

# Calcolo Numerico

(A.A. 2012-2013)

**Esercitazione n. 9**

**Metodo del punto unito, Metodo di Newton per sistemi**

**23-04-2013**

## 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 che

- riceva in input la funzione di iterazione  $\phi$ , il punto iniziale  $x_0$ , la precisione richiesta sulla soluzione  $\epsilon$  e il numero massimo di iterazioni da eseguire *maxiter*;
- restituisca in output l'approssimazione della radice  $x$ , l'errore corrispondente  $\epsilon_n$  e il numero di iterazioni eseguite *n\_iter*
- 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  $\varphi$  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  $\xi$  per ogni **approssimazione iniziale**  $x_0 \in I$

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

$\phi$  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  $x = 1$  è il punto unito di  $\phi$ . Inoltre,

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

mentre

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

con  $\phi'(x)$  funzione continua in  $\mathbf{R}$ .  $\phi(x)$  è quindi una funzione **monotona decrescente** per  $x < 1$  e **monotona crescente** per  $x > 1$ .

Poiché  $\phi'$  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  $\phi$ ,

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

$$\begin{aligned} e_n &= \xi - x_n = \varphi(\xi) - \varphi(x_{n-1}) = \\ &= \varphi'(t_n)(\xi - x_{n-1}) = \varphi'(t_n)e_{n-1} \quad t_n \in [x_{n-1}, \xi] \end{aligned}$$

- **Se**  $0 \leq \varphi'(x) < 1$  per  $x \in I$  la successione  $\{x_n = \varphi(x_{n-1})\}$ ,  $n = 1, 2, \dots$ , 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  $\{x_n = \varphi(x_{n-1})\}$ ,  $n = 1, 2, \dots$ , **non è monotona**  $\Rightarrow$  le approssimazioni sono **alternativamente** per **difetto** e per **eccesso**



Poichè  $\phi'$  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  $|\phi'(x)| \leq k$ ,  $k \in (0, 1) \quad \forall x \in I$ .

Si osserva che

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

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

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

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

Quindi,  $\forall I = [1, \bar{x}]$  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  $\phi$ , 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  $I = [1 - \delta, \bar{x}]$ , con  $\delta < 0.5$  e  $\bar{x} \geq 1$  (in particolare,  $\bar{x} \geq \phi(1 - \delta)$ ).

Poichè risulta

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

l'ordine di convergenza del metodo è  $p = 2$ .

**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\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

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

### Metodo del punto unito

k	$x_k$	$\epsilon_k$
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	$x_k$	$\epsilon_k$
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

## 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

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

## Istruzione break

**break**: arresta l' esecuzione dei cicli **for** o **while**

(non può essere utilizzata all' esterno di un ciclo)

```
X = rand(10,1);
Y=[];
for i = 1:10
    if X(i) >= 0.7
        break,
    else Y(i)=1/(0.7-X(i));
    end
end
```

E' particolarmente utile nel caso di cicli **for** in quanto permette di interrompere il ciclo anche prima che tutte le iterazioni prefissate siano state eseguite.

L'istruzione **break** può essere usata nella scrittura di una function matlab per la ricerca del punto unito in alternativa al ciclo **while**.

## Esempio

```
function [xn,err,n_iter] = punto_unitario_break(f,x0,eps,maxiter)
% [xn,err,n_iter] = punto_unitario(f,x0,eps,maxiter)
% cerca il punto unitario 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 unitario
% 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\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

```

## Esercizi

Modificare il criterio di arresto nella funzione **punto\_unito** nel caso in cui sia dato in input il coefficiente di contrazione  $k$  della funzione  $\phi$  (**stima a posteriori**).

Modificare il criterio di arresto nella funzione **punto\_unito\_break** nel caso in cui sia dato in input il coefficiente di contrazione  $k$  della funzione  $\phi$  usando la **stima a priori**.

