Calcolo Numerico

(A.A. 2013-2014)

Esercitazione n. 9

Metodo del punto unito, Metodo di Newton per sistemi

11-04-2014

Esercizio 1.25

L. Gori, M.L. Lo Cascio, F. Pitolli, *Esercizi di Calcolo Numerico*, **II ed.**

Dimostrare che il seguente **procedimento iterativo**

$$x_{n+1} = x_n + e^{1-x_n} - 1$$

converge qualunque sia il punto iniziale $x_0 > 1$ e calcolarne il limite;

- stabilire se la successione è monotona per ogni $x_0 > 1$;
- studiare il comportamento di $\{x_{n+1}\}$ se $x_0 = 1/2$;
- stabilire l'ordine di convergenza del procedimento iterativo.

Verificare i risultati scrivendo una **function** Matlab con i seguenti parametri di input e output

Input	**Ouput**
ϕ funzione d'iterazione	**xn** approssimazione della radice
x0 punto iniziale	**err** errore di approssimazione
ε precisione richiesta	**niter** il numero di iterazioni
maxiter numero massimo di iterazioni	

Se il numero massimo di iterazioni viene eseguito senza aver raggiunto la precisione richiesta, stampi il messaggio **La soluzione prodotta non ha raggiunto la precisione richiesta in maxiter**.

Convergenza: condizione sufficiente

Teorema 2. Se φ è **derivabile** in $I = [a, b]$ e

$i)$ $\varphi(I) \subseteq I$

$ii)$ $\exists\ k \in (0, 1)$ tale che $|\varphi'(x)| \leq k,\ \ x \in I$

$\Rightarrow \alpha)$ esiste un **unico punto unito** $\xi \in I$ di $\varphi(\xi)$

$\beta)$ la successione $x_n = \varphi(x_{n-1})$ è **convergente** a ξ per ogni **approssimazione iniziale** $x_0 \in I$

Soluzione La funzione di iterazione ϕ è $\phi(x) = x + e^{1-x} - 1$.

ϕ è una funzione **continua** in $\mathbf{R}$ e **derivabile**, inoltre si osserva che

$$\phi(x) = x \Leftrightarrow e^{1-x} - 1 = 0 \Leftrightarrow 1 - x = 0 \Leftrightarrow x = 1.$$

Quindi $\mathbf{x = 1}$ è il punto unito di ϕ. Inoltre,

$$\lim_{x \to -\infty} \phi(x) = +\infty \quad \text{e} \quad \lim_{x \to +\infty} \phi(x) = +\infty$$

Studiamo ora la **derivata** della funzione $\phi(x) = x + e^{1-x} - 1$

$$\phi'(x) = 1 - e^{1-x} \geq 0 \Leftrightarrow e^{1-x} \leq 1 \Leftrightarrow 1 - x \geq 0 \Leftrightarrow x \geq 1,$$

$\phi'(x)$ è una funzione **continua** in $\mathbf{R}$, **monotona decrescente** per $x < 1$ e **monotona crescente** per $x > 1$. Poiché ϕ' cambia segno in $x = 1$, $\phi(x)$ ha un estremo relativo in corrispondenza di questo punto.

$$\phi''(x) = e^{1-x} > 0 \quad \forall\, x \in \mathbf{R},$$

quindi $x = 1$ è un **punto di minimo relativo** per $\phi(x)$.

Poichè $\phi''(x)$ ha segno costante nel dominio di esistenza di ϕ, ne deduciamo che $x = 1$ è l'**unica radice** dell'equazione non lineare considerata.

