CN25 - Homework2

Matteo Mazzetti 0001161552

1 Funzioni da analizzare

a) $f_a(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 2)^2$

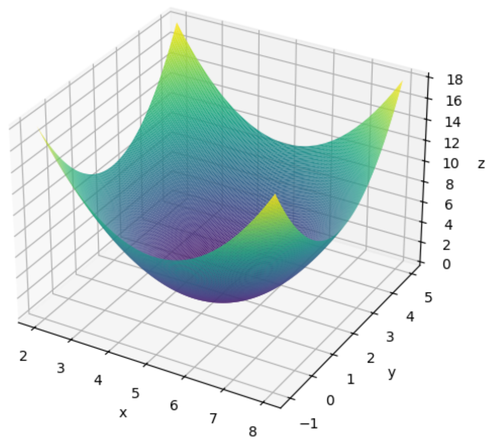

Figure 1: $f_a : \mathbb{R}^2 \to \mathbb{R}$

b) $f_b(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$

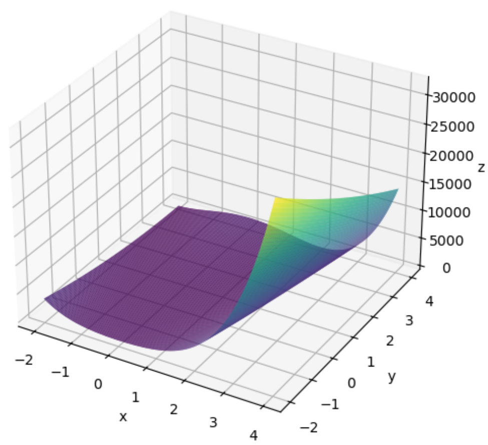

Figure 2: $f_b : \mathbb{R}^2 \to \mathbb{R}$

d) $f_d(x) = \frac{1}{2}\|Ax - b\|_2^2$ dove $A \in M_n(\mathbb{R})$ è una matrice con elementi casuali e $b \in \mathbb{R}^n$ si trova ponendo $\overline{x} = (1, 1, .., 1)^T$ e risolvendo $b = A\overline{x}$.
f) $f_f(x) = \sum_{i=1}^{n}(x_i - i)^2 - \sum_{i=1}^{n} ln(x_i)$ dove $x \in \mathbb{R}^n$ con $x_i > 0\ \forall i = 1..n$

2 Metodo del gradiente

Per utilizzare il metodo del gradiente dobbiamo prima calcolare i gradienti delle quattro funzioni che stiamo studiando:
a) $\nabla f_a(x_1, x_2) = (2(x_1 - 5), 2(x_2 - 2))$
b) $\nabla f_a(x_1, x_2) = (-2(1 - x_1) - 400x_1(x_2 - x_1^2), 200(x_2 - x_1^2))$
d) $\nabla f_d(x) = A^T(Ax - b)$
f) $\partial_{x_i} f_f(x) = 2(x_i - i) - \frac{1}{x_i}$
Il metodo del gradiente è implementato in python così:

```
def GD(f,df,x0,alpha,maxIt,fToll,xToll,alphaConst):
    cont=0
    n=len(x0)
    dfNorm=np.linalg.norm(df(x0))
    while cont<maxIt and dfNorm>fToll:
        p=-df(x0)
        alpha=alpha if alphaConst else backtrackingFun(f,df,x0)
        xNew=x0+alpha*p
        if(np.linalg.norm(xNew-x0)<xToll):
            break
        x0=xNew
        dfNorm=np.linalg.norm(df(x0))
        cont+=1
    return x0,f(x0),cont,alphaConst,alpha
```

Utilizziamo delle tabelle per visualizzare meglio i risultati.
Per la funzione $f_a(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 2)^2$ si ha che:

Test	x0	Alpha	AlphaConst	Iters	x*	f(x*)
1	(7,-1)	1	F	1	(5,2)	0
2	(7,-1)	0.001	T	3288	(5.00276853, 1.9958472)	2.491052370848344e-05
3	(5,0)	0.001	T	2993	(5,1.99500265)	2.4973541699900752e-05
4	(-100,100)	15	F	1	(5,2)	0