## Esercizio 7.11

Data l'equazione

$$\alpha(4 - x^2) = x^3, \quad \alpha > 0$$

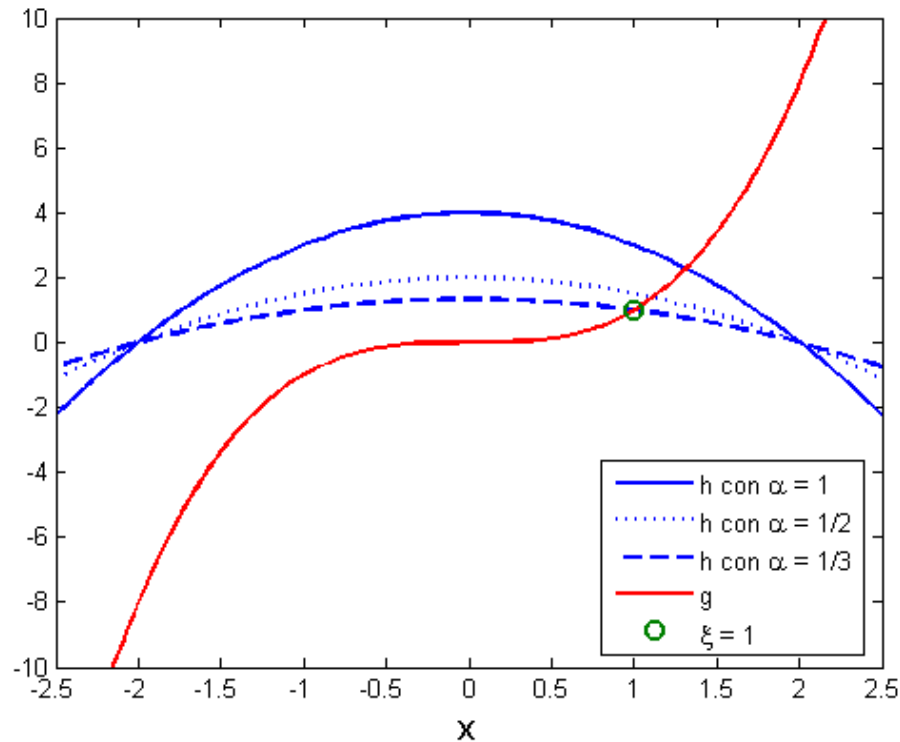
1. separare graficamente le soluzioni positive ed individuare per quali valori di  $\alpha$  l'equazione ammette una radice  $\xi > 1$ ;
2. posto  $\alpha = 1$ , introdurre una opportuna funzione di iterazione  $x = \phi(x)$ ,  $x \in [1, 1.5]$  adatta ad approssimare la radice  $\xi > 1$ ;
3. in base al comportamento di  $\phi$  caratterizzare la successione delle approssimazioni  $x_n = \phi(x_{n-1})$  (ordine di convergenza, monotonia, etc. ).
4. stimare il numero di iterazioni necessarie affinché il procedimento iterativo produca un'approssimazione della soluzione con 5 decimali esatti e confrontarlo con il numero di iterazione realmente necessarie usando la funzione `punto_unito`.



**Soluzione:** Si considerano le funzioni  $h(x) = \alpha(4 - x^2)$  e  $g(x) = x^3$ .

$h(x)$ , al variare del parametro  $\alpha$ , è una parabola con concavità rivolta verso il basso e vertice  $V = (0, 4\alpha)$ . Quindi,  $V$  si avvicina a 0 man mano che il valore del parametro  $\alpha$  decresce.

Inoltre, come mostrato nel grafico, la **radice positiva esiste ed è unica**



Poichè  $h(1) = 3\alpha$  mentre  $g(1) = 1$ , risulta  $\xi = 1$  se  $\alpha = 1/3$ . Ne deduciamo che la radice positiva è **maggiore di 1** se  $\alpha > \frac{1}{3}$

Se  $\alpha = 1$ , l'equazione diventa  $4 - x^2 = x^3$ .