Proprietà della successione delle approssimazioni

Dal **Teorema di Lagrange** si ha

$$e_n = \xi - x_n = \varphi(\xi) - \varphi(x_{n-1}) =$$

$$= \varphi'(t_n)(\xi - x_{n-1}) = \varphi'(t_n)e_{n-1} \quad t_n \in [x_{n-1}, \xi]$$

- Se $0 \leq \varphi'(x) < 1$ per $x \in I$ la successione $\{x_n = \varphi(x_{n-1})\}$, $n = 1, 2, ...$, è **monotona crescente** (se $e_0 > 0$) o **descrescente** (se $e_0 < 0$) $\Rightarrow$ le approssimazioni sono per **difetto** (se $\xi > x_0$) o per **eccesso** (se $\xi < x_0$)
- Se $-1 < \varphi'(x) \leq 0$ per $x \in I$ la successione $\{x_n = \varphi(x_{n-1})\}$, $n = 1, 2, ...$, **non è monotona** $\Rightarrow$ le approssimazioni sono **alternativamente** per **difetto** e per **eccesso**

Poichè ϕ' cambia segno proprio in corrispondenza della radice $x = 1$, l'eventuale **convergenza monotona** del metodo potrà essere garantita solo in intervalli $I \subseteq [1, +\infty)$ mentre il metodo può convergere, eventualmente non in modo monotono, in intervalli $I \subseteq [1 - \delta, +\infty)$, con $\delta \in \mathbf{R}^+$.

Per stabilire tali convergenze è necessario determinare l'intervallo I tale che $\phi(I) \subseteq I$ e $|\phi'(x)| \leq k, \quad k \in (0, 1) \quad \forall x \in I$.

Si osserva che

$$|\phi'(x)| = |1 - e^{1-x}| < 1 \Leftrightarrow \begin{cases} \forall\, x & \text{se} \quad x > 1 \\ x > 1 - \log(2) & \text{se} \quad x < 1 \end{cases}$$

Ne segue che $0 \leq \delta \leq \log(2)$.

Inoltre, poichè $\phi(x)$ è una funzione monotona crescente per $x > 1$, scelto un punto $\overline{x} > 1$, risulta

$$\phi(\overline{x}) \leq \overline{x} \Leftrightarrow e^{1-\overline{x}} \leq 1 \Leftrightarrow x \geq 1.$$

Quindi, $\forall\, I = [1, \overline{x}]$ accade che

$$\phi(I) \subseteq I \quad \text{e} \quad |\phi'(x)| < 1$$

da cui deduciamo che **il metodo converge in modo monotono** $\forall\, x_0 > 1$.

Poiché $x = 1$ è l'unico punto unito di ϕ, si può scegliere un intorno del punto $x = 1$ in cui il metodo converge per ogni scelta del punto iniziale x_0.

In questo caso, la condizione $\phi(I) \subseteq I$ implica

$$e^{1-(1-\delta)} - 1 \geq 0 \Leftrightarrow e^{\delta} \geq 1 \Leftrightarrow \delta \geq 0,$$

che, confrontata con la condizione sulla derivata, da

$$0 \leq \delta \leq \log(2).$$

Da quanto detto sopra, poichè

$$1 - \log(2) < \frac{1}{2} < 1,$$

scegliendo $x_0 = 1/2$ il metodo converge in intervalli $I = [1 - \delta, \overline{x}]$, con $\delta < 0.5$ e $\overline{x} \geq 1$ (in particolare, $\overline{x} \geq \phi(1 - \delta)$).

Poichè risulta

$$\phi(1) = 1 \quad \phi'(1) = 0 \quad \text{e} \quad \phi''(1) \neq 0,$$

l'ordine di convergenza del metodo è $\mathbf{p = 2}$.

