

Skaitiniai metodai ir algoritmai (P170B115)

Projektas

Pirma užduotis

Atliko:

IFF-8/12 gr. studentas

Jokūbas Akramas

2020 m. rugsėjo 28 d.

Priėmė:

lekt. Darius Naujokaitis

Turinys

1. Uždutis.....	3
1.1. Išspręskite netiesines lygtis	3
1.2. Pagal pateiktą uždavinio sąlygą sudarykite netiesinę lygtį ir pasirinktu skaitiniu metodu iš lentelės ją išspręskite.	3
2. Rezultatai	5
2.1. Pirmos dalies rezultatai	5
2.1.1 Daugianario rezultatai.....	5
2.1.2 Funkcijos rezultatai.....	6
2.2. Antros dalies rezultatai	7
3. Programų kodai.....	9
4. Išvados	23

1. Užduotis

Netiesinių lygčių sprendimas (7 variantas)

1.1. Išspręskite netiesines lygtis

a) daugianaris $f(x) = 0$;

b) transcendentinė funkcija $g(x) = 0$.

1. (tik lygčiai su daugianariu $f(x)$) Nustatykite daugianario $f(x)$ šaknų intervalą, taikydami „grubų“ ir tikslesnį įverčius. Grafiškai pavaizduokite apskaičiuotų šaknų intervalo galus.
2. Daugianarį $f(x)$ grafiškai pavaizduokite nustatytame šaknų intervale. Grafiko ašis pakeiskite taip, kad būtų aiškiai matomos daugianario šaknys. Funkciją $g(x)$ grafiškai pavaizduokite užduotyje nurodytame intervale.
3. Naudodami skenavimo algoritmą su nekintančiu skenavimo žingsniu atskirkite šaknų intervalus. Daugianariui skenavimo intervalas parenkamas pagal įverčių reikšmes, funkcija skenuojama užduotyje nurodytame intervale. Šaknies atskyrimo intervalai naudojami kaip pradiniai intervalai (artiniai) šaknų tikslinimui.
4. Skenavimo metodu atskirtas daugianario ir funkcijos šaknis tikslinkite užduotyje nurodytais metodais. Užrašykite skaičiavimų pabaigos sąlygas. Skaičiavimų rezultatus pateikite lentelėje, kurioje nurodykite šaknies tikslinimui naudojamą metodą, pradinį artinį ar intervalą, gautą sprendinį (šaknį), tikslumą, iteracijų skaičių. Palyginkite, kuris metodas randa sprendinį su mažesniu iteracijų skaičiumi.
5. Gautas šaknų reikšmes patikrinkite naudodami išorinius išteklius (pvz., MATLAB funkcijas roots arba fzero, tinklapį wolframalpha.com ir t.t.).

1.2. Pagal pateiktą uždavinio sąlygą sudarykite netiesinę lygtį ir pasirinktu skaitiniu metodu iš lentelės ją išspręskite.

Ataskaitoje pateikite pradinius metodo parametrus (metodo žingsnį, pradinį artinį, izoliacijos intervalą ir pan.), iteracijų pabaigos sąlygą, tikslumą, gautą lygties sprendinį ir sudarytos funkcijos reikšmę, argumentus, kodėl pasirinkote šį metodą. Pateikite grafinį lygties sprendimą.

1 lentelė. Netiesinių lygčių sprendimas. Metodai.

Metodo Nr.	Metodo pavadinimas
1	Stygų
2	Paprastųjų iteracijų
3	Niutono (liestinių)
4	Kvazi-Niutono (kirstinių)
5	Skenavimo su mažėjančiu žingsniu

Uždavinys variantams 6-10

Krentančio parašiutininko greitis užrašomas dėsniu $v(t) = \frac{mg}{c} \left(1 - e^{-\left(\frac{c}{m}\right)t} \right)$, čia $g = 9,8 \text{ m/s}^2$, parašiutininko masė m . Koks pasipriešinimo koeficientas c veikia parašiutininką, jei žinoma, kad po t_1 laisvojo kritimo, jo greitis lygus v_1 ?

