informacijos
sistemų
katedra

KAUNO TECHNOLOGIJOS UNIVERSITETAS
Informatikos fakultetas

P170B115 Skaitiniai metodai ir algoritmai

Laboratorinis darbas Nr. 2

Variantas 1

Dėstytojas:
prof. BARAUSKAS Rimantas
Studentė:
Laura Capaitė IFK-0

KAUNAS, 2022

Turinys

1. Tiesinių lygčių sistemų sprendimas

Pirmo varianto užduotis (2.1 pav.)

Nr.	Metodai	Lygtys (žr. 2 lentelė)
1	QR skaidos, Gauso - Zeidelio	24, 27

24	$\begin{cases} 2x_1 + x_2 + x_3 + x_4 = 6 \\ x_1 + 3x_2 + x_3 - 3x_4 = -4 \\ x_1 + x_2 + 5x_3 + x_4 = 4 \\ 2x_1 + 3x_2 - 3x_3 - 2x_4 = 0 \end{cases}$
27	$\begin{cases} 3x_1 + x_2 - x_3 + 5x_4 = 8 \\ -3x_1 + 4x_2 - 8x_3 - x_4 = 10 \\ x_1 - 3x_2 + 7x_3 + 6x_4 = 11 \\ 5x_2 - 9x_3 + 4x_4 = 1 \end{cases}$

(2.1 pav. užduoties lygčių sistemos)

Pirmiausia užrašomos lygčių sistemos kaip matricos į A matrica, laisvieji nariai į B matrica. Gaunamos tokios matricos (2.2pav ir 2.3 pav.):

```
A24 = np.matrix([[2,1,1,1],[1,3,1,-3],[1,1,5,1],[2,3,-3,-2]], dtype=float);
B24 = np.matrix([[6],[-4],[4],[0]], dtype=float);
```

(2.2 pav. 24 varianto lygčių sistemos matricos)

```
A27 = np.array([[3,1,-1,5],[-3,4,-8,-1],[1,-3,7,6],[0,5,-9,4]], dtype=float);
B27 = np.array([[8],[10],[11],[1]], dtype=float);
```

(2.3 pav. 27 varianto lygčių sistemos matricos)

2.1 QR skaidos metodas

Užrašius lygčių sistemų matricas galima parašyti QR skaidos metodą (2.1.1 pav.) ir išbandyti veikimą

```
def qr_skaida(A, b):
  Ap = A;
  Q=np.identity(n)

  for i in range (0,n-1):
      z=A[i:n,i] # A matricos stulpelis
      zp=np.zeros(np.shape(z)); # atspindetas vektorius
      zp[0]=np.linalg.norm(z)
      omega=z-zp;
      omega=omega/np.linalg.norm(omega)
      Qi=np.identity(n-i)-2*omega*omega.transpose()
      A[i:n,:]=Qi.dot(A[i:n,:])
      Q[:,i:n]=Q[:,i:n].dot(Qi)
      print(A)

  # atgalinis etapas:
  b1=Q.transpose().dot(b);
  x=np.zeros(shape=(n,nb));
  for i in range (n-1,-1,-1):
      x[i,:]=(b1[i,:]-A[i,i+1:n]*x[i+1:n,:])/A[i,i];
  print(x)

  liekana=Ap.dot(x)-b1;print(liekana);
  print(np.linalg.norm(liekana)/ np.linalg.norm(x))
```

(2.1.1 pav. QR skaidos metodas)

Skaičiuojant lygčių sistemos rezultatą, kiekvienoje iteracijoje yra išvedama matrica, kaip ji atrodo po kiekvieno pertvarkymo. Kiekviename stulpelyje po įstrižaine turi atsirasti nuliai. Kai taip nutinka, ciklas pasibaigia ir yra atliekamas atgalinis etapas gauti x reikšmėms. Pabaigoj atliekamas patinkrinimas ar x reikšmės geros ir suskaičiuojama liekana.