OSS: Verificare che il procedimento iterativo corrisponde al metodo di Newton applicato alla funzione $f(x) = e^{x-1} - 1$

Function punto_unito

```
function [xn,err,n_iter] = punto_unito(f,x0,eps,maxiter)
% [xn,err,n_iter] = punto_unito(f,x0,eps,maxiter)
% cerca il punto unito della funzione f  con precisione eps
% scegliendo x0 come punto iniziale.
% Il procedimento iterativo si interrompe se si raggiunge il
% numero massimo di iterazioni maxiter
%
% INPUT
% f = espressione della funzione di cui si vuole cercare il
%     punto unito
% x0 = punto iniziale
% eps = limite superiore dell'errore da usare come criterio
%       di arresto
% maxiter = numero massimo di iterazioni da eseguire
%           indipendentemente dalla convergenza
%
```

```
% OUTPUT
% xn = approssimazione del punto unito
% err = |xn-x(n-1)|
% n_iter = numero di iterazioni eseguite

format long;

xn = x0;

%inizializzazione dei parametri
n_iter = 1; err = 10;
% si cerca la soluzione con una certa precisione oppure quella raggiunta
% dopo un numero fissato di iterazioni
figure, hold on
```

```
while (err>eps) & (n_iter<maxiter)
    xv = xn;
    xn = f(xn);
    n_iter = n_iter+1;
    err = abs(xn-xv);
    fprintf('%3d\t %15.15f\t  %6.15f\f\n',[n_iter  xn  err])
    plot(n_iter,f(xn),'*')
end
if n_iter >= maxiter
   fprintf('La soluzione prodotta non ha raggiunto la...
    precisione richiesta in %5d iterazioni',maxiter)
end
```

Esempio

$I = [1 - log(2),\ \phi(1 - log(2))] \qquad x_0 = 1/2 \qquad \epsilon = 0.5 \cdot 10^{-5}$

Metodo del punto unito

k	x_k	ϵ_k
2	1.148721270700128	0.648721270700128
3	1.010530563626045	0.138190707074083
4	1.000055252269219	0.010475311356827
5	1.000000001526379	0.000055250742840
6	1.000000000000000	0.000000001526379

Metodo di Newton

k	x_k	ϵ_k
1	1.148721270700128	0.648721270700128
2	1.010530563626045	0.138190707074083
3	1.000055252269219	0.010475311356827
4	1.000000001526379	0.000055250742840
5	1.000000000000000	0.000000001526379

Esercizi

Modificare la funzione **punto_unito** utilizzando nel criterio di arresto il comando **break**.

Funzione punto unito (break)

```
function [xn,err,n_iter] = punto_unito_break(f,x0,eps,maxiter)
% [xn,err,n_iter] = punto_unito(f,x0,eps,maxiter)
% cerca il punto unito della funzione f  con precisione eps
% scegliendo x0
% come punto iniziale. Il procedimento iterativo si interrompe
% se si raggiunge il numero
% massimo di iterazioni maxiter
%
% INPUT
% f = espressione della funzione di cui si vuole cercare
%     il punto unito
% x0 = punto iniziale
% eps = limite superiore dell'errore da usare come
%       criterio di arresto
% maxiter = numero massimo di iterazioni da eseguire indipendentemente
%           dalla convergenza
%
```

```
% OUTPUT
% xn = approssimazione del punto unito
% err = |xn-x(n-1)|
% n_iter = numero di iterazioni eseguite

format long;

xn = x0;

%inizializzazione dei parametri
n_iter = 1; err = 10;

% si cerca la soluzione con una certa precisione oppure quella raggiunta
% dopo un numero fissato di iterazioni
figure, hold on
```

```
for n_iter = 2:maxiter
    xv = xn;
    xn = f(xn);
    err = abs(xn-xv);
    fprintf('%3d\t %15.15f\t  %6.15f\f\n',[n_iter  xn  err])
    plot(n_iter,xn,'*')

    if err<=eps
        break
    end

end

if n_iter >= maxiter
   fprintf('La soluzione prodotta non ha raggiunto la precisione ....
   richiesta in %5d iterazioni',maxiter)
end
```

Esercizio

Stabilire se la funzione

$$\phi = \frac{a_g T^2}{2\pi} \tanh\left(\frac{2\pi d}{L}\right)$$

genera un procedimento iterativo convergente alla radice dell'equazione

$$L - \frac{a_g T^2}{2\pi} \tanh\left(\frac{2\pi d}{L}\right) = 0$$

Metodo di Newton per sistemi: $n = 2$

Per $n = 2$ si ha: $\begin{cases} f(X) = f(x, y) = 0 \\ g(X) = g(x, y) = 0 \end{cases}$

