KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

TAIKOMOSIOS INFORMATIKOS KATEDRA

2 Laboratorinis darbas
Nr. 10

Atliko:
IFK-8 grupės stud.
Tomas Odinas

Priėmė
lekt. Andrius Kriščiūnas

KAUNAS, 2020

TURINYS

1. UŽDUOTIS

Tiesinių lygčių sistemų sprendimas

Duota tiesinių lygčių sistema [A][X] = [B] ir jos sprendimui nurodytas metodas (1 lentelė).

1. Išspręskite tiesinių lygčių sistemą. Jeigu sprendinių be galo daug, raskite bent vieną iš jų. Jeigu sprendinių nėra, pagrįskite, kodėl taip yra.
   Jei metodas paremtas matricos pertvarkymu, pateikite matricų išraiškas kiekviename žingsnyje. Jei metodas iteracinis, grafiškai pavaizduokite, kaip atliekant iteracijas kinta santykinis sprendinio tikslumas esant kelioms skirtingoms konvergavimo daugiklio reikšmėms.
2. Patikrinkite gautus sprendinius ir skaidas, įrašydami juos į pradinę lygčių sistemą.
3. Gautą sprendinį patikrinkite naudodami išorinius išteklius (pvz., standartines MATLAB funkcijas).

Nr.	Lygčių sistema	Metodas
10	$\begin{cases} 4x_1 + x_2 + x_3 + 7x_4 = 148 \\ x_1 + 2x_3 - 2x_4 = -37 \\ 2x_1 + 2x_2 - 7x_3 + x_4 = 21 \\ 4x_1 + 14x_2 + 7x_3 = 53 \end{cases}$	Gauso – Žordano

Pav. 1

Netiesinių lygčių sistemų sprendimas.

1. Duota netiesinių lygčių sistema (2 lentelė. I lygčių sistema):
   $\begin{cases} Z_1(x_1, x_2) = 0 \\ Z_2(x_1, x_2) = 0 \end{cases}$

   a. Skirtinguose grafikuose pavaizduokite paviršius $Z_1(x_1, x_2)$ ir $Z_2(x_1, x_2)$.
   b. Užduotyje pateiktą netiesinių lygčių sistemą išspręskite grafiniu būdu.
   c. Užduotyje pateiktą netiesinių lygčių sistemą išspręskite naudodami užduotyje nurodytą metodą su laisvai pasirinktu pradiniu artiniu (išbandykite bent keturis pradinius artinius). Nurodykite iteracijų pabaigos sąlygas. Lentelėje pateikite pradinį artinį, tikslumą, iteracijų skaičių.
   d. Gautus sprendinius patikrinkite naudodami išorinius išteklius (pvz., standartines MATLAB funkcijas).

Nr.	I lygčių sistema	II lygčių sistema	Metodas
10	$\begin{cases} \frac{x_1^2 + x_2^2}{5} - 2\cos\left(\frac{x_1}{2}\right) - 6\cos(x_2) - 8 = 0 \\ \left(\frac{x_1}{2}\right)^5 + \left(\frac{x_2}{2}\right)^4 - 4 = 0 \end{cases}$	$\begin{cases} 2x_2 + x_3 + 2x_4 - 14 = 0 \\ 3x_4^3 + 3x_2x_4 + 18 = 0 \\ -2x_1^2 + 5x_2^3 - 3x_3^2 - 485 = 0 \\ 5x_1 - 6x_2 + 3x_3 - 3x_4 - 11 = 0 \end{cases}$	Greičiausio nusileidimo

Pav. 2

Optimizavimas.

Pagal pateiktą uždavinio sąlygą (3 lentelė) sudarykite tikslo funkciją ir išspręskite jį vienu iš gradientinių metodų (gradientiniu, greičiausio nusileidimo, kvazi-gradientiniu, ar pan.). Gautą taškų konfigūraciją pavaizduokite programoje, skirtingais ženklais pavaizduokite duotus ir pridėtus (jei sąlygoje tokių yra) taškus. Ataskaitoje pateikite pradinę ir gautą taškų konfigūracijas, taikytos tikslo funkcijos aprašymą, taikyto metodo pavadinimą ir parametrus, iteracijų skaičių, iteracijų pabaigos sąlygas ir tikslo funkcijos priklausomybės nuo iteracijų skaičiaus grafiką.