24 varianto lygčių sistemos rezultatai:

Po visų iteracijų galima pastebėti, kad po įstrižaine susidarė nuliai (2.1.2 pav.). Pvz. skaičių *-3.92525577e-17* arba *2.28158224e-16* galima laykyti nuliais, nes jie yra labai maži.

```
Iteracija:  1
[[ 3.16227766  3.79473319  0.63245553 -1.26491106]
 [ 0.          0.59546855  1.31622777 -1.0513167 ]
 [ 0.         -1.40453145  5.31622777  2.9486833 ]
 [ 0.         -1.80906291 -2.36754447  1.8973666 ]]
Iteracija:  2
[[ 3.16227766e+00  3.79473319e+00  6.32455532e-01 -1.26491106e+00]
 [ 0.00000000e+00  2.36643191e+00 -1.01418511e+00 -3.46513244e+00]
 [ 0.00000000e+00 -2.07397409e-16  3.46800310e+00  1.03431274e+00]
 [ 0.00000000e+00 -1.02874939e-16 -4.74809257e+00 -5.68378663e-01]]
Iteracija:  3
[[ 3.16227766e+00  3.79473319e+00  6.32455532e-01 -1.26491106e+00]
 [ 0.00000000e+00  2.36643191e+00 -1.01418511e+00 -3.46513244e+00]
 [ 0.00000000e+00 -3.92525577e-17  5.87974732e+00  1.06904497e+00]
 [ 0.00000000e+00  2.28158224e-16 -1.43372980e-16 -5.00000000e-01]]
```

(2.1.2 pav. A matricos pertvarkymai)

Atlikus atgalini etapą x reikšmės gautos (2.1.3 pav.):

```
[[ 2.00000000e+00]
 [ 1.50129554e-15]
 [-2.26585864e-16]
 [ 2.00000000e+00]]
```

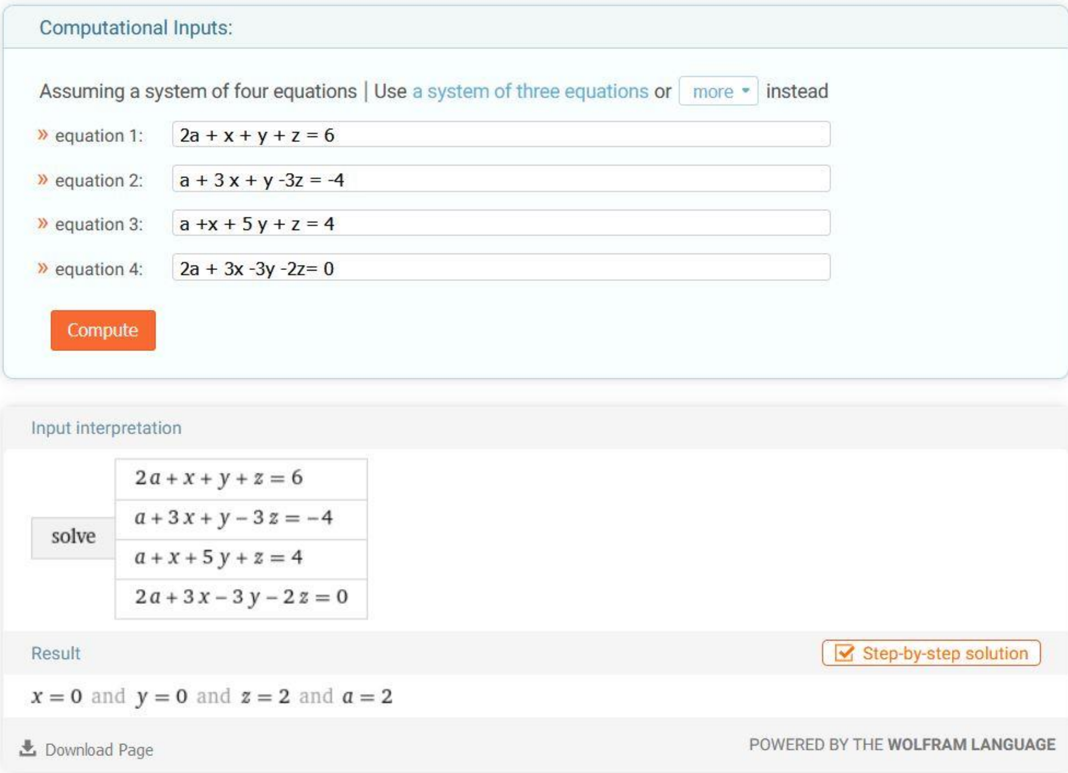
(2.1.3 pav. x reikšmės)