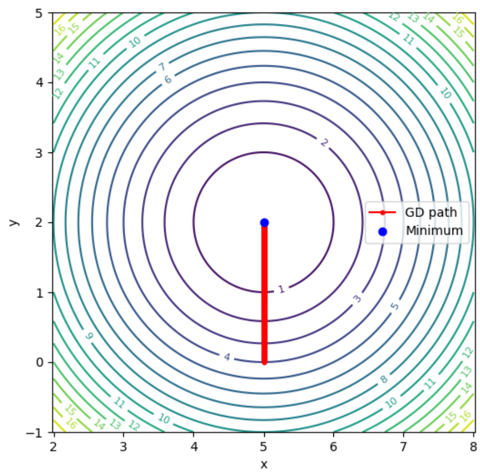

Figure 3: f_a Test 3

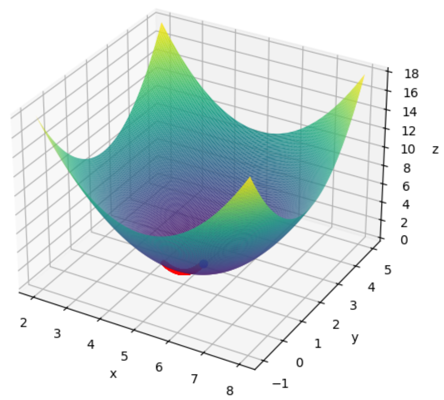

Figure 4: f_a Test 3

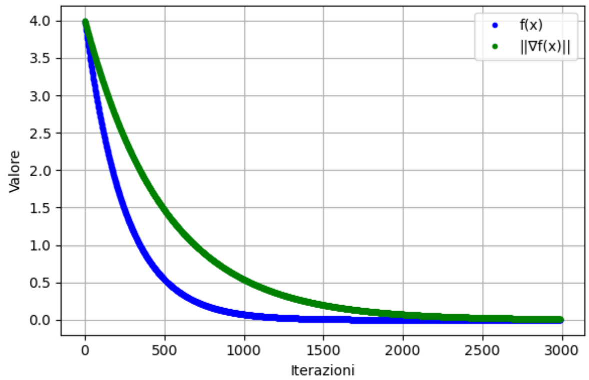

Figure 5: f_a Test 3

Per la funzione $f_b(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$ nota come funzione di Rosenbrock si ha che:

Test	**x0**	**Alpha**	**AlphaConst**	**Iters**	**x***	**f(x*)**
1	(0,2)	0.01	T	\	\	\
2	(0,2)	0.005	T	10000[1]	(0.53419238, 0.21465105)	0.7169736034126697
3	(0,2)	0.001	T	8270	(0.98892228, 0.97792265)	0.0001229150422919122
4	(0,2)	0.001	F	4375	(0.99616119, 0.99231348)	1.4792320012593029e-05