Nell'intervallo  $I = [1, 1.5]$ ,

$$x = \phi(x) = \sqrt[3]{4 - x^2}$$

genera un **procedimento iterativo convergente**.

Infatti,  $\phi$  è una funzione continua, derivabile e positiva in  $I$ .

Inoltre

$$\phi'(x) = -\frac{2}{3} \frac{x}{\sqrt[3]{(4 - x^2)^2}} < 0 \quad \forall x \in I.$$

Quindi  $\phi$  è decrescente, da cui

$$\phi(1) \geq \phi(1.5).$$

Poichè  $\phi(1) = \sqrt[3]{3} \approx 1.44$  e  $\phi(1.5) \approx 1.20$ , risulta

$$\phi(I) \subset I.$$

Inoltre, si osserva che  $|\phi'(x)| = \frac{2}{3} \frac{|x|}{\sqrt[3]{(4-x^2)^2}}$ .

La funzione a secondo membro è una funzione monotona crescente in  $I$  in quanto è il prodotto di funzioni monotone crescenti e positive in  $I$  e quindi

$$|\phi'(x)| \leq |\phi'(1.5)| = k \approx 0.69 < 1.$$

Poichè  $\phi'(x) < 0 \quad \forall x \in I$ , il metodo converge, producendo approssimazioni della radice  $\xi$  alternativamente per eccesso e per difetto.

Poichè  $\phi'(x) = 0 \Leftrightarrow x = 0$ , risulta  $\phi'(\xi) \neq 0$

e quindi l'ordine di convergenza del metodo iterativo è  $\mathbf{p = 1}$ .

Per stimare il **numero di iterazioni** necessarie si usa la seguente disuguaglianza

$$|e_n| = |x_n - \xi| \leq \frac{k^n}{1 - k}(b - a)$$

che nel caso considerato diventa

$$|e_n| = |x_n - \xi| \leq \frac{0.69^n}{1 - 0.69}(1.5 - 1)$$

Richiedendo **5 decimali significativi** alla approssimazione prodotta, risulta

$$|e_n| < \epsilon = 0.5 \cdot 10^{-5}$$

cioè

$$\frac{0.69^n}{0.31}(0.5) < 0.5 \cdot 10^{-5}$$

da cui

$$n \approx \frac{\log(2 \cdot (0.31) \cdot 0.5 \cdot 10^{-5})}{\log(0.69)} \approx 34.$$

Usando la funzione `punto_unito` si ottengono i risultati seguenti  
`[xn,err,n_iter] = punto_unito(f,1.1,.5*10-5,100);`

$k$	$x_k$	$x_k - x_{k-1}$
2	1.407779816636963	0.307779816636963
3	1.263722089604660	0.144057727032303
4	1.339424732696048	0.075702643091388
5	1.301761199775750	0.037663532920298
6	1.321041758493116	0.019280558717367
7	1.311311289649813	0.009730468843303
8	1.316257901162381	0.004946611512568
9	1.313752451659470	0.002505449502911
10	1.315023829271998	0.001271377612529
11	1.314379285273200	0.000644543998798
12	1.314706203444088	0.000326918170888
13	1.314540428132803	0.000165775311285
14	1.314624500689312	0.000084072556509
15	1.314581866160240	0.000042634529072
16	1.314603487493636	0.000021621333396
17	1.314592522800411	0.000010964693225
18	1.314598083302941	0.000005560502530
19	1.314595263428300	0.000002819874641

Infine, si osserva che le funzioni

$$\phi_1(x) = \sqrt{4 - x^3},$$

$$\phi_2(x) = \frac{4}{x^2} - 1$$

$$\phi_3(x) = x^3 + x^2 + x - 4$$

non sono adatte ad approssimare la radice positiva della funzione considerata in quanto per esse non è verificata la condizione  $\phi(I) \subseteq I$ .