Formula di Taylor di punto iniziale $X^{(k)} = (x_k, y_k)$:

$$\Downarrow$$

$$\begin{cases} f(X) = f(X^{(k)}) + f_x(X^{(k)})(x - x_k) + f_y(X^{(k)})(y - y_k) + R_1 = 0 \\ g(X) = g(X^{(k)}) + g_x(X^{(k)})(x - x_k) + g_y(X^{(k)})(y - y_k) + R_2 = 0 \end{cases}$$

dove $R_1 = R_1(X, X^{(k)})$, $R_2 = R_2(X, X^{(k)})$ rappresentano il **resto**.

La **soluzione approssimata** del sistema non lineare è la soluzione del **sistema lineare** che si ottiene trascurando il resto nello sviluppo precedente.

$$\begin{cases} f_x(X^{(k)})(x_{k+1} - x_k) + f_y(X^{(k)})(y_{k+1} - y_k) = -f(X^{(k)}) \\ g_x(X^{(k)})(x_{k+1} - x_k) + g_y(X^{(k)})(y_{k+1} - y_k) = -g(X^{(k)}) \end{cases}$$

Metodo di Newton per sistemi: $n = 2$

Il sistema lineare può essere riscritto nella seguente forma matriciale

$$\Downarrow$$

$$J_F^{(k)}(X^{(k+1)} - X^{(k)}) = -F(X^{(k)})$$

dove $J_F^{(k)} := J_F(X^{(k)}) = \begin{bmatrix} f_x(X^{(k)}) & f_y(X^{(k)}) \\ g_x(X^{(k)}) & g_y(X^{(k)}) \end{bmatrix}$

Il **sistema lineare** ammette soluzione se

$$|J_F^{(k)}| = \det J_F^{(k)} \neq 0$$

La soluzione è

$$\begin{cases} x_{k+1} = x_k - \dfrac{1}{|J_F^{(k)}|}\left[f(X^k)\, g_y(X^{(k)}) - g(X^{(k)})\, f_y(X^{(k)}) \right] \\ y_{k+1} = y_k - \dfrac{1}{|J_F^{(k)}|}\left[g(X^k)\, f_x(X^{(k)}) - f(X^{(k)})\, g_x(X^{(k)}) \right] \end{cases}$$

Esempio

Determinare i punti di intersezione tra il **cerchio** $x^2+y^2=3$ e l'**iperbole** $xy=1$ con 5 decimali esatti.

Soluzione

Si devono trovare i punti che annullano simultaneamente le funzioni $f(x,y)=x^2+y^2-3$ e $g(x,y)=xy-1$.

Si tratta quindi di risolvere il **sistema non lineare**

$$\begin{cases} f(x,y)=x^2+y^2-3=0 \\ g(x,y)=xy-1=0 \end{cases}$$

Separazione grafica: Le due funzioni hanno 4 punti di intersezione: 2 nel primo quadrante e 2 nel terzo.

Ne segue che, detti $\xi_1 = (x_1, y_1)$ e $\xi_2 = (x_2, y_2)$ i punti di intersezione nel primo quadrante, i rimanenti due sono:

$$\xi_3 = (-x_1, -y_1) \quad \text{e} \quad \xi_4 = (-x_2, -y_2).$$

Inoltre, se il punto di coordinate (x_1, y_1) è uno zero sia di f che di g, lo è anche il punto di coordinate (y_1, x_1). Ne segue che

$$\xi_2 = (x_2, y_2) = (y_1, x_1).$$

Il punto $\xi_1 = (x_1, y_1)$ è contenuto in $I_1 = [0, 1] \times [1, \sqrt{3}]$.

Si verifica facilamente che $F(x,y) = [f(x,y), g(x,y)]^T \in C^2(I_1)$.