1 L'algoritmo termina sempre per iterazioni anche con maxIt=30000 e 100000.

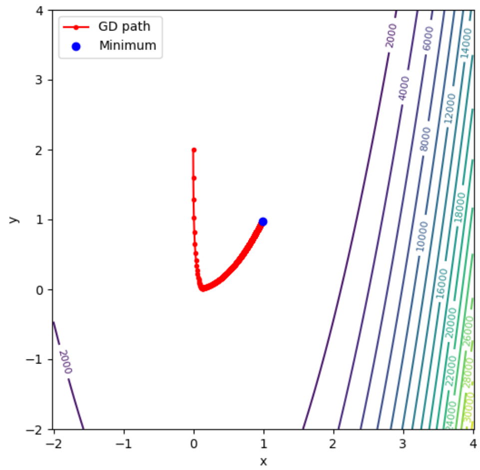

Figure 6: f_b Test 3

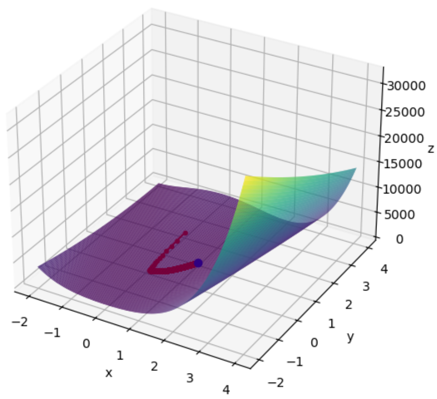

Figure 7: f_b Test 3

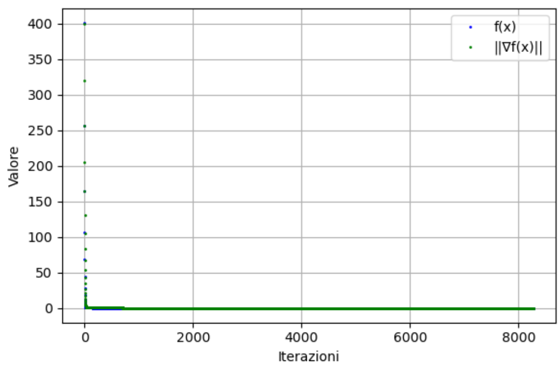

Figure 8: f_b Test 3

La funzione dei minimi quadrati lineari, ossia $f_d(x) = \frac{1}{2}\|Ax - b\|_2^2$ richiede che A sia una matrice quadrata casuale, nel nostro caso

$$A = \begin{pmatrix} 0.53632201 & 0.87712709 & 0.26605249 & 0.29726022 & 0.53965021 \\ 0.37268831 & 0.10548683 & 0.35747810 & 0.72337934 & 0.10020472 \\ 0.13209922 & 0.16335677 & 0.49748149 & 0.43642859 & 0.59638987 \\ 0.36028065 & 0.81779636 & 0.84351303 & 0.44975310 & 0.96032015 \\ 0.91500373 & 0.81463979 & 0.81824263 & 0.78417968 & 0.85253613 \end{pmatrix} \in M_5(\mathbb{R})$$

Test	**x0**	**Alpha**	**AlphaConst**	**Iters**	**x***	**f(x*)**
1	(-0.20281018, 0.21978563, -0.40081749, -0.86387318, -0.03894452)	0.001	F	601	(1.00018352, 0.99965985, 0.99923681, 1.00017566, 1.00072936)	1.0042222485192047e-08
2	(-0.20281018, 0.21978563, -0.40081749, -0.86387318, -0.03894452)	0.001	T	7886	(1.03525667, 0.97567506, 0.99672828, 0.95905503, 1.0312188)	0.00015737513383347375
3	(-0.20281018, 0.21978563, -0.40081749, -0.86387318, -0.03894452)	0.005	T	2458	(1.03488284, 0.97085704, 0.99783373, 0.97531017, 1.02222603)	4.765283381807054e-05
4	(-1.1134856, 0.64990667, 0.07236573, -0.58268406, 0.3822194)	0.001	F	950	(0.99991373, 1.00029836, 1.00091593, 0.99969994, 0.9991971)	1.2436769430423379e-08

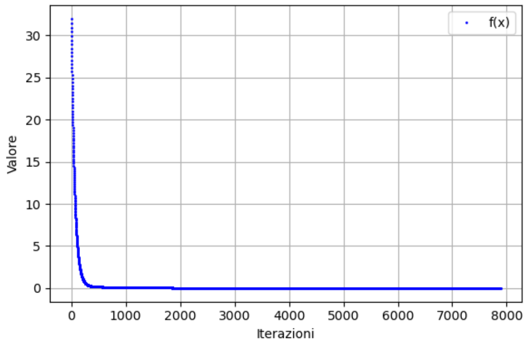

Figure 9: f_d Test 2

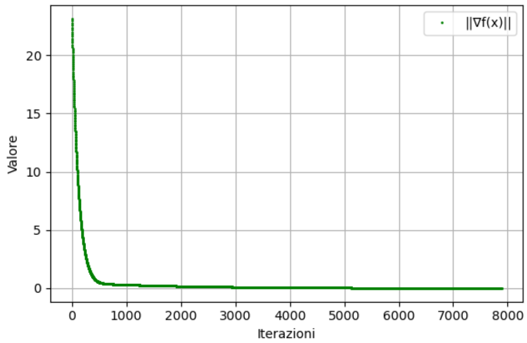

Figure 10: f_d Test 2