

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**Skaitiniai metodai ir algoritmai**

(P170B115)

**Laboratorinis darbas Nr. 1**

Varianto Nr. 15

**Atliko:**

IFF 1/8 gr. Studentas

Matas Palujanskas

**Priėmė:**

prof. BARAUSKAS Rimantas

doc. KRIŠČIŪNAS Andrius

**KAUNAS, 2023**

# TURINYS

[1. TURINYS 2](#_Toc146135561)

[2. ĮVADAS 3](#_Toc146135562)

[2.1. Tikslas 3](#_Toc146135563)

[2.2. Variantas 3](#_Toc146135564)

[3. NETIESINIŲ LYGČIŲ PRAKTINIS SPRENDIMAS 3](#_Toc146135565)

[3.1. Daugianaris f(x) 4](#_Toc146135566)

[3.1.1. Uždavinys 4](#_Toc146135567)

[3.1.2. Sprendimas 4](#_Toc146135568)

[3.1.3. Dalinė išvada 6](#_Toc146135569)

[3.2. Funkcija g(x) 6](#_Toc146135570)

[3.2.1. Uždavinys 6](#_Toc146135571)

[3.2.2. Sprendimas 7](#_Toc146135572)

[3.2.3. Dalinė išvada 8](#_Toc146135573)

[3.3. Praktinė užduotis 8](#_Toc146135574)

[3.3.1. Uždavinys 8](#_Toc146135575)

[3.3.2. Sprendimas 9](#_Toc146135576)

[3.3.3. Dalinė išvada 10](#_Toc146135577)

[4. Programų kodai 10](#_Toc146135578)

[4.1. Iteracinis metodas 10](#_Toc146135579)

[4.2. Skenavimo metodas 11](#_Toc146135580)

[4.3. Niutono metodas 12](#_Toc146135581)

[4.4. Praktinio uždavinio metodas 12](#_Toc146135582)

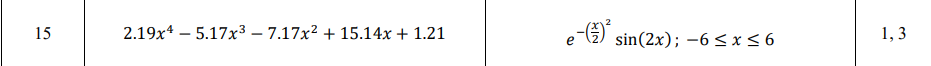
[5. Išvados 12](#_Toc146135583)

# ĮVADAS

## Tikslas

Darbo tikslas - rasti vienos netiesinės lygties šaknis naudojant šaknų atskyrimo (grafinį, skenavimo) ir jų tikslinimo (Niutono, kirstinių, paprastųjų iteracijų, stygų, pusiaukirtos) metodus. Programuojant skaitinius šaknų nustatymo metodus išmokstama parinkti pradinius sprendinių artinius (intervalus), skaičiavimo pabaigos sąlygas, vizualizuoti algoritmų vykdymo eigą. Gautų sprendinių patikrinimui naudojamos išorinių išteklių funkcijos.

## Variantas



Mano:

Daugianaris: 2.19𝑥4 − 5.17𝑥3 − 7.17𝑥2 − 15.14𝑥 + 1.21

Funkcija:

Šios funkcijos bus sprendžiamos šiais metodais:

1. Stygų;
2. Niutono (liestinių).

# NETIESINIŲ LYGČIŲ PRAKTINIS SPRENDIMAS

## Daugianaris f(x)

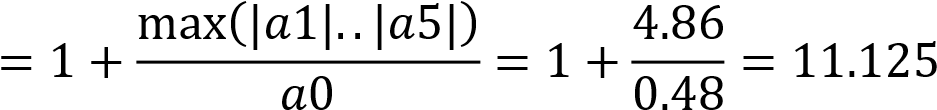
### Uždavinys

𝑓(𝑥) = 2.19𝑥4 − 5.17𝑥3 − 7.17𝑥2 − 15.14𝑥 + 1.21

### Sprendimas

Norint pritaikyti tam tikrus metodus daugianariui, mums būtina žinoti kokius rėžius norime panaudoti, juos gauti panaudosime “grubų” ir tikslesnį įverčius.