Inoltre

$$J_F(x,y) = \begin{bmatrix} f_x(x,y) & f_y(x,y) \\ g_x(x,y) & g_y(x,y) \end{bmatrix} = \begin{bmatrix} 2x & 2y \\ y & x \end{bmatrix}$$

e quindi

$$|J_F(x,y)| = 2x^2 - 2y^2 = 0 \quad \Longleftrightarrow \quad x^2 = y^2$$

$$\Rightarrow \quad |J_F(x,y)| \neq 0 \quad \text{in} \quad I_1.$$

Sono verificate le ipotesi di applicabilità del **metodo di Newton**

Scegliendo il punto $X^{(0)} = (x_0, y_0) = \left(\frac{1}{2}, \frac{3}{2}\right)$ come approssimazione iniziale della soluzione si ha

$$\begin{cases} x_1 = x_0 - \dfrac{1}{|J_F(x_0, y_0)|} \left[f(x_0, y_0)\, g_y(x_0, y_0) - g(x_0, y_0)\, f_y(x_0, y_0) \right] \\ y_1 = y_0 - \dfrac{1}{|J_F(x_0, y_0)|} \left[g(x_0, y_0)\, f_x(x_0, y_0) - f(x_0, y_0)\, g_x(x_0, y_0) \right] \end{cases} =$$

$$= \begin{cases} x_1 = \dfrac{1}{2} + \dfrac{1}{4}\left[\left(\dfrac{1}{4} - \dfrac{9}{4} - 3 \right) \dfrac{1}{2} - \left(\dfrac{1}{2} \dfrac{3}{2} - 1 \right) 2 \dfrac{3}{2} \right] = \dfrac{1}{2} + \dfrac{1}{8} = \dfrac{5}{8} \\ y_1 = \dfrac{3}{2} + \dfrac{1}{4}\left[\left(\dfrac{3}{4} - 1 \right) + \dfrac{1}{2} \dfrac{3}{2} \right] = \dfrac{3}{2} + \dfrac{1}{8} = \dfrac{13}{8} \end{cases}$$

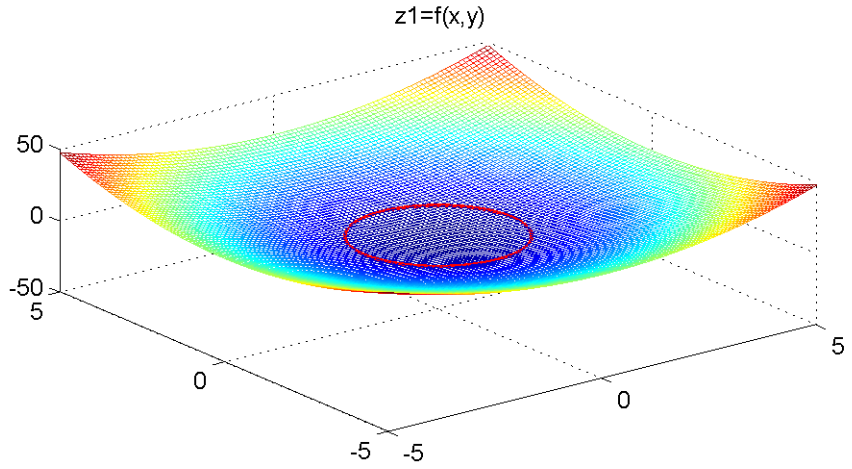
$$\begin{cases} x_2 = x_1 - 0.00694 = 0.61806 \\ y_2 = y_1 - 0.00694 = 1.61806 \end{cases}$$