## Metodo di Newton per sistemi: $n = 2$

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

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

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

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

## 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 facilmente 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

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

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

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

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

$$= \begin{cases} x_1 = \frac{1}{2} + \frac{1}{4} \left[ \left(\frac{1}{4} - \frac{9}{4} - 3\right) \frac{1}{2} - \left(\frac{13}{22} - 1\right) 2 \frac{3}{2} \right] = \frac{1}{2} + \frac{1}{8} = \frac{5}{8} \\ y_1 = \frac{3}{2} + \frac{1}{4} \left[ \left(\frac{3}{4} - 1\right) + \frac{1}{2} \frac{3}{2} \right] = \frac{3}{2} + \frac{1}{8} = \frac{13}{8} \end{cases}$$

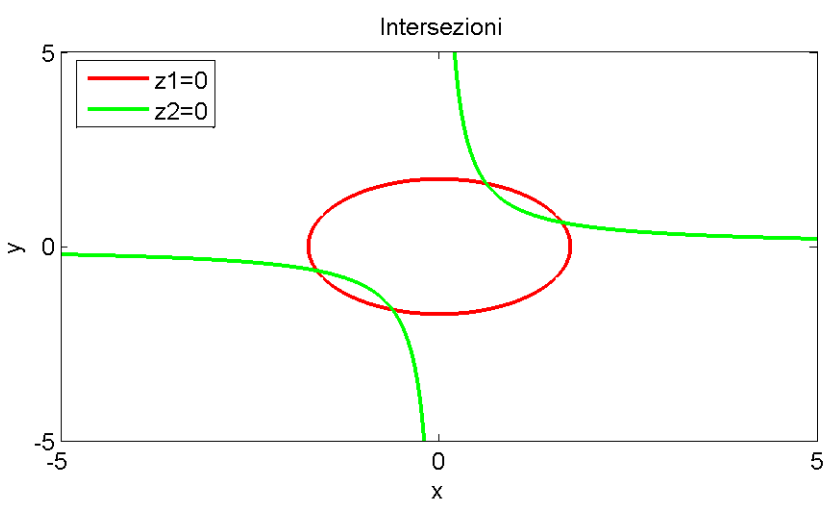
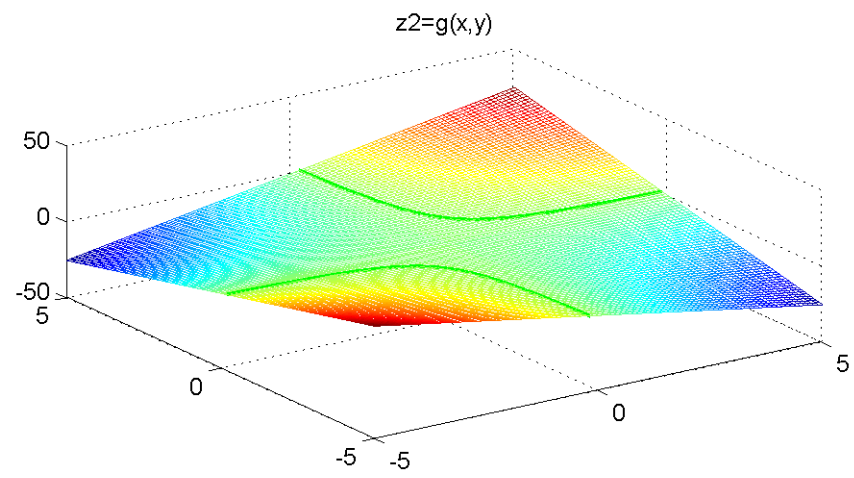
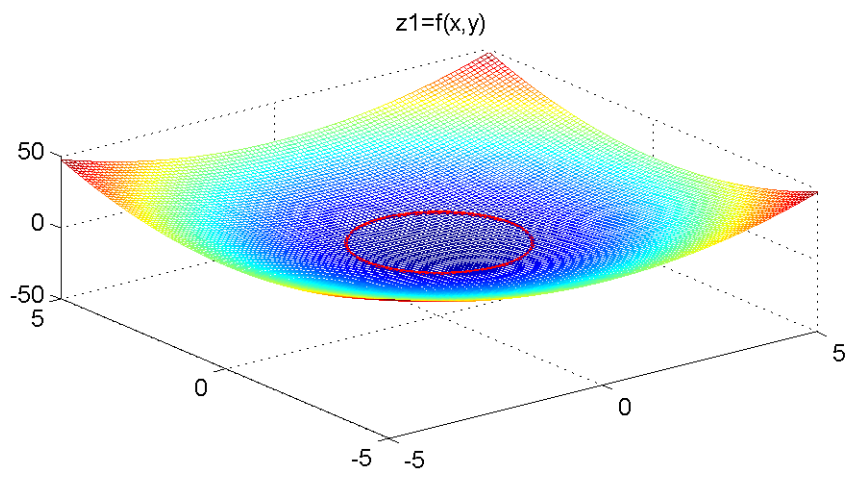
$$\begin{cases} x_2 = x_1 - 0.00694 = 0.61806 \\ y_2 = y_1 - 0.00694 = 1.61806 \end{cases}$$