Paskaičiavus liekana (2.1.4 pav.)matoma, kad gautos x reikšmės yra teisingos

```
[[0.]
 [0.]
 [0.]
 [0.]]
```

(2.1.4 pav. liekana)

Taip pat patikrinus su *www.wolframalpha.com* puslapiu, x reikšmės (2.1.5 pav.) yra tokios pačios kaip ir QR skaidos metode rastos.



(2.1.5 pav. sprendiniai gauti su *www.wolframalpha.com*)

27 varianto lygčių sistemos rezultatai:

Po kiekvienos iteracijos po pagrindine įstrižaine susidaro nuliai (2.1.6 pav.), pradinis metodo veiksmas atliktas teisingai.

```
Iteracija:  1
[[ 4.35889894e+00 -2.75298881e+00  6.42364055e+00  5.50597761e+00]
 [-3.33066907e-16 -4.28535961e+00  8.38894618e+00  1.17031436e-01]
 [ 2.22044605e-16 -2.38213464e-01  1.53701794e+00  5.62765619e+00]
 [ 0.00000000e+00  5.00000000e+00 -9.00000000e+00  4.00000000e+00]]
Iteracija:  2
[[ 4.35889894e+00 -2.75298881e+00  6.42364055e+00  5.50597761e+00]
 [ 2.08577995e-16  6.58946528e+00 -1.23402713e+01  2.75559457e+00]
 [ 2.33909358e-16  1.38777878e-17  1.08294362e+00  5.68545402e+00]
 [-2.49036149e-16 -5.55111512e-16  5.30828189e-01  2.78684799e+00]]
Iteracija:  3
[[ 4.35889894e+00 -2.75298881e+00  6.42364055e+00  5.50597761e+00]
 [ 2.08577995e-16  6.58946528e+00 -1.23402713e+01  2.75559457e+00]
 [ 1.00423451e-16 -2.31865220e-16  1.20604538e+00  6.33173824e+00]
 [ 3.26569628e-16  5.04559118e-16 -1.91159888e-16 -1.54141541e-15]]
```

(2.1.6 pav. A matricos pertvarkymai)

Atlikus atgalini etapą x reikšmės gautos (2.1.7 pav.). Sprendiniai gaunami labai dideli.

```
[[-5.24281972e+00]
 [ 6.52668216e+16]
 [ 3.34293476e+16]
 [-6.36749479e+15]]
```