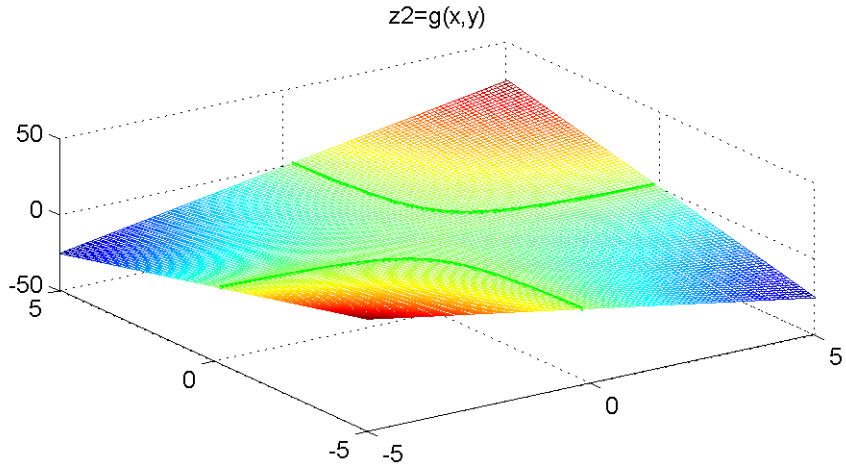
$$\begin{cases} x_3 = x_2 - 0.00003 = 0.61803 \\ y_3 = y_2 - 0.00003 = 1.61803 \end{cases}$$

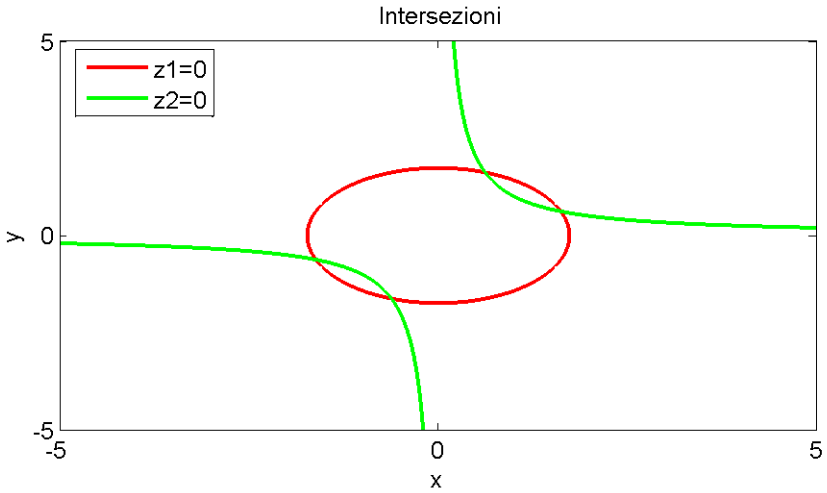
Localizzazione delle radici: rappresentazione grafica