Uždavinys 7-12 variantams
Plokštumoje ($-10 \leq \boldsymbol{x} \leq 10$, $-10 \leq \boldsymbol{y} \leq 10$) išsidėstę **n** taškų ($3 \leq \boldsymbol{n}$), vienas jų fiksuotas koordinačių pradžioje (0; 0). Kiekvienas taškas su visais kitais yra sujungtas tiesiomis linijomis (stygomis). Raskite tokias taškų koordinates, kad atstumas tarp taškų būtų kuo artimesnis vidutiniam atstumui, o stygų ilgių suma kuo geriau atitiktų nurodytą reikšmę **S** ($10 \leq \boldsymbol{S}$).

Pav. 3

2. PAGRINDINĖ DALIS

2.1 Tiesinės lygčių sistemos sprendimas

Užduotyje nurodyta lygčių sistema:

$$\begin{cases} 4x_1 + x_2 + x_3 + 7x_4 = 148 \\ x_1 + 2x_3 - 2x_4 = -37 \\ 2x_1 + 2x_2 - 7x_3 + x_4 = 21 \\ 4x_1 + 14x_2 + 7x_3 = 53 \end{cases}$$

Sistema pertvarkoma į matricos pavidalą:

$$A = \left[\begin{array}{cccc|c} 4 & 1 & 1 & 7 & 148 \\ 1 & 0 & 2 & -2 & -37 \\ 2 & 2 & -7 & 1 & 21 \\ 4 & 14 & 7 & 0 & 53 \end{array}\right]$$

Sprendžiame sistemą Gauso-Žordano algoritmu. Pav. 4 pateiktos matricos išraiškos kiekvieno algoritmo žingsnio metu.

```
[[  4.   1.   1.   7. 148.]
 [  1.   0.   2.  -2. -37.]
 [  2.   2.  -7.   1.  21.]
 [  4.  14.   7.   0.  53.]]

[[  4.     1.     1.     7.   148.  ]
 [  0.    -0.25   1.75  -3.75 -74.  ]
 [  0.     1.5   -7.5   -2.5  -53.  ]
 [  0.    13.     6.    -7.   -95.  ]]

[[    4.       0.       8.      -8.    -148.  ]
 [    0.      -0.25     1.75    -3.75   -74.  ]
 [    0.       0.       3.     -25.    -497.  ]
 [    0.       0.      97.    -202.   -3943.  ]]

[[    4.        0.        0.       58.667  1177.333]
 [    0.       -0.25      0.       10.833   215.917]
 [    0.        0.        3.      -25.     -497.   ]
 [    0.        0.        0.      606.333 12126.667]]

[[    4.        0.        0.        0.        4.   ]
 [    0.       -0.25      0.        0.       -0.75 ]
 [    0.        0.        3.        0.        3.   ]
 [    0.        0.        0.      606.333 12126.667]]
```

Pav. 4

Kiekvieną eilutę padalinę iš įstrižainėje esančio elemento gauname lygties sprendinių vektorių:

$$x = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 20 \end{bmatrix}$$

Įsistatome lygties sprendinius į sistemą, kad patikrintume jų korektiškumą:

$$\begin{cases} 4+3+1+7*20=148 \\ 1+2-2*20=-37 \\ 2+2*3-7+20=21 \\ 4+14*3+7=53 \end{cases}$$

Gautus sprendinius dar kartą patikriname pasinaudodami „Symbolab“ įrankio pagalba:

$$\begin{bmatrix} 4a+b+c+7d=148 \\ a+2c-2d=-37 \\ 2a+2b-7c+d=21 \\ 4a+14b+7c=53 \end{bmatrix}$$

Pav. 5

The solutions to the system of equations are:

$a=1, c=1, b=3, d=20$

Pav. 6

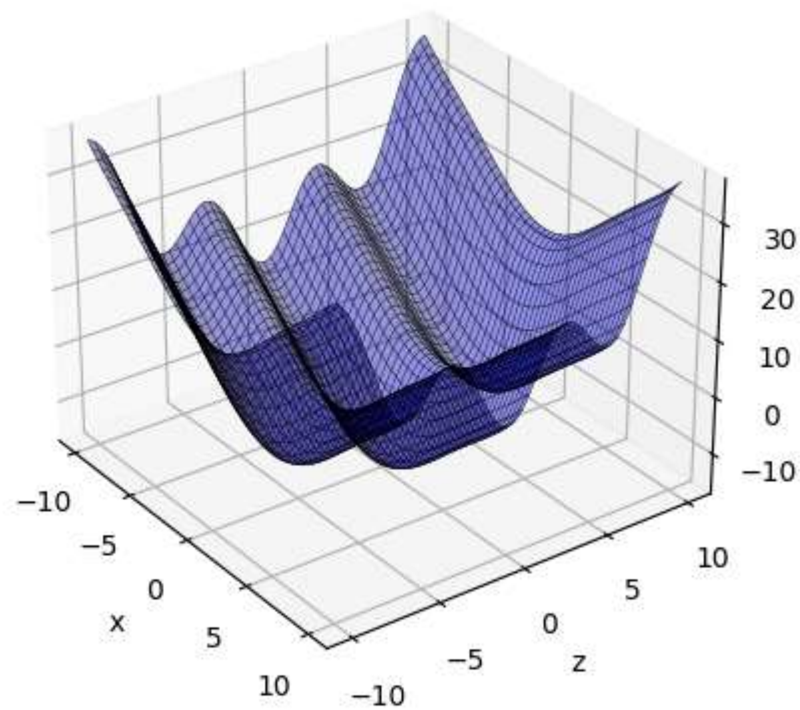
2.2 Netiesinių lygčių sistemų sprendimas

1. Duota dviejų netiesinių lygčių sistema:

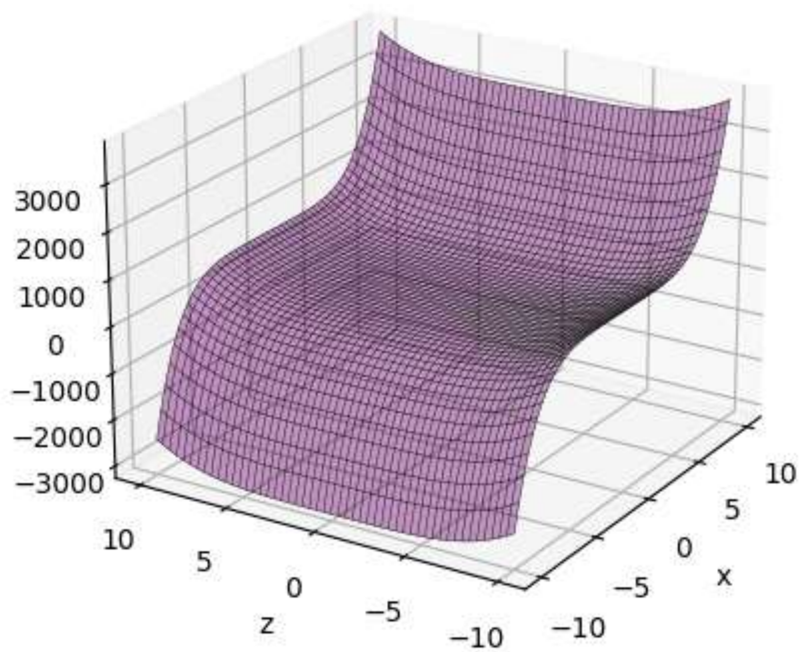
Nr.	I lygčių sistema
10	$\begin{cases} \frac{x_1^2+x_2^2}{5}-2\cos\left(\frac{x_1}{2}\right)-6\cos(x_2)-8=0 \\ \left(\frac{x_1}{2}\right)^5+\left(\frac{x_2}{2}\right)^4-4=0 \end{cases}$