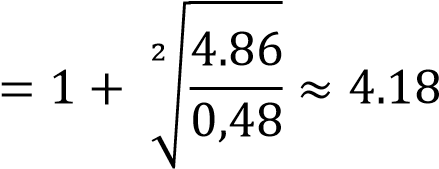
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a5 | a4 | a3 | a2 | a1 | a0 |
| 0.48 | 1.71 | -0.67 | -4.86 | -1.33 | 1.50 |

𝑅 

“Grubus” įvertis (-11.125 ; 11.125)

𝐵 = max(|𝑎3|, |𝑎2|, |𝑎1|) = 4.86

𝑘 = 5 − max(3,2,1) = 5 − 3 = 2

𝑅𝑡𝑒𝑖𝑔 

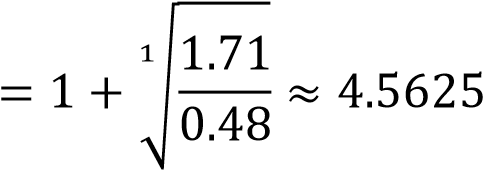
Apsiskaičiuojame atvirkštinę funkciją iš f(x) į -f(-x).

−𝑓(−𝑥) = −(0.48(−𝑥5) + 1.71(−𝑥4) − 0.67(−𝑥3) − 4.86(−𝑥2) − 1.33(−𝑥) + 1.50 = 0.48𝑥5 − 1.71𝑥4 − 0.67𝑥3 + 4.86𝑥2 − 1.33𝑥 − 1.50)

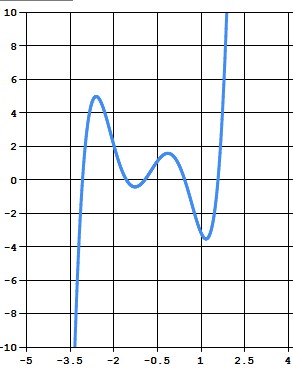
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a5 | a4 | a3 | a2 | a1 | a0 |
| 0.48 | -1.71 | -0.67 | 4.86 | -1.33 | -1.50 |

𝐵 = max(|𝑎4|, |𝑎3|, |𝑎1|, |𝑎0|) = 1.71

𝑘 = 5 − max(4,3,2,0) = 1

𝑅𝑛𝑒𝑖𝑔 

Gauti rėžiai −4.5625 ≤ 𝑥 ≤ 4.18



*1 pav. f(x) funkcija*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metodas | Rėžio pradžia | Rėžio pabaiga | Šaknis (x) | Tikslumas f(x) | Iteracijų kiekis | Konstanta a |
| Paprastųjų iteracijų | -3.5 | -3 | 3.0765025 | 0.0000001 | 20 | -50 |
| -4.5 | -1 | 1.5657301 | 0.0000001 | 18 | 5 |
| -3.5 | 1.5 | 0.9378466 | 0.0000001 | 23 | -5 |
| 0 | 2 | 0.4379006 | 0.0000002 | 7 | 5 |
| 1 | 2 | 1.5796779 | 0.0000002 | 26 | -50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metodas | Rėžio pradžia | Rėžio pabaiga | Šaknis (x) | Tikslumas f(x) | Iteracijų kiekis | Išvestinės reikšmė f’(x) |
| Niutono  (liestinių) | -4 | -2 | -3.07650233271371 | 2.22044604925031E-14 | 7 | 25.3783005466598 |
| -2 | -1 | -1.56572999771624 | 9.41469124882133E-14 | 5 | -2.8695516430588 |
| -1 | 1 | -0.937846428128933 | 9.7431063217357E-10 | 3 | 2.23240804510712 |
|  | 1 | 2 | 0.437900625276267 | 8.95526097721699E-10 | 5 | -5.30921764576956 |
| 2 | 4.18 | 1.5796781338877 | 1.77635683940025E-15 | 6 | 20.2070654194123 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metodas | Rėžio pradžia | Rėžio pabaiga | Šaknis (x) | Tikslumas f(x) | Iteracijų kiekis | Pradinis žingsnis |
| Skenavimo | -4.5 | 4.18 | -3.07650235 | 4.38696758919832E-07 | 7 | 0.5 |
| -1.5657303 | 8.67419406525016E-07 | 8 | 0.5 |
| -0.937846800000002 | 8.29193367080094E-07 | 7 | 0.5 |
| 0.437900449999998 | 9.29684234041162E-07 | 8 | 0.5 |
| 1.5796781 | 6.84770949277436E-07 | 8 | 0.5 |