Varianto Nr.	$m, \text{ kg}$	$t_1, \text{ s}$	$v_1, \text{ m/s}$
6	90	3,5	30
7	80	4	36
8	60	3	25
9	70	3	27
10	60	4	30

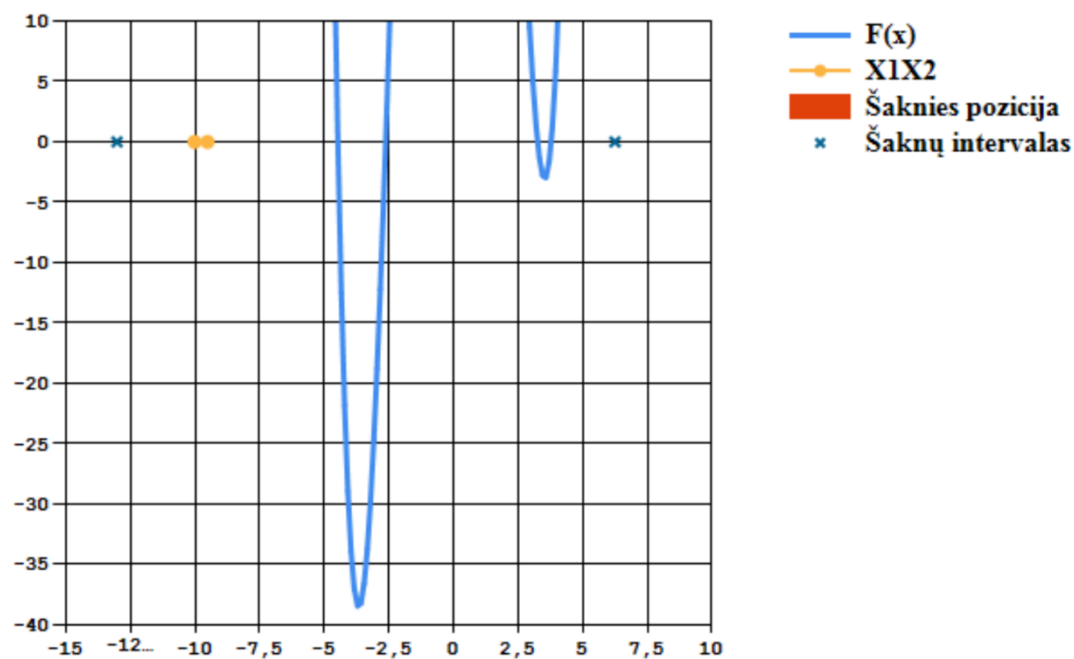
7 varianto pirmos dalies lygtys:

7	$0.89x^4 + 0.07x^3 - 23.05x^2 + 4.03x + 128.68$	$e^{-x} \frac{\cos(x)}{x-6}; -5 \leq x \leq 5$	1, 2, 5
---	---	--	---------

