METODA PASULUI DESCENDENT

Problemă Fie $f(x,y) = x^2/4 + y^2$ si domeniul $(x,y) \in [-6,6]X[-4,4]$. Să se afle punctul de minim local conform metodei pasului descendent. Să se reprezinte grafic atât funcţia f cât şi punctul de minim. Mai mult, într-o altă figură să se reprezinte traseul format din punctele de la fiecare iteraţie, liniile de nivel care trec prin punctele respective şi direcţiile gradientului.

```
#Definim doi tupli avand drept valori capetele intervalelor.
(a,b)  = (-6,6)
(c,d) = (-4,4)

# Numarul de puncte pe cele doua directii
(Nx, Ny) = (1000, 1000)

#Discretizarea fiecarui interval in parte
x_grafic = np.linspace(a, b, Nx)
y_grafic = np.linspace(c, d, Ny)

#Crearea unei retele de puncte (X,Y) care sa acopere domeniul [a, b]X[c, d]
[X, Y] = np.meshgrid(x_grafic, y_grafic)

#Reprezentarea grafică a reţelei
plt.figure(1)
ax = plt.axes()
ax.plot(X, Y, 'o', markerfacecolor = 'red', markersize = 10)
ax.grid(True)
plt.title('Fig 1: Reteaua de puncte care acopera domeniul [a,b]X[c,d]')
plt.show()

# Definirea funcţiei pentru care dorim să calculăm punctul de minim
f = lambda x,y: x**2 / 4 + y**2

#Se calculeaza cotele punctelor din retea
Z = f(X, Y)

#Reprezentarea grafică a suprafeţei dată de ecuaţia z = f(x,y)
fig = plt.figure(2)
ax = plt.axes(projection='3d')
ax.plot_surface(X, Y, Z, cmap = plt.cm.jet)
plt.title('Fig 2: Reprezentarea grafica a suprafetei z = f(x,y)')

#Construirea procedurii care implementează metoda pasului descendent
def PasDescendent(A,b,x0,eps = 10**(-16)):
""""
Metoda pasului descendent afla punctul de minim al formei patratice f(x) = 1/2<Ax,x> - <b,x>, x = (x,y)^T
Date de intrare: A - matricea asociata formei patratice
                 x0 - punct de initiere  (vector din numpy)
Date de ieşire: xnew - vector cu 2 elemente reprezentând souţia problemei la ultima iteraţia
                xmatrice - reprezintă o matrice formată din valorile intermediare ale lui xnew
                vmatrice - reprezintă o matrice formată din direcţiile gradientului la fiecare iteraţie
```

```
"""
#Inițializare liste goale
x = []
v = []

#Adăugăm primul punct, reprezentând punctul de pornire
x.append([x0[0],x0[1]])

alpha = []

while True:
    v0 = A @ x0 - b
    if np.sqrt(v0[0]**2 + v0[1]**2)<eps:
        break
    v.append([v0[0], v0[1]])
    alpha0 = np.inner(v0, v0) / np.inner(v0, A @ v0)
    alpha.append(alpha0)
    xnew = x0 - alpha0 * v0
    x.append([xnew[0], xnew[1]])
    x0 = xnew.copy()
    xmatrice = np.array(x)
    vmatrice = np.array(v)

return xnew,xmatrice, vmatrice

#Rezolvare problemă
A = np.array([[1.0/2.0,0],[0,2]])
b = np.zeros(2)
x0 = np.array([6,4])
xnew,xtraseu,vtraseu= PasDescendent(A,b,x0)

ax.plot3D(xnew[0],xnew[1], f(xnew[0],xnew[1]), linestyle='None', marker = 'o', markersize = 10)
plt.title( "Punctul de minim pe suprafata z = f(x,y)")

plt.figure(3)
ax = plt.axes()
for i in range(1,xtraseu.shape[0]):
    ax.plot(xtraseu[i,0], xtraseu[i,1],linestyle = 'None',marker = 'o', markerfacecolor = 'red', markersize=10)
    ax.plot(xtraseu[i-1:i+1, 0], xtraseu[i-1:i+1, 1], linestyle = '-', linewidth=3)
    ax.contour(X, Y, Z, levels = [f(xtraseu[i,0], xtraseu[i,1])])
    ax.axis('equal')
plt.title("Traseul parcurs pana la punctul de minim")
```

Fig 1: Reteaua de puncte care acopera domeniul [a,b]X[c,d]