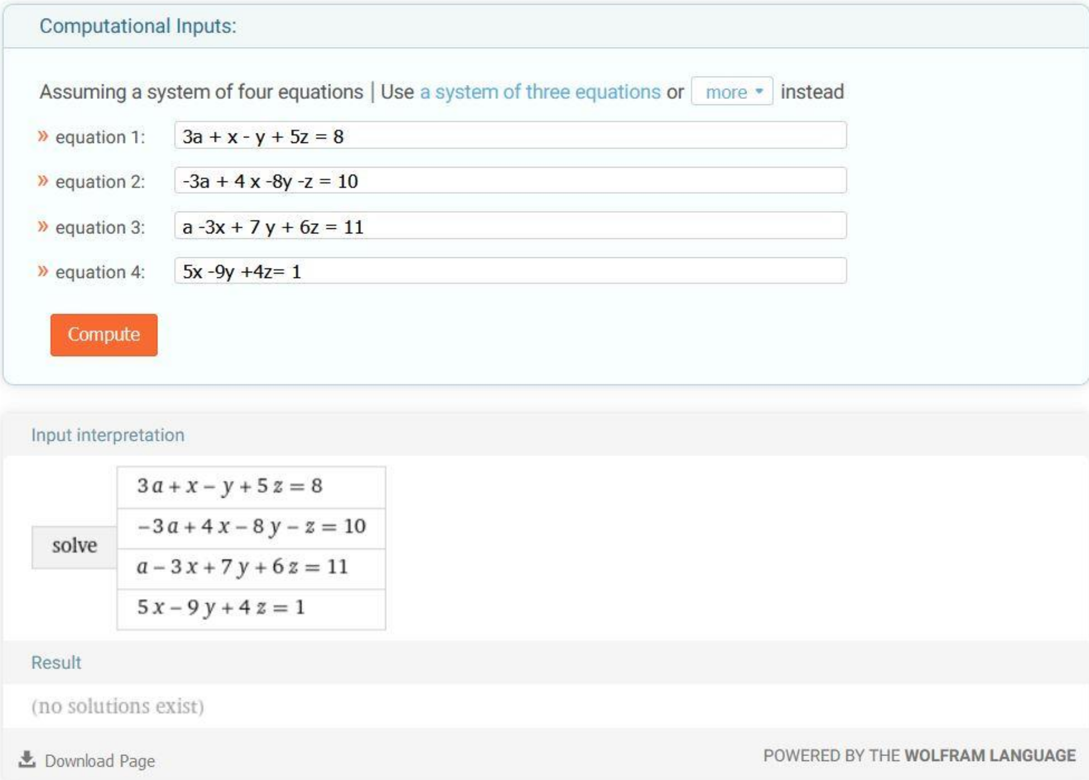
(2.1.7 pav. x reikšmės)

Paskaičiavus liekana (2.1.8 pav.)matoma, kad gautos reikšmės yra neteisingos, sprendiniai netinka.

```
[[ -1.14707867]
 [ -3.51438148]
 [-13.26649916]
 [ 26.54061958]]
```

(2.1.8 pav. liekana)

Patikrinus gaunama, kad matrica yra singuliari, tai reiškia, kad ši lygčių sistema neturi sprendinių.
Patikrinus su *www.wolframalpha.com* puslapiu (2.1.9 pav.), gaunama, kad lygčių sistema neturi sprendinių



(2.1.9 pav. sprendiniai gauti su *www.wolframalpha.com*)

QR metodas surado netikslias reikšmes. Jas įstačius į lygčių sistemą negaunami nuliai. Galima teigti, kad lygčių sistema sprendinių neturi.

2.2 Gauso – Zeidelio metodas

Gauso – Zeidelio metodas parodytas 2.2.1 pav. Parenkamos atitinkamos alpha reikšmės ir išbandomas metodas su lygčių sistema.

```
def gauso_zeidelio(A, b, n):
  Ap = A;
  alpha = np.array([1, 1, 1, -10]);
  Atld=np.diag(1./np.diag(A)).dot(A)-np.diag(alpha)
  btld=np.diag(1./np.diag(A)).dot(b)
  nitmax=1000;
  eps=1e-12
  x=np.zeros(shape=(n,1));
  x1=np.zeros(shape=(n,1));

  for it in range (0,nitmax):
    for i in range (0,n) :
      x1[i]=(btld[i]-Atld[i,:].dot(x1))/alpha[i];
    prec=(np.linalg.norm(x1-x)/(np.linalg.norm(x)+np.linalg.norm(x1)))
    if prec < eps : break
    x[:]=x1[:]

  print(x)
  liekana=Ap.dot(x)-b;
  print(liekana);
  print(np.linalg.norm(liekana)/ np.linalg.norm(x))
```

(2.2.1 pav. Gauso – Zeidelio metodas)

24 varianto lygčių sistemos rezultatai:

Pritaikius metodą 24 varianto lygčių sistemai gaunamos x reikšmės (2.2.2 pav.):

```
[[ 2.00000000e+00]
 [-8.05171485e-11]
 [ 1.46620494e-11]
 [ 2.00000000e+00]]
```

(2.2.2 pav. lygties sprendiniai)

-8.05171485e-11 ir *1.46620494e-11* reikšmės yra mažos ir artimos nuliui, todėl galima sakyti, kad tai lygu 0.

