****

**Kauno technologijos universitetas**

Informatikos fakultetas

**1 laboratorinis darbas**

Skaitiniai metodai ir algoritmai (P170B115)

|  |  |
| --- | --- |
|  |  |
| **Simona Ragauskaitė IFF-0/3 gr.**  Studentė | (parašas) (data) |
| **Doc. Andrius Kriščiūnas** |  |
| **Prof. Rimantas Barauskas**  Dėstytojai | (parašas) (data) |
|  |  |

**Kaunas, 2022**

Turinys

[1. Darbo užduotis 3](#_Toc113566779)

[1.1. Išspręsti netiesines lygtis: a) daugianaris f(x) = 0, b) transcendentinė funkcija g(x) = 0; 3](#_Toc113566780)

[1.2. Pagal pateiktą uždavinio sąlygą sudaryti netiesinę lygtį su pasirinktu skaitiniu metodu ją išspręsti. 3](#_Toc113566781)

[2. Užduoties sprendimas 4](#_Toc113566782)

[2.1. Daugianario „grubaus“ ir „tikslesnio“ šaknų intervalų nustatymas 4](#_Toc113566783)

[2.2. Daugianario ir transcendentinės funkcijų grafikai 5](#_Toc113566784)

[2.3. Skenavimo algoritmo pritaikymas 6](#_Toc113566785)

[2.4. Tekstinio uždavinio sprendimas 6](#_Toc113566786)

[3. Išvados 7](#_Toc113566787)

[4. Programinio kodo fragmentai 8](#_Toc113566788)

# Darbo užduotis

2 variantas

## Išspręsti netiesines lygtis: a) daugianaris f(x) = 0, b) transcendentinė funkcija g(x) = 0;

Table

Description automatically generated

## Pagal pateiktą uždavinio sąlygą sudaryti netiesinę lygtį su pasirinktu skaitiniu metodu ją išspręsti.

Table

Description automatically generated

Table

Description automatically generated

# Užduoties sprendimas

## Daugianario „grubaus“ ir „tikslesnio“ šaknų intervalų nustatymas

Daugianario „grubus“ šaknų įvertis

Daugianaris:

„Grubaus“ įverčio formulė:

Kai: ir .

Norėdama pritaikyti šią formulę turiu pasidauginti iš (–1), kadangi .

Intervalų skaičiavimui naudojami koeficientai:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a4 | a3 | a2 | a1 | a0 |
| 1.35 | 0.93 | -26.46 | -16.20 | 76.19 |

Įsistačius koeficientus gaunu:

Gaunu tokį grubios šaknies įvertį:.

Daugianario „tikslesnis“ šaknų įvertis

„Tikslesnio“ įverčio formules:

, kai

Skaičiuoju tikslesnę teigiamo įverčio reikšmę:

Intervalų skaičiavimui naudojami koeficientai:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a4 | a3 | a2 | a1 | a0 |
| 1.35 | 0.93 | -26.46 | -16.20 | 76.19 |

Skaičiuoju tikslesnę neigiamo įverčio reikšmę:

Intervalų skaičiavimui naudojami koeficientai:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a4 | a3 | a2 | a1 | a0 |
| 1.35 | -0.93 | -26.46 | 16.20 | 76.19 |

Chart, line chart

Description automatically generated Chart, line chart, histogram

Description automatically generated

„Grubaus“ intervalo grafikas „Tikslesnio“ intervalo grafikas

## Daugianario ir transcendentinės funkcijų grafikai

Daugianaris: | : (–1)

Grafiko šiek tiek pakeičiau ašis, intervale -4 < x < -2.25, grafike aiškiai yra matyti daugianario šaknys.

Chart, line chart

Description automatically generated

Funkcijos f(x) grafikas, -4 < x < -2.25 intervale

Transcendentinė funkcija:

Chart, line chart

Description automatically generated

Funkcijos g(x) grafikas pagal duotąjį intervalą

## Skenavimo nekintančiu žingsniu algoritmo pritaikymas

Šaknų intervalams surasti naudosiu skenavimo nekintančių žingsnių metodą. Žingsnį gausiu padalinus visą „tikslesnį“ intervalą iš 100. Tada eisiu per visą intervalą ir tikrinsiu ar dabartinio x ženklas nesutampa su x + žingsnio ženklu. Jei nesutampa, tuomet reiškia, kad intervale yra šaknis.

def getInterval(start, end):

step = (end - start) / 100

x = start

intervals = []

while x < end:

x\_next = x + step

if (f(x) > 0 > f(x\_next)) or (f(x\_next) > 0 > f(x)):

intervals.append([x, x\_next])

x = x\_next

return intervals

Daugianario funkcijos gautieji intervalai:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Daugianario gautieji intervalai, kuriuose yra funkcijos šaknis | | | | |
| Intervalo pradžia | -4.2029 | -2.3801 | 1.5231 | 3.8656 |
| Intervalo pabaiga | -3.9426 | -2.1207 | 1.7834 | 4.1258 |

Pasiimu pirmą daugianario intervalą: , jame yra ryškiai matoma ieškomoji šaknis.