2. Rezultatai

2.1. Pirmos dalies rezultatai

2.1.1 Daugianario rezultatai

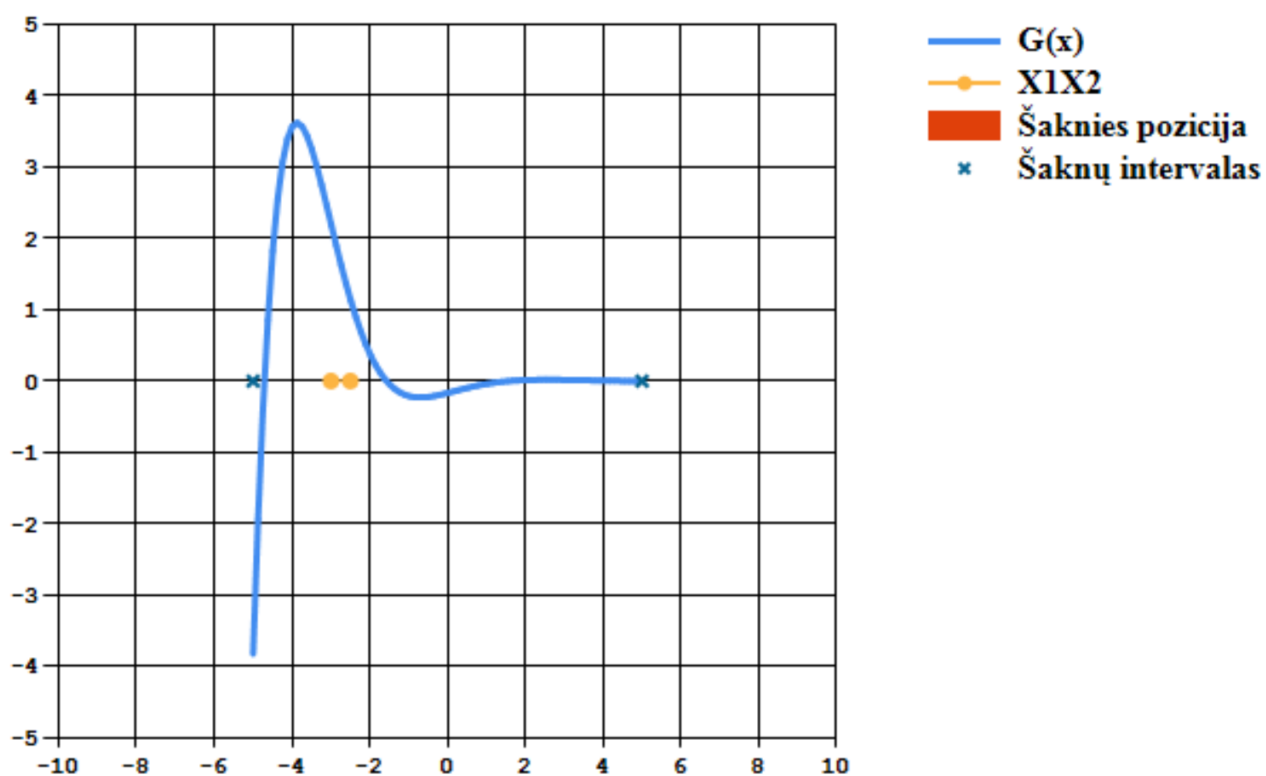


pav. 1 Daugianario grafikas

lentelė 1 Daugianario rezultatai

Metodas Šaknies nr.	Stygų	Paprastųjų iteracijų	Skenavimo su mažėjančiu žingsniu	Tikrinimas išoriniais šaltiniais (wolframalpha.com)
1	$x = -4.48$	$x = -4.48$	$x = -4.5$	$x = -4.48$
2	$x = -2.63$	$x = -2.63$	$x = -2.9$	$x = -2.63$
3	$x = 3.25$	$x = 3.25$	$x = 3.1$	$x = 3.25$
4	$x = 3.78$	$x = 3.78$	$x = 3.6$	$x = 3.78$