Patikrinus liekanas (2.2.3 pav.), jos taip pat yra labai mažos ir artimos nuliui, todėl galiam teigti, kad sprendiniai yra teisingi

```
[[ 5.90461013e-12]
 [-8.56026361e-12]
 [ 2.60858002e-12]
 [-5.73923131e-11]]
```

(2.2.3 pav. liekana)

Palyginus gautas x reikšmes su 2.1.5 pav. galima matyti, kad sprendiniai sutampa, metodas surado tinkamas x reikšmes

27 varianto lygčių sistemos rezultatai:

Pritaikius metodą 27 varianto lygčių sistemai gaunamos x reikšmės (2.2.4 pav.):

```
[[ 13.91945277]
 [172.9796834 ]
 [ 84.1345678 ]
 [-24.53168687]]
```

(2.2.4 pav. lygties sprendiniai)

Patikrinus liekanas (2.2.5 pav.), matoma, kad jos nesigauna artimos nuliui. Vadinasi, sprendiniai yra neteisingi.

```
[[-5.49604130e-02]
 [-8.38448021e+00]
 [-7.42677440e+01]
 [ 8.56055937e+00]]
```

(2.2.5 pav. liekana)

Galima matyti, kad sistema neturi sprendinių iš 2.1.9 pav. ir dėl to, kad matrica gaunama singuliari, todėl metodas neveikia.

2. Netiesinių lygčių sistemų sprendimas

Duota lygčių sistema ir metodas (3.1 pav.)

Nr.	Lygčių sistema	Metodas
1	$\begin{cases} \sin(x_1)\cos(x_2) + \frac{x_2}{4} - 0.5 = 0 \\ e^{-3x_1^2 - x_2^2 + 3} - 0.1 = 0 \end{cases}$	Broideno

(3.1 pav. duota lygčių sistema)

Pagal duotą sistemą užrašoma funkcija (3.2 pav.)

```
def LF(x):
  s=np.matrix( [[np.sin(x[0])+np.cos(x[1]) + (x[1]/4) - 0.5], [math.exp(-3*x[0]**2 - x[1]**2 + 3 ) -0.1]], dtype=float)
  return s
```

(3.2 pav. lygčių sistemos funkcija)

Norint naudoti Broideno metodą, pirmiausia reikia gauti Jakobio matricą (3.3 pav.). Ji gaunama iš funkcijos dabartiniame artinyje + priaugis reikšmė atėmus funkcijos reikšmę tame taške ir padalinus iš prieaugio reikšmės.

```
x = np.matrix(np.zeros(shape=(n,1)));
dx=0.1   # dx pradiniam Jakobio matricos iverciui
A=np.matrix(np.zeros(shape=(n,n)), dtype=float)

x1=np.zeros(shape=(n,1));
for i in range (0,n):
   x1=np.matrix(x);
   x1[i]+=dx;
   A[:,i]=(LF(x1)-LF(x))/dx
```

