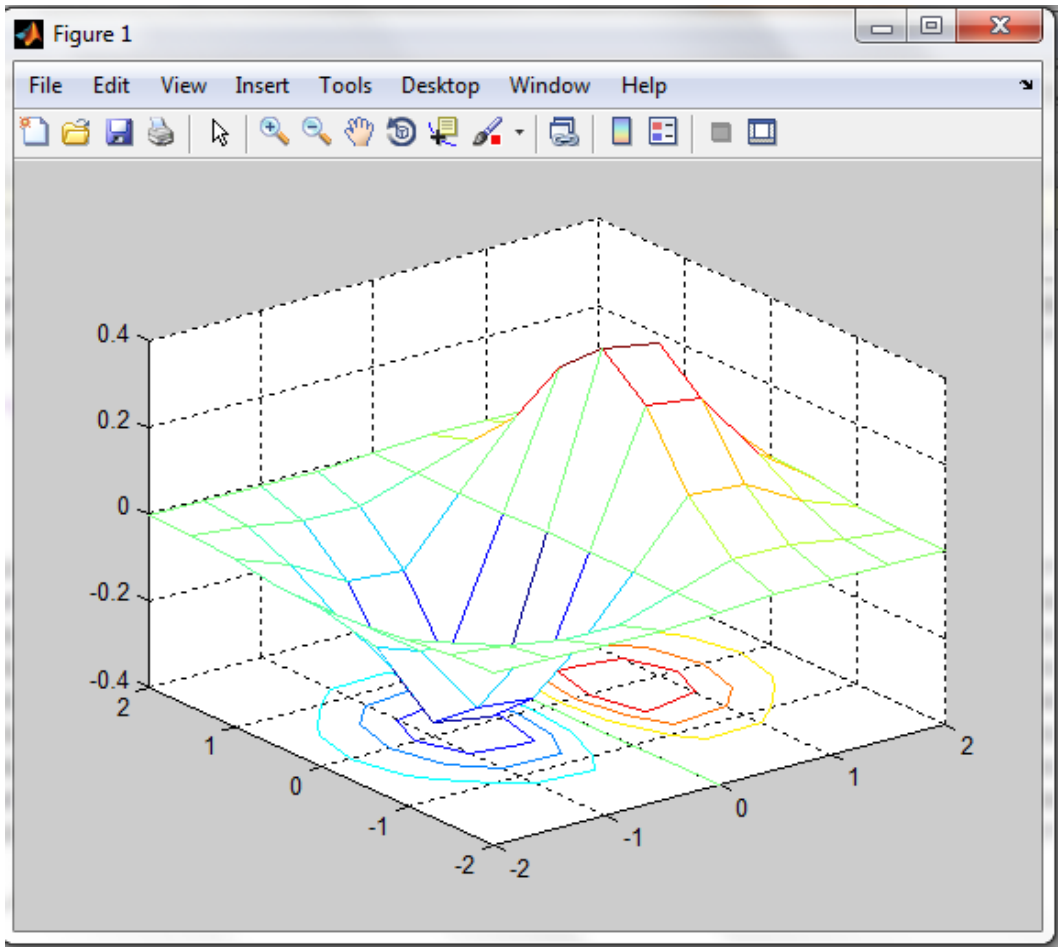


Grafici

Il comando mesh(x,y,z) si può usare in alternativa al comando surf

I comandi meshc(x,y,z) e surfc(x,y,z) disegnano contemporaneamente la superficie e le linee di livello

```
>> meshc(x,y,z)
```



Grafici

Il grafico si può completare aggiungendo etichette e titoli, esattamente come accade per il caso 1D

Altre funzioni utili sono

view	cambia l'orientamento del grafico (punto di vista)
colormap	cambia il colore del grafico
shading	cambia l'ombreggiatura del grafico

Usare lo help per stabilire quando usare le funzioni
pcolor e plot3

Grafici: esempi

Per disegnare il grafico della seguente funzione $f(x,y) = x^2+y^2-2x-3$ sul dominio $D = [-3,3]$ è necessario