Šaknys pagal wolframalpha.com

|  |  |
| --- | --- |
| Pirma šaknis | -3.0765 |
| Antra šaknis | -1.56573 |
| Trečia šaknis | -0.937846 |
| Ketvirta šaknis | 0.437901 |
| Penkta šaknis | 1.57968 |

### Dalinė išvada

Geriausias tikslumas buvo pasiektas iteraciniu niutono (liestinių) metodu, taip pat buvo sunaudota atlikta mažiausiai iteracijų. Prasčiausiai pasirodė paprastųjų iteracijų metodas, tikslumas nebuvo toks geras bei iteracijų kiekis buvo didžiausias. Su skaitiniais metodais rastos šaknys buvo tikslesnės negu wolframalpha.com.

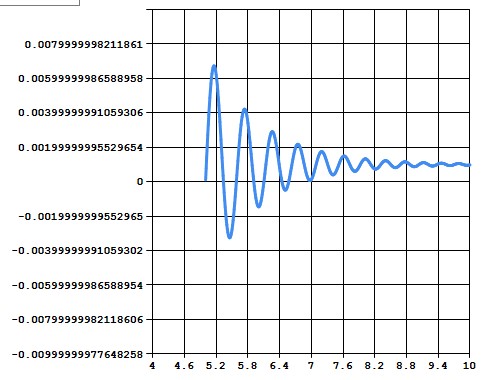
## Funkcija g(x)

### Uždavinys

𝑓(𝑥) = 𝑒−𝑥 sin(𝑥2) + 0.001;

Šiam uždaviniui rėžius suteikia sąlyga. 5 ≤ 𝑥 ≤ 10

### Sprendimas



*2 pav. g(x) funkcija*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metodas | Rėžio pradžia | Rėžio pabaiga | Šaknis (x) | Tikslumas f(x) | Iteracijų kiekis | Konstanta a |
| Paprastųjų iteracijų | 5 | 6 | 5.3370173 | 0.0000025 | 7 | 0.05 |
| 5 | 6 | 5.5809835 | 0.0000006 | 9 | -0.05 |
| 5.8 | 6.2 | 5.9106160 | 0.0000002 | 18 | 0.05 |
| 5.8 | 6.2 | 6.1021241 | 0.0000002 | 21 | -0.05 |
| 6 | 7 | 6.4436463 | 0.0000001 | 19 | 0.01 |
| 5 | 10 | 6.5716313 | 0.0000002 | 36 | -0.009 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metodas | Rėžio pradžia | Rėžio pabaiga | Šaknis (x) | Tikslumas f(x) | Iteracijų kiekis | Išvestinės reikšmė f’(x) |
| Niutono  (liestinių) | 5.3 | 5.5 | 5.33701723316268 | 2.08729518044737E-13 | 3 | 0.0492224329799163 |
|  | 5.5 | 5.8 | 5.58098358855646 | 1.64091440088554E-12 | 4 | 0.0415600158557291 |
| 5.8 | 6.2 | 5.91061589848537 | 4.07226335985555E-16 | 5 | -0.0287812990730925 |
| 6.2 | 6.42 | 6.10212424977112 | 6.98584655834528E-10 | 4 | 0.0254363379946523 |
| 6.42 | 6.6 | 6.44364636296136 | 1.05765520907325E-10 | 3 | -0.0149406825654843 |
| 6.6 | 10 | 6.57163135207751 | 4.83802204027939E-11 | 3 | 0.0138685576465538 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metodas | Rėžio pradžia | Rėžio pabaiga | Šaknis (x) | Tikslumas f(x) | Iteracijų kiekis | Pradinis žingsnis |
| Skenavimo | 5 | 10 | 5.33699999999999 | 8.48288687540144E-07 | 2 | 0.2 |
| 5.58095999999999 | 9.80325291262711E-07 | 2 | 0.2 |
| 5.91059999999998 | 4.57603485647671E-07 | 2 | 0.2 |
| 6.10209999999996 | 6.16096474429732E-07 | 2 | 0.2 |
| 6.44357999999995 | 9.92041870109383E-07 | 2 | 0.2 |
| 6.57155999999994 | 9.89125204289279E-07 | 2 | 0.2 |