Pav. 7

Skirtinguose grafikuose pavaizduojame funkcijų paviršius:

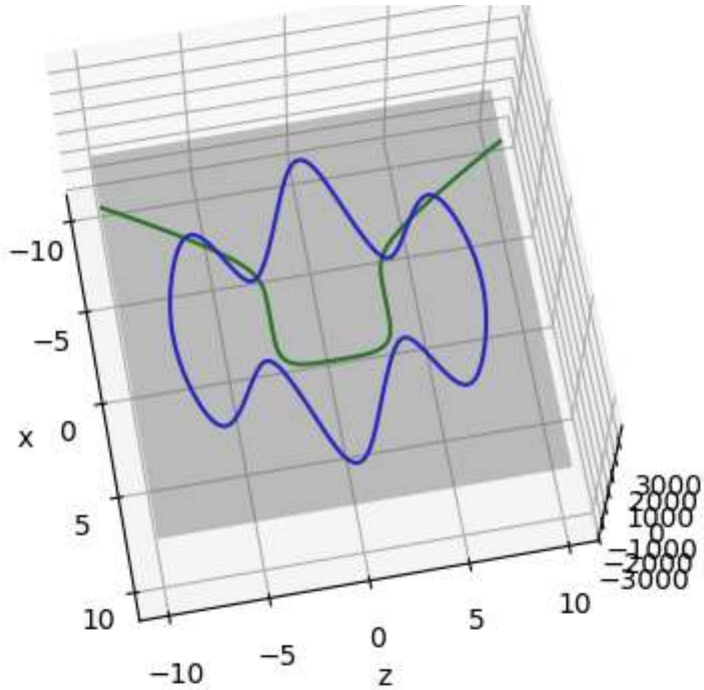


Pav. 8



Pav. 9

Lygčių sistema sprendžiame grafiniu būdu:



Pav. 10

Lygčių sistemą sprendžiame greičiausio nusileidimo metodu:

Pradinis artinys	**Tikslumas (funkcijų reikšmės)**	**Iteracijų skaičius**
[-1, 4]	$[10^{-13}, 10^{-15}]$	242
[-2, 4]	$[10^{-14}, 10^{-13}]$	265
[-1, 3]	$[10^{-13}, 10^{-14}]$	273
[-1, -1]	$[10^{-13}, 10^{-13}]$	600

Gautas sprendinys x = [-2.22182008 3.08919825]

Gautos šaknies patikrinimui tos pačios funkcijos šaknis randame „WolframAlpha“ įrankio pagalba:

Input interpretation:

solve

$$-8 + \frac{1}{5}\left(x^2 + y^2\right) - 2\cos\left(\frac{x}{2}\right) - 6\cos(y) = 0$$

$$-4 + \frac{x^5}{32} + \frac{y^4}{16} = 0$$

Result:

$x = -2.22182$ and $y = \pm 3.0892$

Pav. 11

2. Duota keturių netiesinių lygčių sistema:

II lygčių sistema

$$\begin{cases} 2x_2 + x_3 + 2x_4 - 14 = 0 \\ 3x_4^3 + 3x_2x_4 + 18 = 0 \\ -2x_1^2 + 5x_2^3 - 3x_3^2 - 485 = 0 \\ 5x_1 - 6x_2 + 3x_3 - 3x_4 - 11 = 0 \end{cases}$$

Pav. 12

Sprendžiame duotą sistemą greičiausio nusileidimo metodu naudojant žingsnio dydžio kitimą:

Pradinis artinys	**Tikslumas (funkcijų reikšmės)**	**Iteracijų skaičius**
[1, 1, 1, 1]	$[10^{-9}, 10^{-10}, 10^{-10}, 10^{-10}]$	252291

Gautas sprendinys:

x = [4. 5. 6. -1.]