- definire la griglia (matrice) dei punti $\mathrm{X} = [\mathrm{x},\mathrm{y}]$ su cui è definita la funzione f

```
[x,y]=meshgrid(-3:.1:3,-3:.1:3);
```

dove le variabili di output x e y sono matrici

- definire la funzione di cui disegnare il grafico

```
f=@(x,y)[x.^2+y.^2-2*x-3];
```

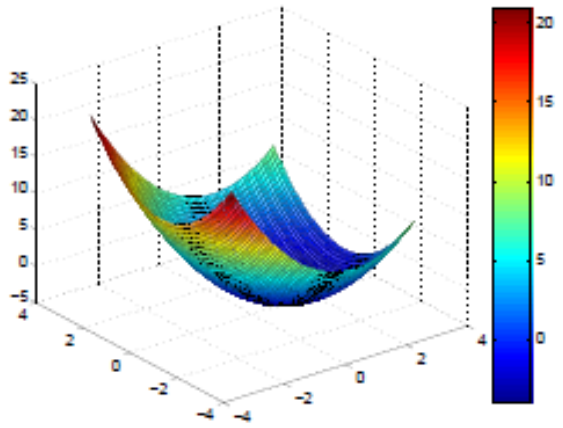
- valutare la funzione nei punti della griglia

```
z = f(x,y);
```

Grafici: esempi

- disegnare la funzione valutata nei punti $X = [x, y]$

```
surf(x,y,z); colorbar
```



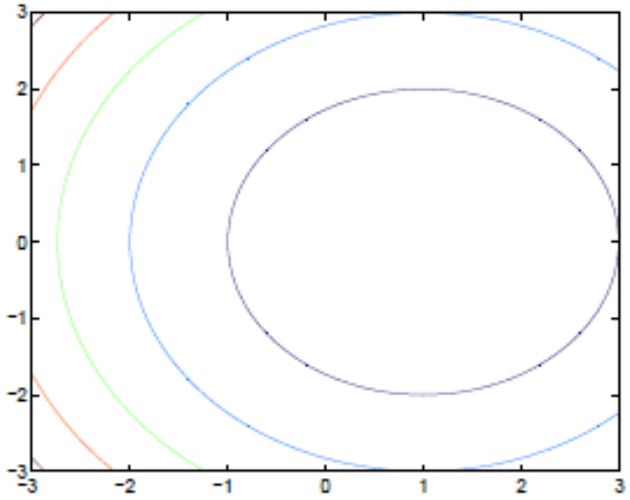
I punti della superficie risultano colorati diversamente secondo il valore assunto. La colorbar riporta la scala dei colori