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

## 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')
```





## 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  $\mathbf{X} = [x, y]$  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

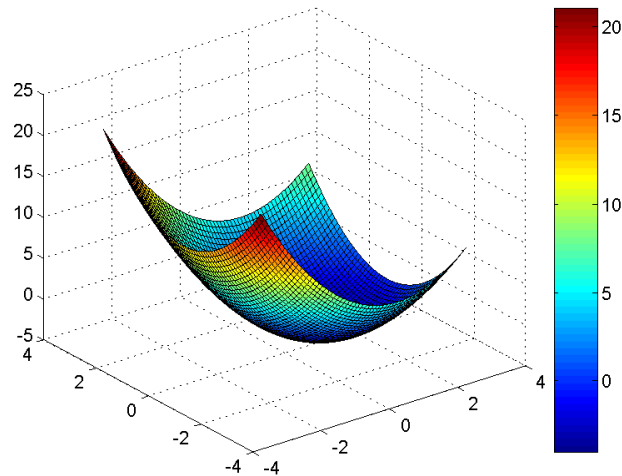
```
f=@(x,y) [x.^2+y.^2-2*x-3];
```

- valutare la funzione nei punti della griglia

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

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

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

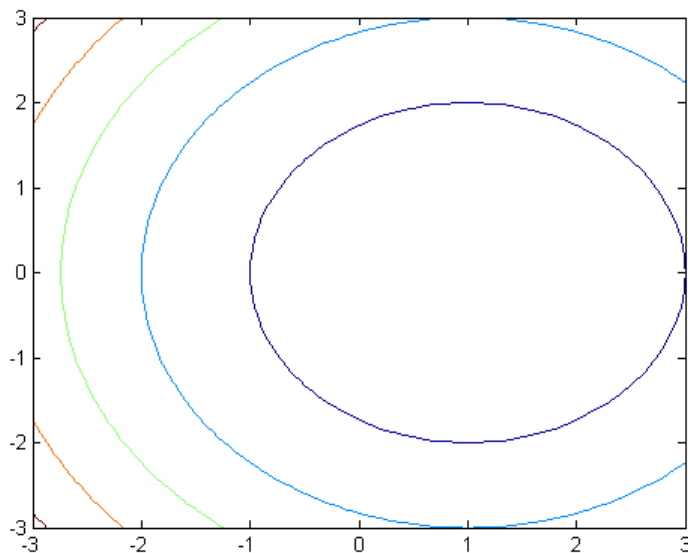