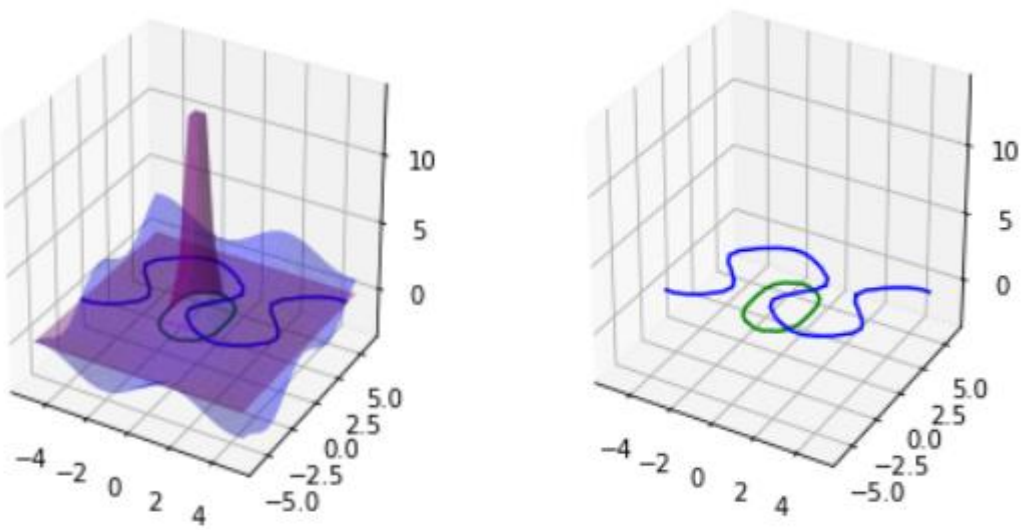
(3.3 pav. metodas gauti Jakobio matricai)

Turint Jakobio matricą, galima užrašyti Broideno metodą (3.4 pav.).

```
for i in range (1,maxiter):
    deltax=-np.linalg.solve(A,ff);
    x1=np.matrix(x+deltax);
    ff1=LF(x1)
    A+=(ff1-ff-A*deltax)*deltax.transpose()/(deltax.transpose()*deltax);
    tikslumas=np.linalg.norm(deltax)/(np.linalg.norm(x)+np.linalg.norm(deltax));
    ff= ff1;
    x=x1;
    if tikslumas < eps: break;
print(x)
```

(3.4 pav. Broideno metodas)

Nubraižomi funkcijų grafikai (3.5 pav.). Dešinėje esančiame grafike matoma, kuriose vietose funkcijos kertasi.



(3.5 pav. funkcijų grafikai)

Maksimalus iteracijų skaičius – 50
Pabaigos sąlyga - 1e-6

Pirmasis sprendinys:
Artinys: [2.5, 5.5]
Iteracija: 1 Tikslumas: 1.0
Iteracija: 2 Tikslumas: 0.747379607184177
Iteracija: 3 Tikslumas: 0.5193762156621007
Iteracija: 4 Tikslumas: 0.7426470589663657
Iteracija: 5 Tikslumas: 0.24814234651639971
Iteracija: 6 Tikslumas: 0.8174887526862238
Iteracija: 7 Tikslumas: 0.4907631714470798
Iteracija: 8 Tikslumas: 0.26770050432852666
Iteracija: 9 Tikslumas: 0.30355210422161283
Iteracija: 10 Tikslumas: 0.42021032868204217
Iteracija: 11 Tikslumas: 0.13462547835445424
Iteracija: 12 Tikslumas: 0.061931184773902075
Iteracija: 13 Tikslumas: 0.08486762574244631
Iteracija: 14 Tikslumas: 0.05152481581886595
Iteracija: 15 Tikslumas: 0.03322469229826943
Iteracija: 16 Tikslumas: 0.015561514754831427
Iteracija: 17 Tikslumas: 0.0047199747049550475
Iteracija: 18 Tikslumas: 0.0007137854165646514
Iteracija: 19 Tikslumas: 4.500054054018201e-05
Iteracija: 20 Tikslumas: 1.0484259029727696e-06
Iteracija: 21 Tikslumas: 8.001180317536842e-09
Rezultatas:
[[0.51595448]
[2.12225305]]
Įstačius į sistemą:
[[1.53987934e-12]
[1.49295437e-11]]