Gautą sprendinį patikriname „WolframAlpha“ įrankio pagalba:

Input interpretation:

solve
$$-14 + 2k + 2y + z = 0$$
$$18 + 3k^3 + 3ky = 0$$
$$-485 - 2x^2 + 5y^3 - 3z^2 = 0$$
$$-11 - 3k + 5x - 6y + 3z = 0$$

Results:

$$x = 4 \wedge y = 5 \wedge z = 6 \wedge k = -1$$

Pav. 13

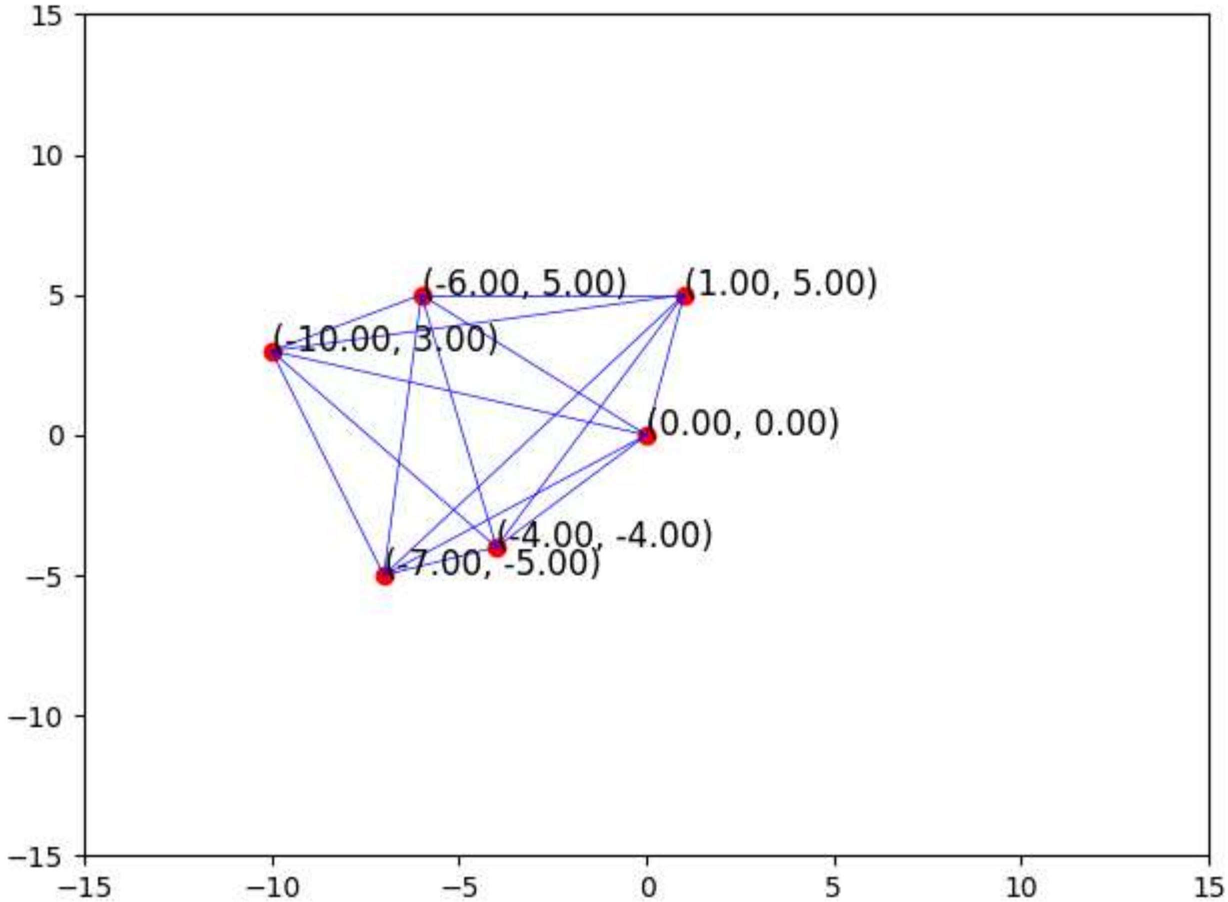
2.3 Optimizavimas

Duotas tekstinis uždavinys:

* **rekomenduojama** $\boldsymbol{n} \leq 20$, $\boldsymbol{m} \leq 20$ * (*papildymas*) Sudaryta struktūra turėtų būti analizuojama kaip pilnasis grafas, visų papildomai dedamų taškų pozicijos turėtų būti optimizuojamos vienu metu.
Uždavinys 7-12 variantams
Plokštumoje ($-10 \leq \boldsymbol{x} \leq 10$, $-10 \leq \boldsymbol{y} \leq 10$) išsidėstę **n** taškų ($3 \leq \boldsymbol{n}$), vienas jų fiksuotas koordinačių pradžioje (0; 0). Kiekvienas taškas su visais kitais yra sujungtas tiesiomis linijomis (stygomis). Raskite tokias taškų koordinates, kad atstumas tarp taškų būtų kuo artimesnis vidutiniam atstumui, o stygų ilgių suma kuo geriau atitiktų nurodytą reikšmę **S** ($10 \leq \boldsymbol{S}$).

Pav. 14

Programa sugeneruoja nurodytą skaičių taškų. Taškas (0,0) yra fiksuotas, jo pozicija nekeičiama. Pav. 15 parodyta sugeneruotų 6 taškų konfigūracija.



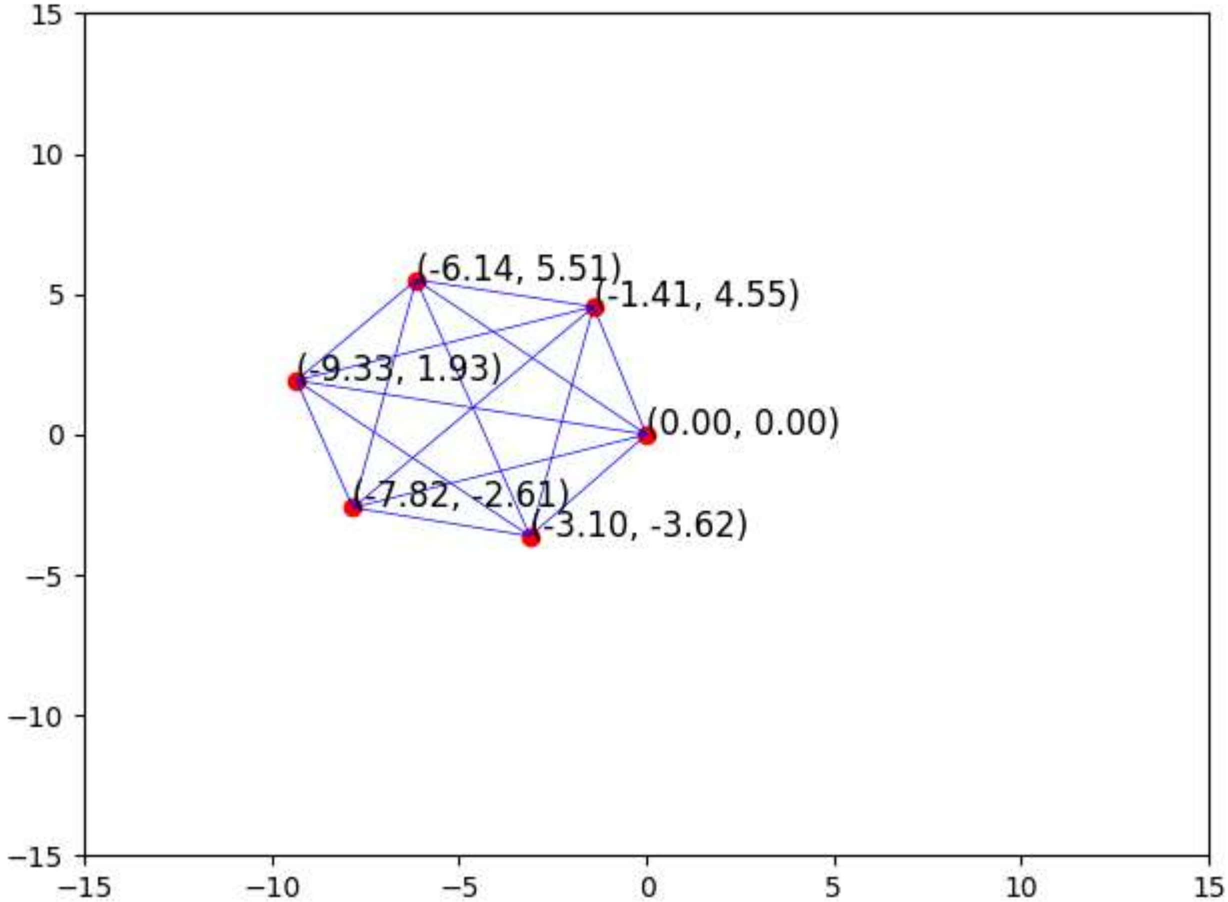
Pav. 15

Pasirinkta tikslo funkcija skaičiuoja briaunų ilgio nuokrypį nuo vidurkio ir stengiasi užtikrinti, kad briaunų ilgių suma būtų kuo artimesnė dydžiui S. Šiame pavyzdyje S=10. Pav. 16 pateiktas tikslo funkciją realizuojantis programinis kodas.

```
def loss(x, y, average, s):
    sum = 0
    n = len(x)
    _, length = calculate_parameters(x, y)
    for i in range(n):
        for j in range(i + 1, n):
            dist = ((x[j] - x[i]) ** 2 + (y[j] - y[i]) ** 2) ** 0.5
            sum += (dist - average) ** 2
    return sum + abs(length - s)
```