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')
```


z1=f(x,y)
50
0
-50
5
0
-5
-5
0
5


z2=g(x,y)
50
0
-50
5
0
-5
-5
0
5


Intersezioni
z1=0
z2=0
5
0
-5
y
-5
0
5
x

Grafici 3D

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 $\mathbf{D} = [\mathbf{x},\mathbf{y}]$ 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

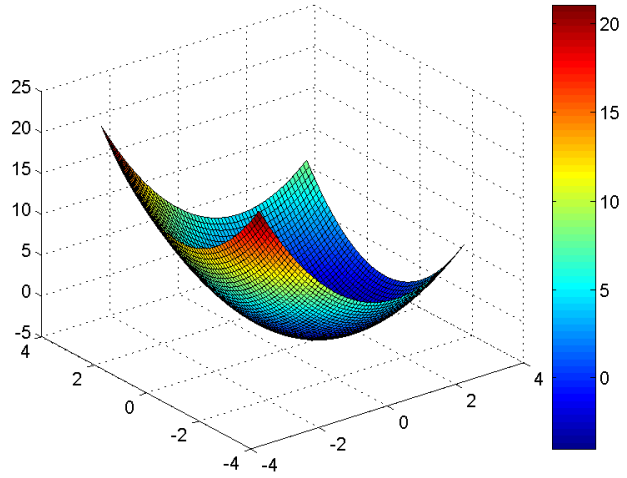
```
f=@(x,y)[x.^2+y.^2-2*x-3];
```

- **valutare la funzione** nei punti della griglia

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

- **disegnare** la funzione valutata nei punti $\mathbf{D} = [\mathbf{x}, \mathbf{y}]$

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



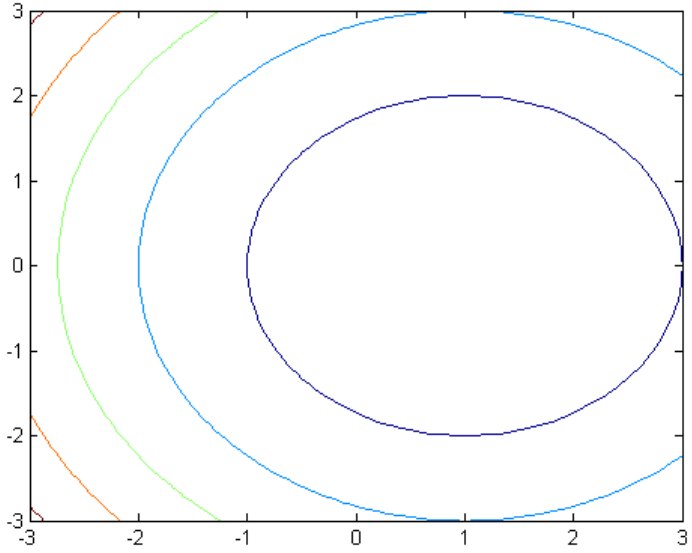
I punti della superficie risultano colorati diversamente secondo il valore assunto. La **colorbar** riporta la scala dei colori

In alternativa si può usa il comando

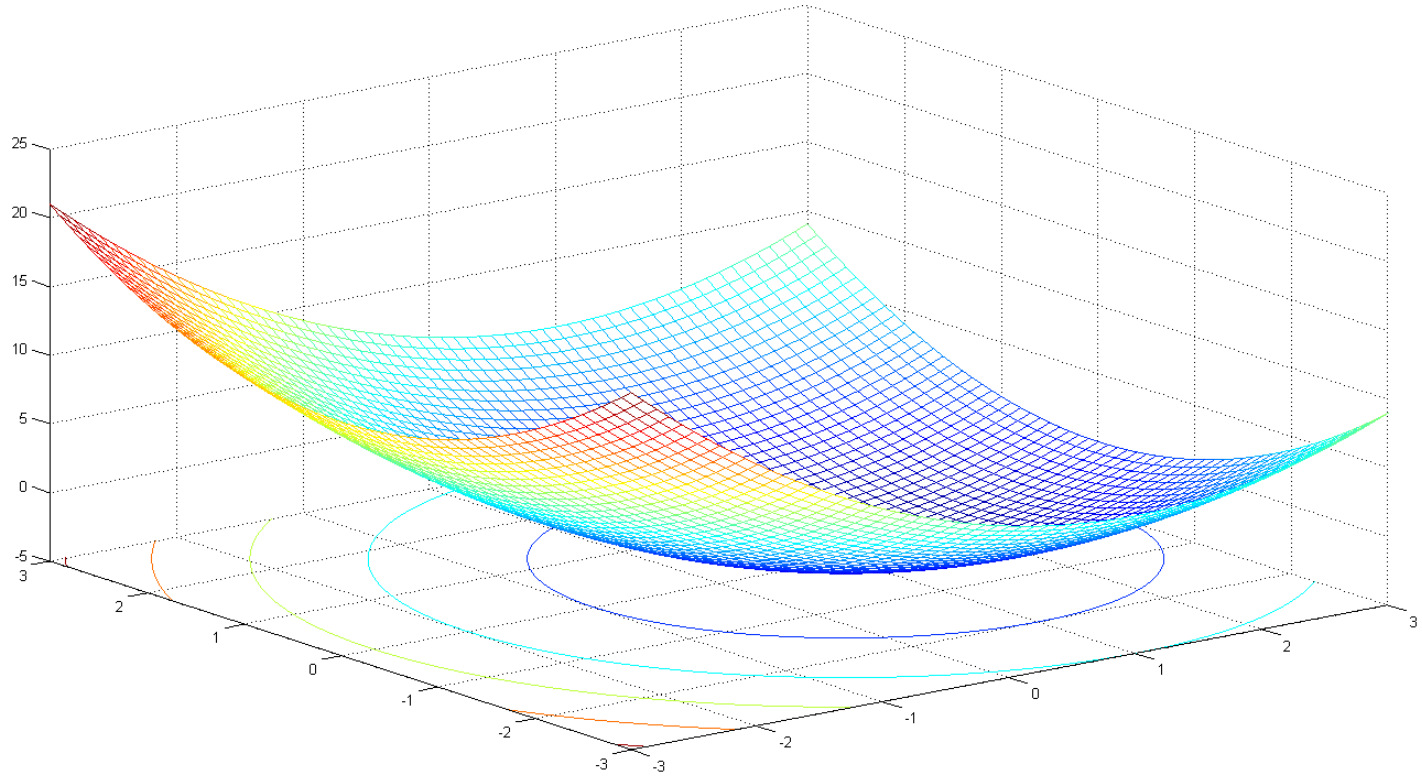
```
mesh(x,y,z)
```

Grafici 3D

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



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



Esercizio Usare l'help per stabilire quando usare le seguenti funzioni **pcolor**(x,y,z), **surf**(x,y,z,gradient(z)), **plot3**(x,y,z)

Esercizio Scrivere uno script matlab per la soluzione del sistema di due equazioni non lineari

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

g =  @(x,y)(x*y-1);
gx = @(x,y)(y);
gy = @(x,y)(x);

