RISOLUZIONE NUMERICA DI SISTEMI NON LINEARI

Esempio:

si vuole risolvere numericamente

$$\begin{cases} x_1^2 - x_2^2 = 1 \\ x_1^2 + x_2^2 - 2x_1 = 3. \end{cases} \tag{1}$$

Definisco:

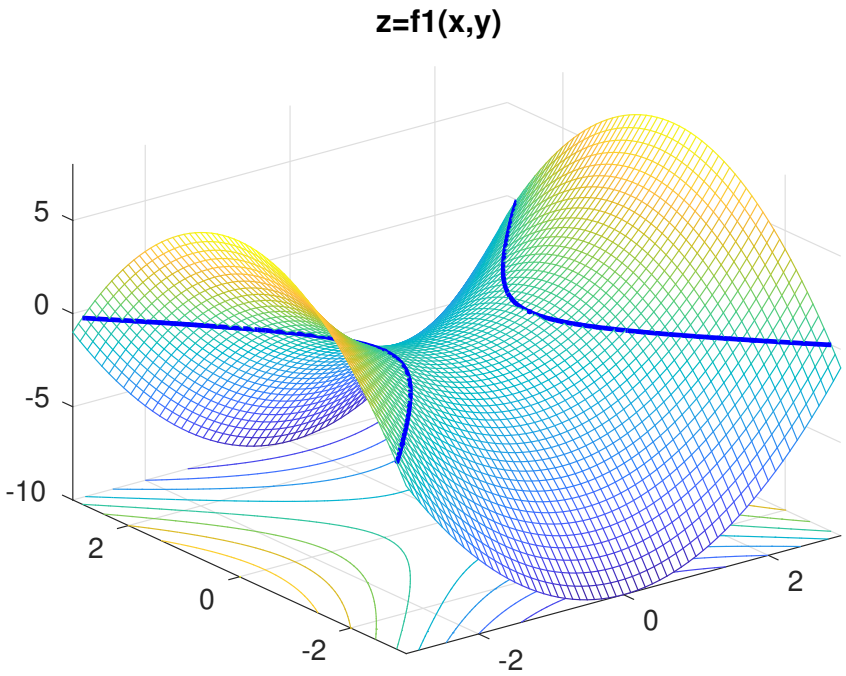
$$f_1(x_1, x_2) = x_1^2 - x_2^2 - 1$$
$$f_2(x_1, x_2) = x_1^2 + x_2^2 - 2x_1 - 3$$