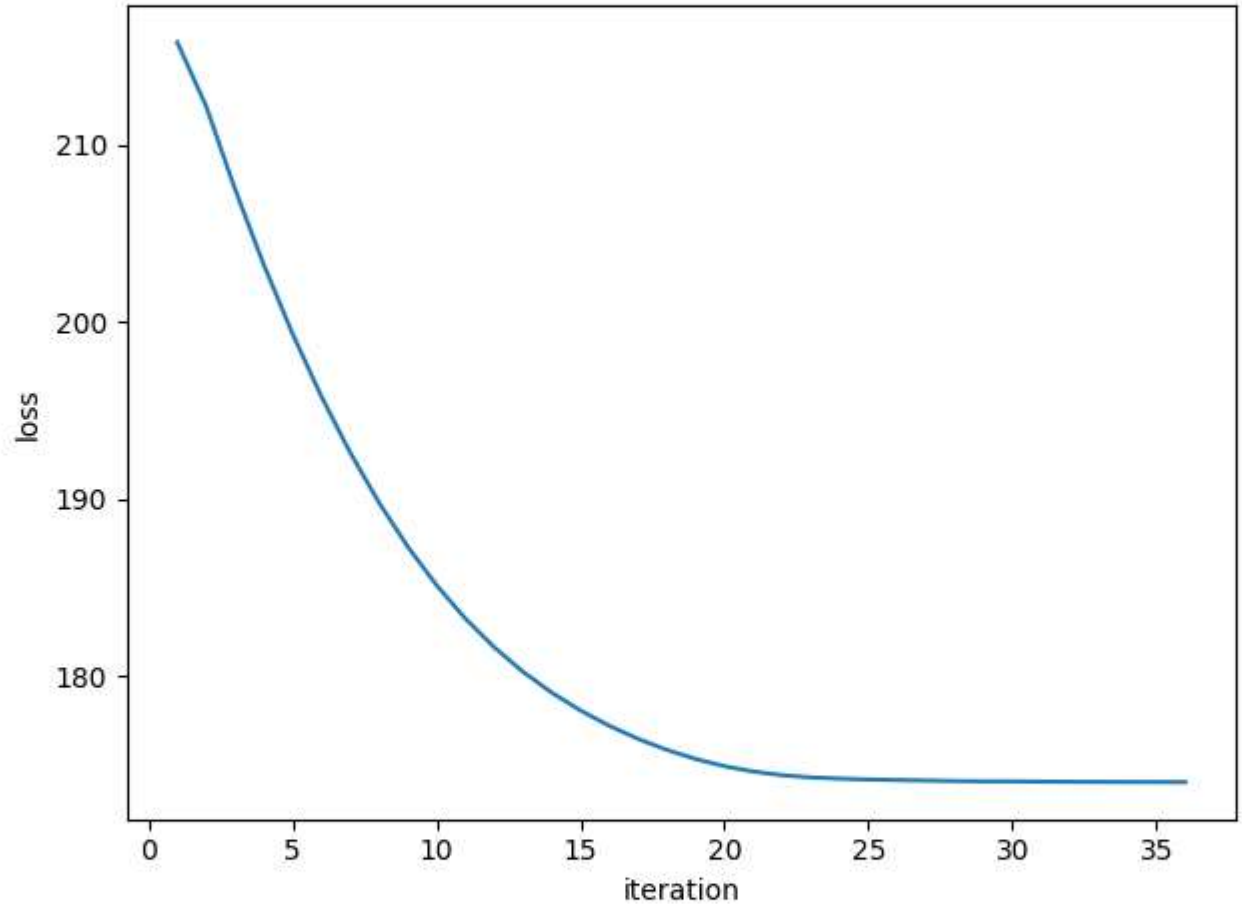
Pav. 16

Minimizavus tikslo funkciją gradientinio nusileidimo metodu gauname tokią taškų konfigūraciją:



Pav. 17

Pav. 18 pateikta tikslo funkcijos reikšmės priklausomybė nuo iteracijų skaičiaus algoritmo vykdymo metu.



Pav. 18

Naudojama iteracijų pabaigos sąlyga (čia x yra prieš tai aptartos tikslo funkcijos reikšmė, o $\varepsilon = 10^{-6}$):

$$\frac{\left\| \left\{ \mathbf{x} \right\}^{(k+1)} - \left\{ \mathbf{x} \right\}^{(k)} \right\|}{\left\| \left\{ \mathbf{x} \right\}^{(k+1)} \right\| + \left\| \left\{ \mathbf{x} \right\}^{(k)} \right\|} < \varepsilon$$

Pav. 19

3. IŠVADOS

Labai svarbu parinkti tinkamą pradinį artinį, nes nėra garantijų jog metodai konverguos.

Gradientą dalinant iš jo normos gaunamas vienetinis gradiento vektorius. Jį lengviau valdyti algoritme, tačiau prarandama automatiška gradiento korekcija, nes kai tikslo funkcija artėja prie minimumo - gradiento reikšmės artėja prie nulio, taip mažindamos žingsnio dydį.

Gradientiniai metodai gali „užstrigti“ lokaliame tikslo funkcijos minimume ir nerasti globalaus minimumo. Galimas sprendimas – keli skirtingi pradiniai artiniai.

Gauso-Žordano algoritmas yra netinkamas jeigu matrica yra singuliari, nes tada nebus aišku ar lygčių sistema neturi sprendinių, ar turi be galo daug sprendinių.