Šaknys pagal wolframalpha.com

|  |  |
| --- | --- |
| Pirma šaknis | 5.337017 |
| Antra šaknis | 5.580984 |
| Trečia šaknis | 5.910616 |
| Ketvirta šaknis | 6.102124 |
| Penkta šaknis | 6.443646 |
| Šešta šaknis | 6.571631 |

### Dalinė išvada

Kaip ir daugianario atvėju, paprastųjų iteracijų metodas nebuvo labai efektyvus. Mažiausiai iteracijų užtruko skenavimo metodas, o didžiausią tikslumą parodė niutono metodas. Sulyginus su wolframalpha.com atsakymai yra geri ir tikslesni.

## Praktinė užduotis

### Uždavinys

Krentančio parašiutininko greitis užrašomas dėsniu

𝑣(𝑡) = 𝑚𝑔 (1 − 𝑒−(𝑚𝑐)𝑡), čia g = 9,8 m/s2 , parašiutininko masė 𝑚. Koks pasipriešinimo

𝑐

koeficientas 𝑐 veikia parašiutininką, jei žinoma, kad po 𝑡1 laisvojo kritimo, jo greitis lygus 𝜐1?

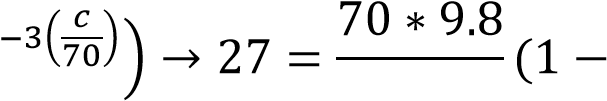
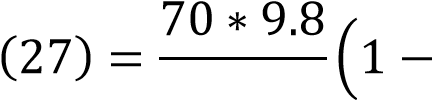
Pagal 9-ąjį variantą buvo gauti tokie skaičiai:

m = 90kg t1 = 3s v1 = 27m/s

### Sprendimas

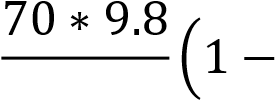
Įsistatome duotas reikšmes į lygtį ir gauname.

𝑐

𝑣 𝑒 𝑒−3(70))

𝑐 𝑐

Persikeliame v(27) į vieną pusę ir prisilyginame lygtį nuliui, kadangi ieškome *c* reikšmės, todėl tai tampa mūsų nežinomuoju *x.*

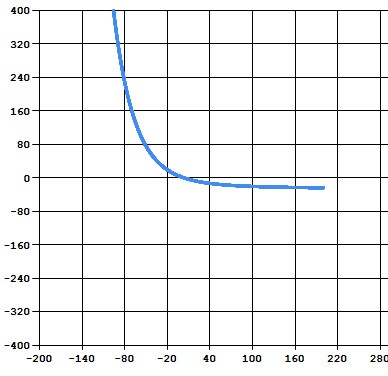
𝑥

−3()

𝑒 70 ) − 27 = 0

𝑥

Šios lygties šaknį, tai yra pasipriešinimo koeficientą *c* ieškosime paprastųjų iteracijų metodu.



*3 pav. Parašiutininko galutinė funkcija*

Galutinė šaknis buvo rasta per **45** iteracijas, jos reikšmė **4.0320801**, su tikslumu **0.0000001.**

### Dalinė išvada

Pasinaudojus paprastųjų iteracijų metodu, buvo rastas apytikslis pasipriešinimo koeficientas parašiutininko uždavinyje.