Artinys: [-1, -1]
Iteracija: 1 Tikslumas: 1.0
Iteracija: 2 Tikslumas: 0.7977471444831032
Iteracija: 3 Tikslumas: 0.48337757860093644
Iteracija: 4 Tikslumas: 0.4471625420942462
Iteracija: 5 Tikslumas: 0.34851915437796727
Iteracija: 6 Tikslumas: 0.31957062242358625
Iteracija: 7 Tikslumas: 0.07106484181854335
Iteracija: 8 Tikslumas: 0.09293718989905113
Iteracija: 9 Tikslumas: 0.04654565004188685
Iteracija: 10 Tikslumas: 0.031160145479848557
Iteracija: 11 Tikslumas: 0.013445244907554646
Iteracija: 12 Tikslumas: 0.0036106203889473575
Iteracija: 13 Tikslumas: 0.00041822934643260146
Iteracija: 14 Tikslumas: 1.921734709943846e-05
Iteracija: 15 Tikslumas: 6.214493311137967e-07
Rezultatas:
[[0.51595446]
[2.12225302]]

Įstačius į sistemą:
[[3.52825591e-09]
 [1.95891753e-08]]

Antrasis sprendinys:
Artinys: [0, 0]
Iteracija: 1 Tikslumas: 1.0
Iteracija: 2 Tikslumas: 0.7493672708974679
Iteracija: 3 Tikslumas: 0.5177351973752284
Iteracija: 4 Tikslumas: 0.6032419622991819
Iteracija: 5 Tikslumas: 0.4852557640499449
Iteracija: 6 Tikslumas: 0.044289501194456286
Iteracija: 7 Tikslumas: 0.09638872204444511
Iteracija: 8 Tikslumas: 0.016682430344035507
Iteracija: 9 Tikslumas: 0.07080011324599762
Iteracija: 10 Tikslumas: 0.358009842006891
Iteracija: 11 Tikslumas: 0.09165823470203295
Iteracija: 12 Tikslumas: 0.08444372485510142
Iteracija: 13 Tikslumas: 0.01811465127060997
Iteracija: 14 Tikslumas: 0.1857147955421958
Iteracija: 15 Tikslumas: 0.16783628049160557
Iteracija: 16 Tikslumas: 0.04128912419837121
Iteracija: 17 Tikslumas: 0.5925566486590855
Iteracija: 18 Tikslumas: 0.42554786510626164
Iteracija: 19 Tikslumas: 0.029015688163567956
Iteracija: 20 Tikslumas: 0.1359811245444435
Iteracija: 21 Tikslumas: 0.05889619832042835
Iteracija: 22 Tikslumas: 0.027224777321540074
Iteracija: 23 Tikslumas: 0.04306140523941992
Iteracija: 24 Tikslumas: 0.07907502927561405
Iteracija: 25 Tikslumas: 0.1442028447942039
Iteracija: 26 Tikslumas: 0.7136049766051897
Iteracija: 27 Tikslumas: 0.4949737180040975
Iteracija: 28 Tikslumas: 0.3011876608180538
Iteracija: 29 Tikslumas: 0.22880049304638903
Iteracija: 30 Tikslumas: 0.10132569649058004
Iteracija: 31 Tikslumas: 0.9377517141077231
Iteracija: 32 Tikslumas: 0.4932975558232879
Iteracija: 33 Tikslumas: 0.06675502766207375
Iteracija: 34 Tikslumas: 0.031008084106796806
Iteracija: 35 Tikslumas: 0.0163116093819996
Iteracija: 36 Tikslumas: 0.02995813470898467
Iteracija: 37 Tikslumas: 0.1953651653138275
Iteracija: 38 Tikslumas: 0.18080294920628032
Iteracija: 39 Tikslumas: 0.024200134430770334
Iteracija: 40 Tikslumas: 0.08554448380131731
Iteracija: 41 Tikslumas: 0.025312053587497368
Iteracija: 42 Tikslumas: 0.016113173568057006
Iteracija: 43 Tikslumas: 0.038877719383605486
Iteracija: 44 Tikslumas: 0.04673784820694688
Iteracija: 45 Tikslumas: 0.13584468378756723
Iteracija: 46 Tikslumas: 0.11289294353119075

