Lab 11

Filip Jędrzejewski

June 27, 2023

Zadanie 1

Opis problemu

Celem zadania było wyznaczenie punktów krytycznych każdej z poniższych funkcji. Następnie należało scharatkeryzować każdy znaleziony punkt jako minimum, maksimum lub punkt siodłowy. Dla każdej funkcji zbadano również, czy posiada minimum lub maksimum globalne na zbiorze $\mathbb{R}^2$.

$f_1(x, y) = x^2 - 4xy + y^2$

Gradient oraz Hessian:

$$\nabla f_1(x, y) = \begin{bmatrix} 2x - 4y \\ 2y - 4x \end{bmatrix} \tag{1}$$

$$H_1(x, y) = \begin{bmatrix} 2 & -4 \\ -4 & 2 \end{bmatrix} \tag{2}$$

$$D_1(x, y) = det(H_1(x, y)) = -12 \tag{3}$$

Wyznaczanie punktów krytycznych:

$$\begin{cases} 2x - 4y = 0 \\ -4x + 2y = 0 \end{cases} \tag{4}$$

$$(x, y) = (0, 0) \tag{5}$$

Wyznacznik hessianu dla $(0, 0)$ jest ujemny, zatem jest to punkt siodłowy. Funkcja $f_1(x, y)$ nie ma innych punktów krytycznych, zatem nie ma maksimum lub minimum globalnego.

$f_2(x, y) = x^4 - 4xy + y^4$

Gradient oraz Hessian:

$$\nabla f_2(x, y) = \begin{bmatrix} 4x^3 - 4y \\ 4y^3 - 4x \end{bmatrix} \tag{6}$$

$$H_2(x, y) = \begin{bmatrix} 12x^2 & -4 \\ -4 & 12y^2 \end{bmatrix} \tag{7}$$

$$D_2(x, y) = det(H_2(x, y)) = 144x^2y^2 - 16 \tag{8}$$

Wyznaczanie punktów krytycznych:

$$\begin{cases} 4x^3 - 4y = 0 \\ -4x + 4y^3 = 0 \end{cases} \tag{9}$$

$$(x_1, y_1) = (0, 0) \tag{10}$$

$$(x_2, y_2) = (1, 1) \tag{11}$$

$$(x_3, y_3) = (-1, -1) \tag{12}$$

Wyznacznik hessianu dla $(0, 0)$ jest ujemny, zatem jest to punkt siodłowy. Hessian dla punktów $(1, 1)$ i $(-1, -1)$ ma wyznacznik dodatni, oraz odpowiednie drugie pochodne są dodatnie, z tego wynika, że są to minima.

$$f_2(-1, -1) = f_2(1, 1) = -2 \tag{13}$$

Są to minima globalne.

$f_3(x, y) = 2x^3 - 3x^2 - 6xy(x - y - 1)$

Po przekształceniu:

$$f_3(x, y) = 2x^3 - 3x^2 - 6x^2y + 6xy^2 + 6xy \tag{14}$$

Gradient oraz Hessian:

$$\nabla f_3(x, y) = \begin{bmatrix} 6x^2 - 6x - 12xy + 6y^2 + 6y \\ 12xy + 6x - 6x^2 \end{bmatrix} \tag{15}$$

$$H_3(x, y) = \begin{bmatrix} 12x - 6 - 12y & 12y + 6 - 12x \\ 12y + 6 - 12x & 12x \end{bmatrix} \tag{16}$$

$$D_3(x, y) = det(H_3(x, y)) = 36 \cdot (4xy + 2x - 4y - 4y^2 - 1) \tag{17}$$

Wyznaczanie punktów krytycznych:

$$\nabla f_3(x,y) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{18}$$

$$(x_1, y_1) = (0,0) \tag{19}$$

$$(x_2, y_2) = (1,0) \tag{20}$$

$$(x_3, y_3) = (0,-1) \tag{21}$$

$$(x_4, y_4) = (-1,-1) \tag{22}$$

Wyznacznik hessianu dla $(0,0)$ oraz $(0,-1)$ jest ujemny, zatem jest to punkt siodłowy. Dla $(1,0)$ wyznacznik hessianu jest dodatni i odpowiednie drugie pochodne są dodatnie, zatem jest to minimum lokalne. Dla punktu $(-1,-1)$ wyznacznik hessianu również jest dodatni, jednak odpowiednie drugie pochodne są ujemne, zatem jest to maksimum lokalne. Funkcja $f_3(x,y)$ nie ma maksimum lub minimum globalnego.

$f_4(x,y) = (x-y)^4 + x^2 - y^2 - 2x + 2y + 1$

Gradient oraz Hessian:

$$\nabla f_4(x,y) = \begin{bmatrix} 4(x-y)^3 + 2x - 2 \\ -4(x-y)^3 + -2y + 2 \end{bmatrix} \tag{23}$$

$$H_4(x,y) = \begin{bmatrix} 12(x-y)^2 + 2 & -12(x-y)^2 \\ -12(x-y)^2 & 12(x-y)^2 - 2 \end{bmatrix} \tag{24}$$

$$D_4(x,y) = det(H_4(x,y)) = -4 \tag{25}$$

Wyznaczanie punktów krytycznych:

$$\nabla f_4(x,y) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{26}$$

Rozwiązanie:

$$(x,y) = (1,1) \tag{27}$$

Wyznacznik hessianu dla tego punktu jest ujemny, zatem jest to punkt siodłowy. Funkcja $f_4(x,y)$ nie ma maksimum lub minimum globalnego.

Zadanie 2

Opis problemu

Celem zadania było napisanie programu znajdującego minimum funkcji Rosenbrocka:

$$f(x, y) = 100(y - x^2) + (1 - x)^2 \tag{28}$$

Metoda największego spadku

W celu znalezienia minimum danej funkcji metodą największego spadku, zaimplementowano następującą funkcję:

```
def steepest_descent(x0, N):

  #funkcja pomocnicza minimalizujaca alphe za pomoca metody zlotego podzialu
  def get_alpha(x, g):
    a = 0
    b = 1
    phi = (np.sqrt(5) - 1) / 2
    c = a + (b - a) * phi
    d = b - (b - a) * phi
    while abs(b - a) > 0.001:
        if f(x-d*g) < f(x-c*g):
            b = c
        else:
            a = d
        c = a + (b - a) * phi
        d = b - (b - a) * phi
    return (a + b) / 2

  #kopia dla niezmiennosci
  x = x0.copy()

  #glowna petla
  for i in range(N):
    #wyznaczam gradient
    g = gradient(x)

    #minimalizuje wspolczynnik alpha
    alpha = get_alpha(x, g)
    #wyznaczam nowego x
    x -= alpha * g

  #return wyniku
  return x
```

Funkcja przyjmuje jako parametry punkt początkowy x_0 oraz oczekiwaną liczbę iteracji N. Funkcja korzysta ze zdefiniowanej globalnie metody `gradient(x)`, która wyznacza gradient funkcji `f` w oczekiwanym punkcie `x`.

Metoda Newton'a

W celu znalezienia minimum danej funkcji metodą Newton'a, zaimplementowano następującą funkcję:

```
def newton(x0, N):
    #kopia dla bezpieczenstwa
    x = x0.copy()

    #glowna petla
    for i in range(N):
        #obliczam gradient i hessian
        g = gradient(x)
        h = hessian(x)

        #odwracam hessian, jesli jest osobliwy,
        #to go lekko zmieniam, zeby byl odwracalny
        if np.linalg.det(h) == 0:
            h += np.eye(2) * 0.0000001
        h_inv = np.linalg.inv(h)

        #wyznaczam kolejnego x
        x -= h_inv.dot(g)

    #return wyniku
    return x
```

Funkcja przyjmuje identyczne parametry jak `steepest descent`, jednak wykorzystuje ona również zdefiniowaną globalnie metodę `hessian(x)`, która wyznacza hessian funkcji `f` w oczekiwanym punkcie `x`.