In alternativa si può usare il comando

```
mesh(x,y,z)
```

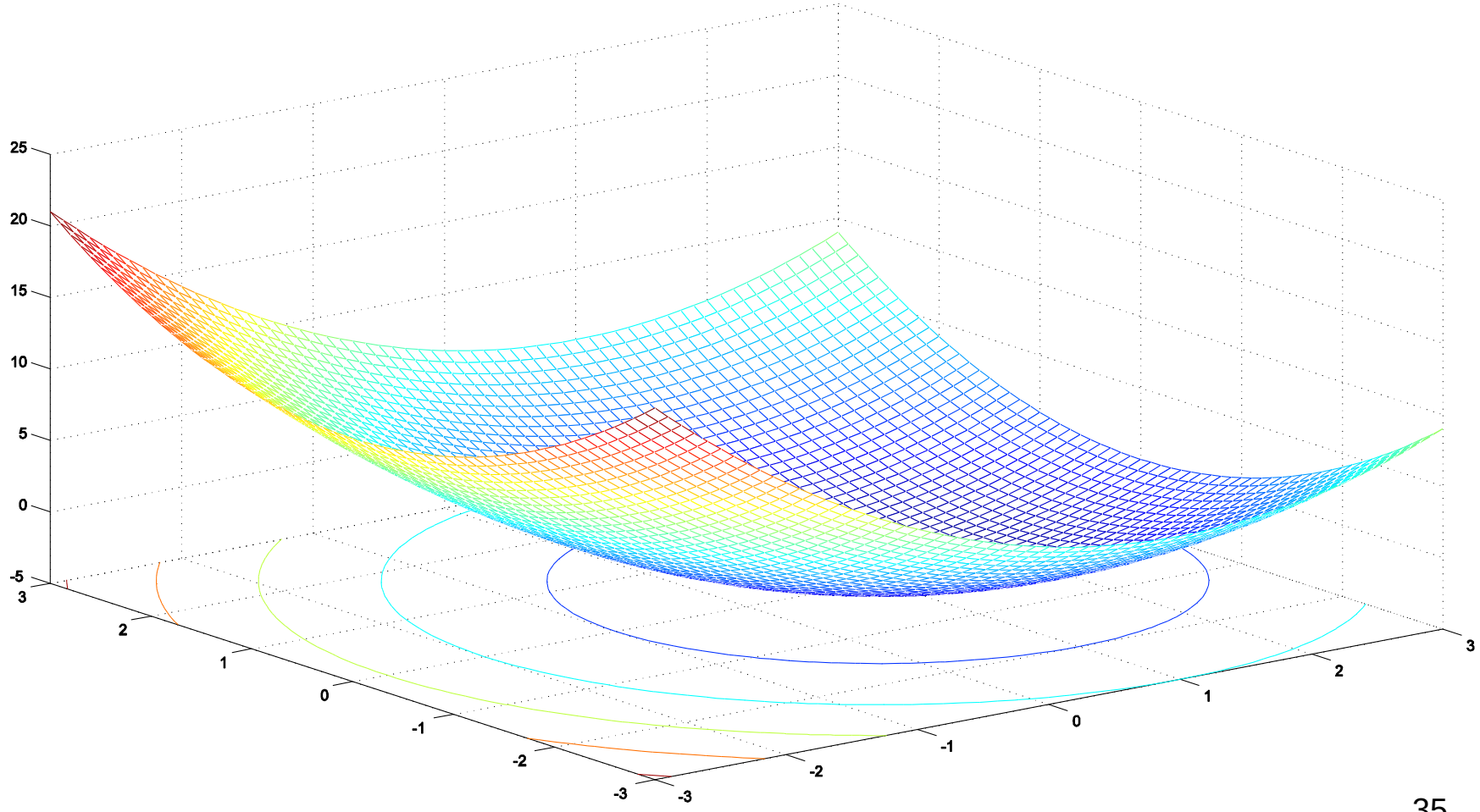
Grafici: esempi

Il comando **contour**(x,y,z) disegna le linee di livello, cioè le curve dei punti in cui la superficie assume un valore fissato costante



Grafici: esempi

meshc(x,y,z) *oppure* surfc(x,y,z) disegnano contemporaneamente la superficie e le linee di livello



Localizzazione delle radici: rappresentazione grafica

Esempio: $f(x; y) = x^2+y^2-3$, $g(x,y) = xy-1$

Per localizzare le radici del sistema si disegnano le superfici $z1 = f(x, y)$ e $z2 = g(x, y)$ e le curve di livello $f(x, y) = 0$ e $g(x, y) = 0$.

```
[x,y]=meshgrid(-3:.1:3);
z1=x.^2+y.^2-3;
z2=x.*y-1;
figure,
subplot(2,2,1), mesh(x,y,z1), title('z1=f(x,y)')
hold on,
contour(x,y,z1,[0 0],'r','linewidth',2);  % linee di livello f(x,y) = 0
subplot(2,2,2), mesh(x,y,z2), title('z2=g(x,y)')
hold on
contour(x,y,z2,[0 0],'g','linewidth',2); % linee di livello g(x,y) = 0
subplot(2,2,3), contour(x,y,z1,[0 0],'r','linewidth',2);
hold on, contour(x,y,z2,[0 0],'g','linewidth',2);
xlabel('x'), ylabel('y'), legend('z1=0','z2=0',2)
title('Intersezioni')
```

Esempio: $f(x; y) = x^2+y^2-3$, $g(x,y) = xy-1$