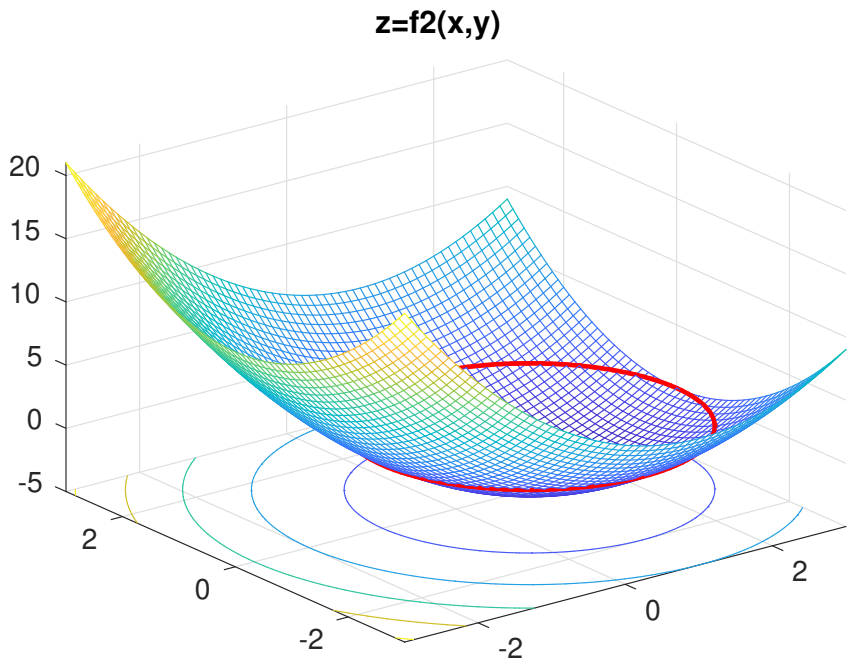
e

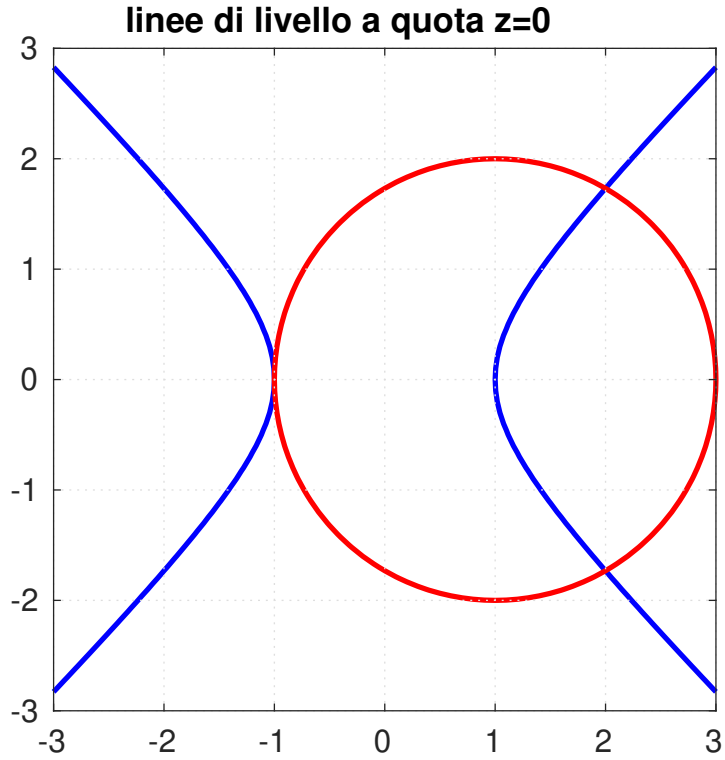
$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix}$$

Risolvere (1) vuol dire

$$\text{trovare } \boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T \in \mathbb{R}^2 : \text{tale che } \mathbf{F}(\boldsymbol{\alpha}) = \mathbf{0}.$$


z=f1(x,y)
5
0
-5
-10
2
0
-2
2
0
-2


z=f2(x,y)
20
15
10
5
0
-5
2
0
-2
2
0
-2


linee di livello a quota z=0
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

Più in generale, per risolvere il sistema

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

si definiscono $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ e la funzione vettoriale

$$\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n : \quad \mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) = f_1(x_1, x_2, \ldots, x_n) \\ f_2(\mathbf{x}) = f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(\mathbf{x}) = f_n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

Si cerca $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T \in \mathbb{R}^n$: tale che $\mathbf{F}(\boldsymbol{\alpha}) = \mathbf{0}$.

METODO DI NEWTON PER SISTEMI

Ricordo che Newton per equazioni scalari è:

$$\begin{cases} x^{(0)} \text{ dato} \\ x^{(k+1)} = x^{(k)} - \dfrac{f(x^{(k)})}{f'(x^{(k)})}, \quad k = 0, 1, ... \end{cases}$$

Per lavorare con funzioni vettoriali $\mathbf{F}(\mathbf{x})$ devo sostituire: $f'(x^{(k)})$ con la matrice Jacobiana di $\mathbf{F}$ valutata in $\mathbf{x}^{(k)}$

$$J_F(\mathbf{x}^{(k)}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}^{(k)}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}^{(k)}) & \dots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}^{(k)}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}^{(k)}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}^{(k)}) & \dots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}^{(k)}) \\ \dots & \dots & \ddots & \dots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}^{(k)}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}^{(k)}) & \dots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}^{(k)}) \end{bmatrix}$$

e reinterpretare la divisione $1/f'(x^{(k)})$ come calcolo della matrice inversa $J_F^{-1}(\mathbf{x}^{(k)})$.