4. PRIEDAI

4.1 Realizuotų algoritmų programinio kodo fragmentai:

Greičiausio nusileidimo metodas:

```
def steepest_descent(func, alpha = 0.01):
    x = np.array([-1, 4])
    g = gradient(x)
    for i in range(601):
        prev_loss = loss(x)
        x = x - alpha * g

        if loss(x) > prev_loss:
            x = x + alpha * g   # reverse step
            g = gradient(x)     # recalculate gradient
            x = x - alpha * g   # step in new direction
        print(f'iteration: {i} loss: {loss(x)}')
        if loss(x) < 1e-25:     # good enough
            break
    print()
    print(f'function value: {func(x)}')
    print(f'x = {x}')
```

Gauso-Žordano metodas:

```
def gauss_jordan(A):
    mprint(A)
    n = A.shape[0]
    for i in range(n):
        for j in range(n):
            if i != j:
                ratio = A[j][i] / A[i][i]
                A[j][i:] = A[j][i:] - ratio * A[i][i:]
        mprint(A)

    # make 1's in diagonal and save result
    x = np.zeros(n)
    for i in range(n):
        x[i] = A[i][n] / A[i][i]
    return x
```

4.2 Nuoroda į programinio kodo repozitoriją

https://github.com/d0ubletr0uble/Skaitiniai-metodai-ir-algoritmai