

Laimonas Beniušis studento nr. 1410102  
Kompiuterių mokslas 1 1gr  
Naudota priemonė: SageMath  
Optimizavimo metodai užduotis 2

Tikslas: Surasti stačiakampio gretasienio formos dėžės matmenis, kad vienetiniam paviršiaus plotui jos tūris būtų maksimalus

Tikslo funkcija:

$$f(x, y, z) = xyz$$

Apribojimo funkcija

$$g(x, y, z) = 2xy + 2xz + 2yz - 1$$

Išsireikškiamas z:

$$2xy + 2xz + 2yz = 1$$

$$z(2x + 2y) = 1 - 2xy$$

$$z = \frac{1 - 2xy}{(2x + 2y)}$$

Nauja tikslo funkcija:

$$f(x, y) = xy \frac{1 - 2xy}{(2x + 2y)}$$

$$a = 0$$

$$b = 2$$

Taškai:

$$X_0 = (0,0)$$

$$X_1 = (1,1)$$

$$X_m = (a/10, b/10) = (0,0.2)$$

Optimizavimo iš taškų rezultato palyginimo santrauka:

Taškas	Iteracijos	Gradientų skaičiavimas	Funkcijos skaičiavimas	Galutinis taškas po optimizavimo	z reikšmė įstačius gauto taško reikšmes
(0,0)	0	1	1	(0,0)	
(1,1)	4	5	5	(0.4082372629230197, 0.4082372629230197)	0.408270345992373
(0,0.2)	10	11	11	(0.4082459467134123, 0.4082500643939627)	0.408248860289704

Taškas  $X_0$  (0,0):

Optimizacija nesėkminga, nes įvyksta dalyba iš 0

Taškas  $X_1$  (1,1):

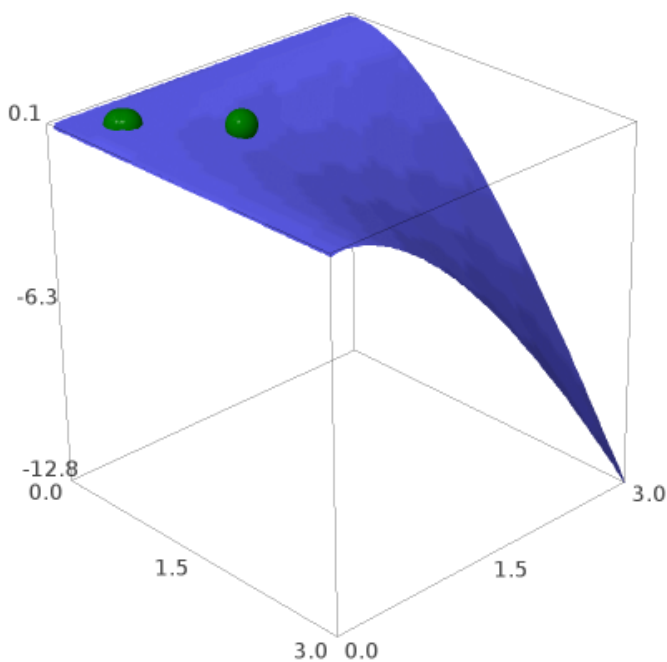
Sėkminga optimizacija

Iteracijos: 4

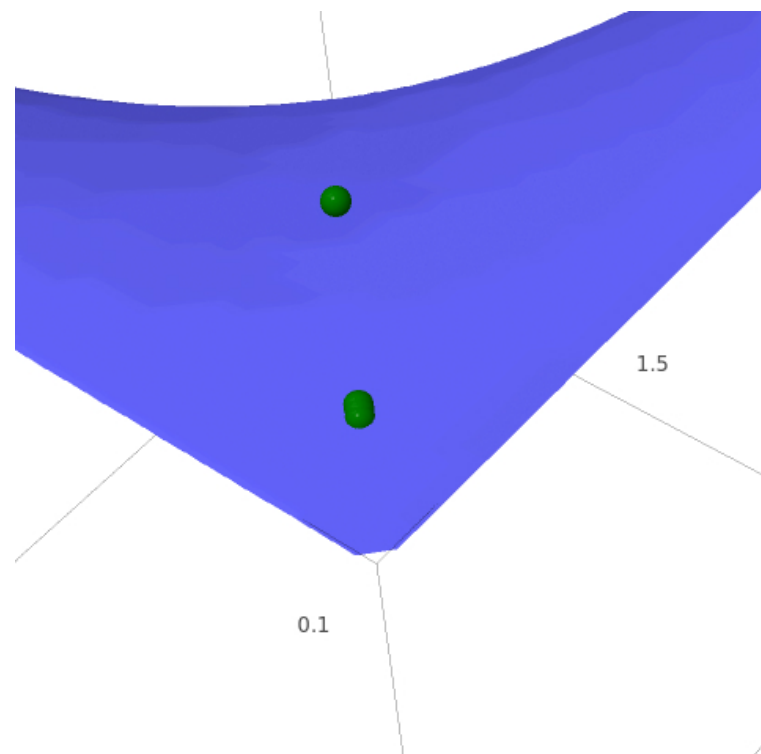
Funkcijos iškvietai: 5

Gradiento įvertinimai: 5

Bandymo taškų vizualizacija



Priartinta versija



Iteracijos:

0: (1, 1)

1: (0.375, 0.375)

2: (0.3939393939393939, 0.3939393939393939)

3: (0.40886699507389157, 0.40886699507389157)

4: (0.4082372629230197, 0.4082372629230197)

Galutinė funkcijos reikšmė: 0.068041

$z = 0.4082703459923731$

Taškas  $X_m$  (0,0.2):

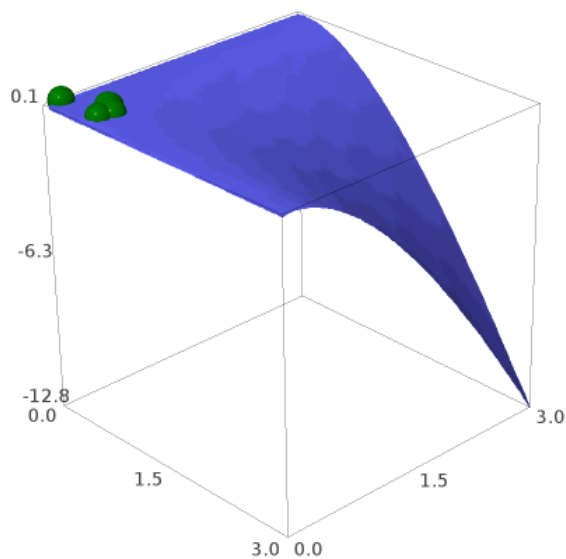
Sėkminga optimizacija

Iteracijos: 10

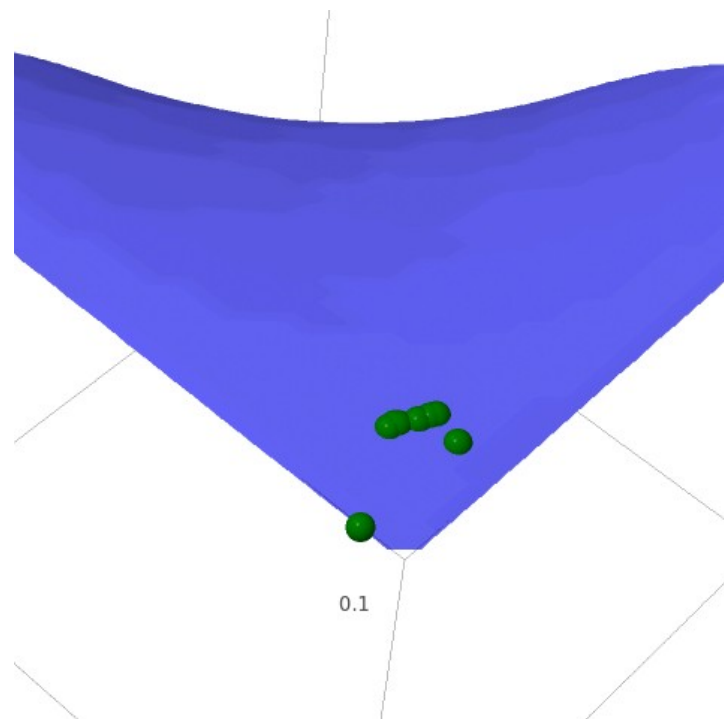
Funkcijos iškvietimai: 11

Gradiento įvertinimai: 11

Bandymo taškų vizualizacija



Priartinta versija



Iteracijos:

0: (0, 0.2)

1: (0.5, 0.2)

2: (0.5398905993327575, 0.3337448559670782)

3: (0.5193589440568842, 0.3427494227813368)

4: (0.4753638417823214, 0.36189455382271773)

5: (0.3790496777426611, 0.4100141367769412)

6: (0.4152679714454278, 0.40047929601468746)

7: (0.40960646787297583, 0.4053957861860429)

8: (0.4080787110290539, 0.4079383156748355)

9: (0.4081962470664747, 0.40823702803395934)

10: (0.4082459467134123, 0.4082500643939627)

Galutinė funkcijos reikšmė: 0.068041

$z = 0.4082488602897038$

Kodas:

```
var("x y z")
z(x,y) = (1 -2*x*y)/(2* (x + y))
fgoal(x,y) = x * y * z
show(z)
show(fgoal)
#constraint = 2*x*y + 2*x*z + 2*y*z -1 = 0
x0 = (0,0)
x1 = (1,1)
xm = (0,0.2)
tryPoints = []
listOfPoints =[x0,x1,xm]
for point in listOfPoints:
    tryPoints = [point]
    for iter in range(1,12):
        print("Point="+str(point))
        try:
            #nėra maximize, todėl minimizuojam neigiamą tikslo funkciją
            ans = minimize(-fgoal,point,maxiter=iter)
            tryPoints.append(ans)
            print("Value="+str(-fgoal(point[0],point[1])))
            print("Iter:"+str(iter)+" Ans="+ str(ans))
            print ("z="+str(z(ans[0],ans[1])))
        except Exception as e:
            print(str(e))
    showPoints = list_plot(tryPoints, color = "green", size = 40)
    print(str(fgoal))
    show(showPoints + plot3d(fgoal,(0,3),(0,3)))

for p in tryPoints:
    print(str(p))
```