Iteracija: 47 Tikslumas: 0.057284656739077185
Iteracija: 48 Tikslumas: 0.9854998446208878
Iteracija: 49 Tikslumas: 0.498982218717454
Iteracija: 50 Tikslumas: 0.07897715642532427
Iteracija: 51 Tikslumas: 0.5049613742651711
Iteracija: 52 Tikslumas: 0.4780471407224297
Iteracija: 53 Tikslumas: 0.06454823622686946
Iteracija: 54 Tikslumas: 0.006589793075446307
Iteracija: 55 Tikslumas: 0.017092254901593787
Iteracija: 56 Tikslumas: 0.4760879756329363
Iteracija: 57 Tikslumas: 0.37186205123797883
Iteracija: 58 Tikslumas: 0.19643362161802752
Iteracija: 59 Tikslumas: 0.5555912390855491
Iteracija: 60 Tikslumas: 0.31982020969442093
Iteracija: 61 Tikslumas: 0.2179786304294072
Iteracija: 62 Tikslumas: 0.45169775953825186
Iteracija: 63 Tikslumas: 0.3959737309793239
Iteracija: 64 Tikslumas: 0.03542274757312486
Iteracija: 65 Tikslumas: 0.015191811335489653
Iteracija: 66 Tikslumas: 0.002807751220578613
Iteracija: 67 Tikslumas: 0.00019892592505327527
Iteracija: 68 Tikslumas: 2.072493893699562e-05
Iteracija: 69 Tikslumas: 2.5892799741645755e-06
Iteracija: 70 Tikslumas: 4.0750114379944835e-08
Rezultatas:
[[0.99484194]
 [-1.5275646]]
Įstačius į sistemą:
[[3.10032000e-11]
 [2.18872698e-11]]

Artinys: [2.5, 2.5]
Iteracija: 1 Tikslumas: 1.0
Iteracija: 2 Tikslumas: 0.4896531208199483
Iteracija: 3 Tikslumas: 0.9573771175356403
Iteracija: 4 Tikslumas: 0.19717068093777743
Iteracija: 5 Tikslumas: 0.025617026735981915
Iteracija: 6 Tikslumas: 0.04767335301494929
Iteracija: 7 Tikslumas: 0.05405211727049381
Iteracija: 8 Tikslumas: 0.03245403420511992
Iteracija: 9 Tikslumas: 0.015951868950436095
Iteracija: 10 Tikslumas: 0.005156504414552624
Iteracija: 11 Tikslumas: 0.000851998756387261
Iteracija: 12 Tikslumas: 4.922286274849974e-05
Iteracija: 13 Tikslumas: 5.013052611143138e-07
Rezultatas:
[[0.99484194]
 [-1.5275646]]
Įstačius į sistemą:
[[9.68760627e-11]
 [-7.75707429e-10]]

Panaudojus po du skirtingus artinius kiekvienam sprendiniui rezultatai gauti teisingi. Įstačius x reikšmes į lygčių sistemą gaunami skaičiai artimi nuliui. Pirmasis sprendinys yra [0.51595446, 2.12225302], antrasis - [0.99484194, -1.5275646].

Išvados

Pirmai užduočiai reikėjo panaudoti QR skaidos ir Gauso – Zeiderio metodus. Pirmajai lygčių sistemai pavyko surasti sprendinius sėkmingai, su antrąja kilo šiek tiek problemų. Patikrinus matrica gaunama, kad ji yra singuliari ir neturi sprendinių.

Antrai užduočiai reikėjo panaudoti Broideno metodą. Kilo truputėlį sunkumų randant pradinius artinius. Pirmąjį sprendinį buvo gana lengva rasti, antrąjį teko šiek tiek paieškoti keičiant pradinius artinius.

Trečios užduoties nepavyko padaryti.