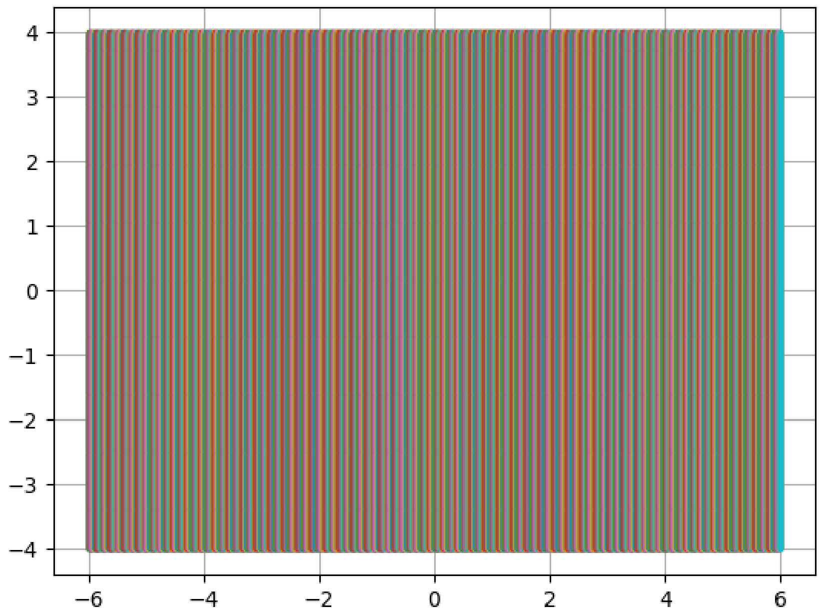
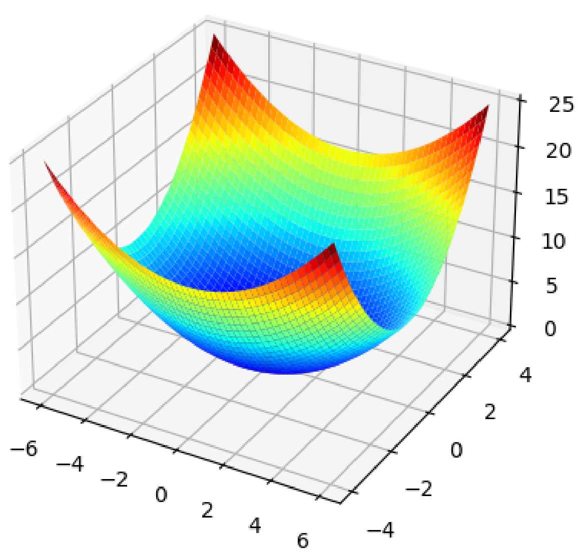
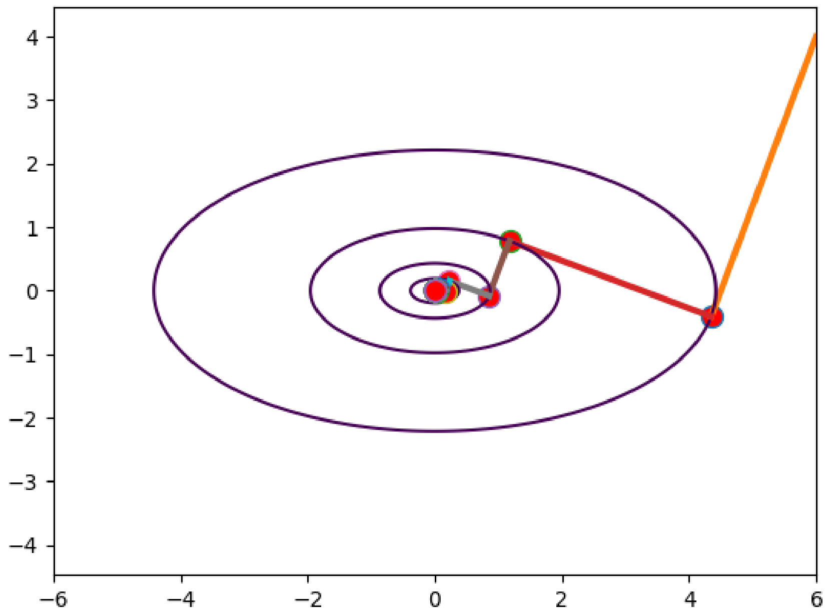


Fig 2: Reprezentarea grafica a suprafetei z = f(x,y)



Traseul parcurs pana la punctul de minim



Punctul de minim pe suprafata z = f(x,y)