Chart, line chart

Description automatically generated

Tą patį metodą pritaikau transcendentinei funkcijai.

Transcendentinės funkcijos gautieji intervalai:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Daugianario gautieji intervalai, kuriuose yra funkcijos šaknis | | | | | | |
| Intervalo pradžia | 1.36 | 3.69 | 4.15 | 6.49 | 7.66 | 9.46 | |
| Intervalo pabaiga | 1.45 | 3.79 | 4.24 | 6.58 | 7.75 | 9.55 | |

Pasiimu pirmą transcendentinės funkcijos intervalą: intervalą, jame yra ryškiai matoma ieškomoji šaknis.

Chart, line chart

Description automatically generated

## Netiesinių lygčių sprendimo metodai

Skenavimo nekintančiu žingsniu metodo dėka atsiskyriau daugianario ir funkcijos šaknis. Jas tikslinsiu trimis metodais: 1 – stygų, 3 – Niutono(liestinių) ir 5 – skenavimo su mažėjančiu žingsniu.

Metodų kodai:

1 – Stygų

def sign(number):

if (number > 0):

return 1

if (number < 0):

return -1

return 0

def f\_derivative(x):

return -27\*x\*\*3 / 5 - 279\*x\*\*2 / 100 + 1323\*x / 25 + 81/5

def absolute\_terminate(function, x\_mid, epsilon=1e-6):

return abs(function(x\_mid)) < epsilon

def ChordMethod(x\_n, x\_n1, iteration=1):

if (x\_n > x\_n1):

raise Exception("Incorrect function usage")

k = abs(funkcija.f(x\_n) / funkcija.f(x\_n1))

x\_mid = (x\_n + k \* x\_n1) / (1 + k)

if (funkcija.f(x\_mid) == 0 or absolute\_terminate(funkcija.f, x\_mid)):

return x\_mid, iteration

if (sign(funkcija.f(x\_mid)) == sign(funkcija.f(x\_n))):

return ChordMethod(x\_mid, x\_n1, iteration + 1)

return ChordMethod(x\_n, x\_mid, iteration + 1)

3 – Niutono (liestinių)

def discrete\_method\_approx(f, x, h=.001):

return (f(x + h) - f(x)) / h

def NewtonsMethod(f, x, tolerance=.001):

count = 0

while abs(f(x)) > tolerance:

df = discrete\_method\_approx(f,x)

x = x - f(x) / df

count += 1

return x, count

5 – Skenavimo su mažėjančiu žingsniu

def scanDecreasing(x1, x2):

zingsnis = (x2 - x1) / 100

while np.abs(funkcija.f(x1)) > 0.001 and x1 + zingsnis <= x2:

if np.sign(funkcija.f(x1)) != np.sign(funkcija.f(x1 + zingsnis)):

zingsnis /= 2

continue

x1 += zingsnis

return x1

Daugianarės funkcijos rezultatai pritaikius metodus:

A picture containing scatter chart

Description automatically generated

A picture containing scatter chart

Description automatically generated

A picture containing chart

Description automatically generated

Transcendentinės funkcijos rezultatai pritaikius metodus:

A picture containing text

Description automatically generatedA picture containing table

Description automatically generatedA picture containing text

Description automatically generated

Mažiausią iteracijų skaičių turi Niutono (liestinių) metodas, tačiau tiksliausias metodas gavosi stygų.

## Rezultatų patikrinimas naudojant išorinius išteklius

Šaknų reikšmes tikrinu <https://www.desmos.com/calculator> tinklapyje, nes rekomenduotame tinklapyje wolframalpha.com transcendentinės funkcijos šaknų nepateikia.

Daugianarės funkcijos rezultatai:

Diagram

Description automatically generated

Transcendentinės funkcijos rezultatai:

Chart

Description automatically generated

## Tekstinio uždavinio sprendimas

Vertikaliai į viršų iššauto objekto greitis užrašomas dėsniu: , čia , pasipriešinimo koeficientas c, pradinis greitis v0 . Kokia objekto masė, jeigu žinoma, kad laiko momentu t1 objekto greitis buvo lygus v1?

Susidarau netiesinę lygtį: . Ją spręsiu skenavimo su mažėjančiu žingsniu metodu (nes nereikia išvestinės ir nėra didelis metodas).

Metodo žingsnis – 0,01

Intervalo pradžia – 0,5

Intervalo pabaiga – 1,5

Surasta šaknis – 0,9797265625000003

Surastos šaknies tikslumas – 3,638442079401116e-05

Chart, line chart

Description automatically generated

Tekstinio uždavinio funkcijos grafikas

# Išvados

Pagal gautus šaknų rezultatus, galiu teigti:

* Iš visų trijų metodų Niutono (liestinės) metodas yra greičiausias, nes mažiausias iteracijų skaičius. Tačiau yra trūkumas, nes reikia turėti funkcijos išvestinę.
* Tiksliausius rezultatus davė stygų metodas. Ir iteracijų skaičiumi nebuvo labai nutolęs nuo Niutono (liestinės) metodo.
* Skenavimas su mažėjančiu skaičiumi turėjo didžiausius iteracijų skaičius bei nebuvo.