xn = 1/2; yn = 3/2;
iter = 0; errx = 10; erry = 10;
errf = abs(f(xn,yn)); errg = abs(g(xn,yn));

[iter xn errx errf yn erry errg]
for iter= 1:10
  xv = xn; yv = yn;
  % calcolo J(X(k))
  Jn = [fx(xv,yv), fy(xv,yv);
        gx(xv,yv), gy(xv,yv)];
```

```
  [iter det(Jn)]
  % calcolo F(X(k))
  Bn = [f(xv,yv); g(xv,yv)];

  % risolvo il sistema lineare
  % J(X(k))*(X(k+1) - X(k)) = -F(X(k))
  % per risolvere un sistema lineare il modo pi semplice
  % in Matlab  usare il comando \
  Vn = Jn\Bn;
  xn = xv-Vn(1);
  yn = yv-Vn(2);

  errx = abs(xn-xv); erry = abs(yn-yv);
  errf = abs(f(xn,yn)); errg = abs(g(xn,yn));
  [iter xn errx errf yn erry errg]
end

subplot(2,2,3)
plot(xn,yn,'.k', 'MarkerSize', 20)
```

Esercizio

Trasformare lo script per la risoluzione del sistema lineare in una function **newton_sistemi** avente i seguenti parametri di input e output:

Input	Ouput
F funzione	**X** soluzione trovata
J jacobiano	**FX** valore di F nello zero calcolato
X0 punto iniziale	**iter** il numero di iterazioni effettuate
eps tolleranza desiderata	

```
% definire F e J nel seguente modo
F = @(x,y)[x.^2+y.^2-3; x*y-1];
J = @(x,y)[2*x, 2*y; ...
           y,   x];
```

Usare la funzione **newton_sistemi** per trovare gli zeri del seguente sitema non lineare

$$\begin{cases} f(x,y) = x^2 + y^2 - 9 = 0 \\ g(x,y) = x + y - 3 = 0 \end{cases}$$

radici: $(0;3)(3;0)$

punto iniziale $x0 = (1;5); x0 = (2;3)$