Wyniki i wnioski

Program wykorzystywał globlanie zdefiniowane funkcje obliczające wartość badanej funkcji, jej gradient oraz hessian dla danego punktu x:

```
#funkcja Rosenbrocka
f = lambda x: 100 * (x[1] - x[0]**2)**2 + (1 - x[0])**2

#gradient i hessian
gradient = lambda x: np.array([-400*x[0]*x[1] + 400*x[0]**3 - 4 + 2*x[0],
                               200*x[1] - 200*x[0]**2])
hessian = lambda x: np.array([[-400*x[1] + 1200*x[0]**2, -400*x[0]],
                              [-400*x[0], 200]])
```

Zdefiniowane funkcje przetestowano z następującymi punktami startowymi z liczbą iteracji równą 10 ($N = 10$):

$$x_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{29}$$

$$x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{30}$$

$$x_3 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{31}$$

Kod wywołujący funkcje:

```
#wywolania
X = [
                np.array([-1., 1.]),
                np.array([0., 1.]),
                np.array([2., 1.])
        ]

for x in X:
        print("--------------------------------------------------------")
        print("x_0 = (", x[0], ",", x[1], ")")
        print("Metoda najwiekszego spadku:", steepest_descent(x, 10))
        print("Metoda Newtona:", newton(x, 10))
```

Otrzymane wyniki:

```
--------------------------------------------------------
x_0 = ( -1.0 , 1.0 )
Metoda najwiekszego spadku: [1.00279902 1.00169753]
Metoda Newtona: [-1.36530161e+08  1.86404850e+16]
--------------------------------------------------------
x_0 = ( 0.0 , 1.0 )
Metoda najwiekszego spadku: [1.16503372 1.35430657]
Metoda Newtona: [-2.33524178e+08  5.18416905e+16]
--------------------------------------------------------
x_0 = ( 2.0 , 1.0 )
Metoda najwiekszego spadku: [1.70463885 2.90499583]
Metoda Newtona: [-4.09068824e+09  1.67337303e+19]
```

Oczekiwanym wynikiem (czyli minimum globalnym funkcji Rosenbrocka) był punkt:

$$x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{32}$$

Najlepsze wyniki dawała metoda najszybszego spadku.

Po zauważeniu dużych niezgodności pomiędzy otrzymywanymi wynikami, a tymi oczekiwanymi, przetestowano również zaimplementowane funkcje dla:

$$f(x,y) = x^2 + y^2 + 5x \tag{33}$$

Z oczekiwanym minimum:

$$x_0 = \begin{bmatrix} -\frac{5}{2} \\ \frac{3}{2} \end{bmatrix} = \begin{bmatrix} -2.5 \\ 1.5 \end{bmatrix} \tag{34}$$

Kod funkcji, jej gradientu i hessianu:

```
#funkcja testowa
f = lambda x: x[0]**2 + x[1]**2 + 5*x[0] - 3*x[1]    #minimum -> (-2.5, 1.5)
gradient = lambda x: np.array([2*x[0]+5, 2*x[1]-3])
hessian = lambda x: np.array([[2, 0], [0, 2]])
```

Otrzymane wyniki:

```
x_0 = ( -1.0 , 1.0 )
Metoda najwiekszego spadku: [-2.5  1.5]
Metoda Newtona: [-2.5  1.5]
```

```
x_0 = ( 0.0 , 1.0 )
Metoda najwiekszego spadku: [-2.5  1.5]
Metoda Newtona: [-2.5  1.5]
```

```
x_0 = ( 2.0 , 1.0 )
Metoda najwiekszego spadku: [-2.5  1.5]
Metoda Newtona: [-2.5  1.5]
```

Wyniki w tym przypadku były poprawne.