# Programų kodai

Visi metodai buvo realizuoti naudojant **C#**

## Iteracinis metodas

Iteraciniam metodui buvo naudojama papildoma funkcija.

private double F2(double x)

{

return x+F(x)/a1;

}

Pačio iteracinio metodo realizacija

private void Timer5\_Tick(object sender, EventArgs e)

{

if (iii == 0)

{

xtemp = (x1 + x2) / 2;

}

if (Math.Abs(xtemp - x0) > 1e-7 & iii <= N)

{

X1X2.Points.Clear();

XMid.Points.Clear();

XMid.Points.AddXY(xtemp, 0);

richTextBox1.AppendText(String.Format(" {0,6:d} {1,12:f7} {2,12:f7} \n", iii, xtemp, xtemp - x0));

x0 = xtemp; xtemp = F2(x0); iii = iii + 1;

} else {

richTextBox1.AppendText("Skaičiavimai baigti"); timer5.Stop();

}

}

## Skenavimo metodas

Pačio skenavimo metodo realizacija.

private void Timer6\_Tick(object sender, EventArgs e)

{

if (Math.Abs(F(x1)) > 1e-6 & iii <= N)

{

X1X2.Points.Clear();

XMid.Points.Clear();

X1X2.Points.AddXY(x1, 0);

if (Math.Sign((double)F(x1)) != Math.Sign((double)F(x1 + dx)))

{

richTextBox1.AppendText(String.Format("iteracija: {0}; x1={1}; F(x1)={2}; dx={3}\n",

iii,x1, F(x1), dx)); dx = dx / 10;

iii = iii + 1;

}

x1 = x1 + dx;

} else { iii = 0; dx = 0.5; xList[listCount] = x1; fxList[listCount++] = F(x1); x1 = x1 + dx/2;

richTextBox1.AppendText("Viena saknis rasta!!! :)\n"); }

if(x1 > x2)

{

richTextBox1.AppendText(String.Format("Saknu atskamymai:\n")); for (int i = 0; i < listCount; i++)

{

richTextBox1.AppendText(String.Format("Saknis x={0}, kai F(x)={1}\n", xList[i], fxList[i])); }

timer6.Stop();

}

}

## Niutono metodas

Niutono metodo realizacija.

double tol = 10e-10; while (Math.Abs(F(x1)) > tol)

{

X1X2.Points.Clear();

XMid.Points.Clear();

XMid.Points.AddXY(x1, 0); x1 = x1 - (1 / DF(x1)) \* F(x1);

richTextBox1.AppendText(String.Format("i={0} ;x1={1} Fx={2} DF={3}\n", iii, x1, F(x1), DF(x1))); iii++;

}

richTextBox1.AppendText(String.Format("Gauta saknis!!! - x1={0} Fx={1} DF={2}\n", x1, F(x1), DF(x1)));

## Praktinio uždavinio metodas

Praktinis uždavinys buvo realizuotas Iteraciniu metodu, kodas pateiktas **4.1.** punkte.

# Išvados

Šio laboratorinio metu buvo išmokta rasti įvairiais būdais duotų funkcijų su vienu nežinomuoju šaknis. Buvo panaudoti trys šaknų radimo būdai.

1. Paprastųjų iteracijų
2. Niutono (liestinių)
3. Skenavimo

Kadangi paprastųjų iteracijų metodas reikalauja gana tikslaus *a* koeficiento pasirinkimo kiekvienu atvėju, tai manau dėl nevisiškai kokybiško pasirinkimo, jis tapo blogiausiu.

Vidutiniškai pasirodė (iš visų tryjų) skenavimo metodas, jam taip pat reikia nurodyti gerą pradinį žingsnį bei mažėjimo žingsnį. Geriausiai pasirodė Niutono (liestinių) metodas, jo tikslumas buvo geriausias, mažiausiai naudota įteracijų, kadangi nereikėjo jokios konstantos pasirinkti, išvestinė padarė tai už mus.

Praktiniame uždavinyje buvo įtvirtintas išmoktas paprastųjų iteracijų metodas.