I punti della superficie risultano colorati diversamente secondo il valore assunto. La colorbar riporta la scala dei colori

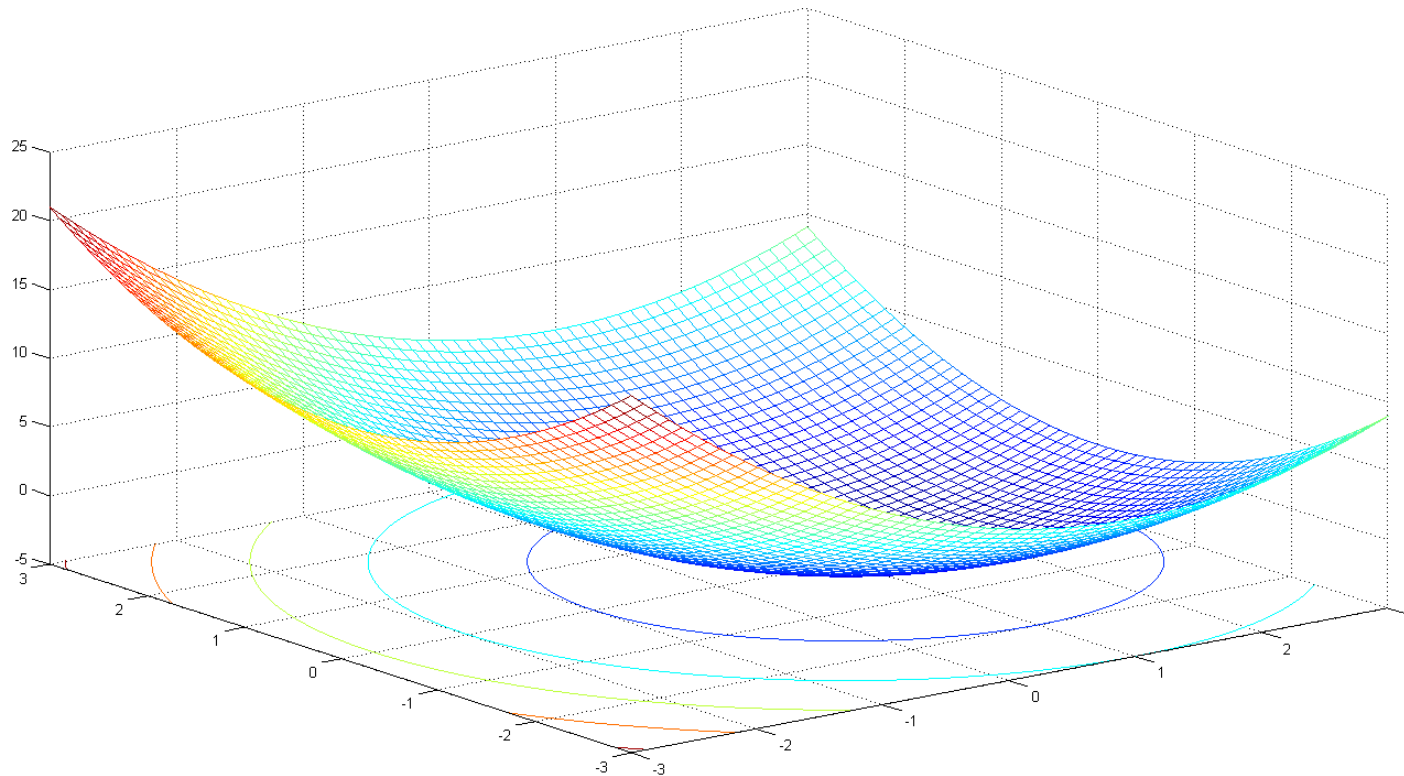
In alternativa si può usare il comando

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

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 lo help per stabilire quando usare le seguenti funzioni `pcolor(x,y,z)`, `surf(x,y,z,gradient(z))`, `plot3(x,y,z)`

**Esercizio** Scrivere una function matlab per la soluzione di una sistema di due equazioni non lineari con opportune variabili di input e output.