Allora il metodo di Newton per equazioni scalari

$$\begin{cases} x^{(0)} \text{ dato} \\ x^{(k+1)} = x^{(k)} - \dfrac{f(x^{(k)})}{f'(x^{(k)})}, \quad k \geq 0 \end{cases}$$

diventa, per equazioni vettoriali:

$$\begin{cases} \mathbf{x}^{(0)} \text{dato} \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - J_F^{-1}(\mathbf{x}^{(k)})\mathbf{F}(\mathbf{x}^{(k)}) \quad k \geq 0 \end{cases} \tag{2}$$

Il problema cruciale è: come calcolare il vettore $\mathbf{z} = -J_F^{-1}(\mathbf{x}^{(k)})\mathbf{F}(\mathbf{x}^{(k)})$

Come calcolare $\mathbf{z} = -J_F^{-1}(\mathbf{x}^{(k)})\mathbf{F}(\mathbf{x}^{(k)})$

Ad ogni iterazione k:
- valuto $J_F(\mathbf{x}^{(k)})$ e la memorizzo in una matrice A
- valuto $-\mathbf{F}(\mathbf{x}^{(k)})$ e lo memorizzo in un vettore $\mathbf{b}$
- calcolo $\mathbf{z} = A\backslash\mathbf{b}$

Attenzione: Il comando $\backslash$ non calcola in maniera esplicita A^{-1}, ma calcola $\mathbf{z}$ risolvendo il sistema lineare $A\mathbf{z} = \mathbf{b}$

Se A è non singolare $\quad A\mathbf{z} = \mathbf{b} \Leftrightarrow \mathbf{z} = A^{-1}\mathbf{b}$

Partendo dalla function `newton.m` scrivere NEWTON per SISTEMI

`newtonsys.m`

Input: `f, Jf, x0, tol, kmax`

`k=0, err=tol+1`
mentre `k` $<$ `kmax` e `err` $\geq$ `tol`
valuto $\mathbf{b} = -\mathbf{F}(\mathbf{x}^{(k)})$
valuto $A = J_F(\mathbf{x}^{(k)})$
risolvo $A\mathbf{z} = \mathbf{b}$
aggiorno la soluzione $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{z}$
calcolo l'errore `err`$=\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| = \|\mathbf{z}\|$
aggiorno `k`: `k=k+1`
aggiorno `x`

Output: `zero, f(zero), n_iterazioni, [err].`

Teorema (conv di Newton per sistemi)

Sia $\underline{F}: \text{dom}(\underline{F}) \subseteq \mathbb{R}^n \to \mathbb{R}^n$ e $\underline{\alpha}$ radice di $\underline{F}$ cioè t.c. $\underline{F}(\underline{\alpha}) = \underline{0}$.

Se $\underline{F} \in C^2(\text{dom}(\underline{F}))$ e J_F è non singolare $(\det \neq 0)$ $\forall \underline{x} \in I(\underline{\alpha}) \setminus \{\underline{\alpha}\} \Rightarrow$ Newton converge alla radice $\underline{\alpha}$ $\forall x^{(0)}$ sufficiente vicino ad $\underline{\alpha}$ con ordine 2 se $\underline{\alpha}$ è semplice, con ordine 1 se $\underline{\alpha}$ è multipla

---

Estensione del metodo di secanti in $\mathbb{R}^n$

I diversi metodi che generalizzano secanti.

In particolare consideriamo il metodo di Broyden

sia $\underline{x}^{(0)}$ dato in $\mathbb{R}^n$

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - B_k^{-1} \underline{F}(\underline{x}^{(k)}) \qquad \text{per } k \geq 0$$

(B_k sostituisce $J_F(\underline{x}^{(k)})$)

dove B_0 è assegnata (es $B_0 = \mathbf{I}$)

e $$B_{k+1} = B_k + \frac{\underline{F}(\underline{x}^{(k+1)})\,(\underline{x}^{(k+1)} - \underline{x}^{(k)})^T}{\| \underline{x}^{(k+1)} - \underline{x}^{(k)} \|^2}$$

B_k sia un'approx $J_F(\underline{x}^{(k)})$ nel senso che

$$B_k(\underline{x}^{(k)} - \underline{\alpha}) \text{ sia un'ottima approx di}$$

$$J_F(\underline{x}^{(k)})(\underline{x}^{(k)} - \underline{\alpha})$$

Broyden è un metodo a convergenza locale ($\underline{x}^{(0)}$ deve essere vicino ad $\underline{\alpha}$) e superlineare ($l'ordine\ p\ di\ conv\ è\ 1 < p < 2$).