2.1.2 Funkcijos rezultatai



pav. 2 Funkcijos grafikas

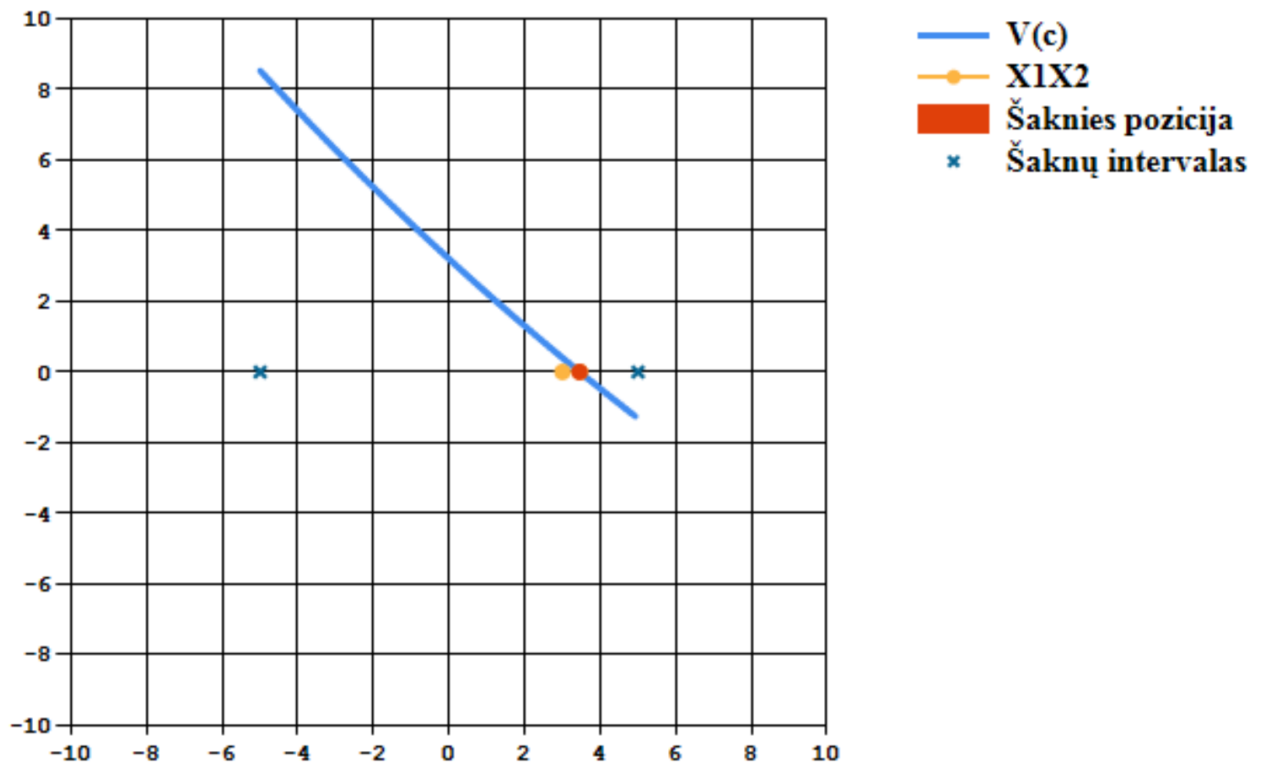
lentelė 2 Funkcijos rezultatai

Metodas Šaknies nr.	Stygų	Paprastųjų iteracijų	Skenavimo su mažėjančiu žingsniu	Tikrinimas išoriniais šaltiniais (wolframalpha.com)
1	$x = -4.71$	$x = -4.71$	$x = -4.88$	
2	$x = -1.57$	$x = -1.57$	$x = -1.88$	
3	$x = 1.57$	$x = 1.57$	$x = 1.53$	
4	$x = 4.71$	$x = 4.71$	$x = 4.63$	

2.2. Antros dalies rezultatai

Sudaryta lygtis:

$$V(c) = \frac{mg}{c} \left(1 - e^{-\left(\frac{c}{m}\right)t} \right) - v = 0$$



pav. 3 Antros dalies lygties grafikas

Lygties sprendimui pasirinkau stygų metodą, nes iš nagrinėtų metodų šis man pasirodė labiausiai automatizuotas (paprastųjų iteracijų metodui tiksliam šaknies paskaičiavimui iš pradžių reikėjo apsiskaičiuoti lygties išvestinę ir pagal vizualų įvertinimą įstatyti šaknies reikšmę ir gauti įvertį, iš kurio prisilyginus įvertio reikšmių intervalą $[-2; 0]$, gauti alfa koeficientą). Taip pat stygų metodas, mano nuomone, pranašesnis ir už skėnavimą su mažėjančiu žingsniu, nes reikšmes gavo su mažesniu iteracijų kiekiu.

Programos vykdymo išrašas:

```
Skenuojamas intervalas (-5, 5) su žingsniu: 1 atskirti šaknų intervalus
1-asis intervalas: (3; 4)

1-tos šaknies paieška
Iteracija      x          V(x)          x1          x2          V(x1)          V(x2)
0              3,4601440    -0,0035623    3,0000000    4,0000000    0,4016488    -0,4712276
1              3,4560990    -0,0000266    3,0000000    3,4601440    0,4016488    -0,0035623

1-ta šaknis rasta. Metodas: Stygų, pradinis šaknies intervalas: (3; 4), gauta šaknis:
x=3,456069, tikslumas: 1E-05, iteracijų skaičius: 2

Skaičiavimai baigti.
```


3. Programų kodai

Programinis kodas patalpintas viešojoje GitHub saugykloje adresu: <https://github.com/juokavom/P170B115-Lab1>

Taip pat pateikiu metodų realizacijos kodą:

```
//Jokubas Akramas IFF-8/12 7 var.  
//P170B115 Skaityniai metodai ir algoritmai (6 kr.)  
//I projektinė užduotis  
  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Windows.Forms.DataVisualization.Charting;  
using MathNet.Numerics.LinearAlgebra;  
using MathNet.Numerics.LinearAlgebra.Factorization;  
using MathNet.Numerics.Optimization;  
using MathNet.Numerics.Statistics;  
  
namespace Pvz1  
{  
    public partial class Form1 : Form  
    {  
        List<Timer> Timerlist = new List<Timer>();  
  
        private class Interval  
        {  
            public float x1;  
            public float x2;  
            public bool done;  
        }  
  
        public Form1()  
        {  
            InitializeComponent();  
            Initialize();  
        }  
  
        float x1, x2, xtemp, zingsnis, x_nueita, x1prad, x2prad;  
        int N = 1000;  
        int iii, i_interval, current;  
  
        Series Fx, X1X2, XMid, Iv;  
        float[] a = { 128.68f, 4.03f, -23.05f, 0.07f, 0.89f };  
        float[] a_neg = { 128.68f, -4.03f, -23.05f, -0.07f, 0.89f };  
        List<Interval> intervalsArray;  
        Interval[] intervals;  
  
        /// <summary>  
        /// Sprendžiama  $F(x) = 0.89x^4 + 0.07x^3 - 23.05x^2 + 4.03x + 128.68$   
        /// </summary>  
        /// <param name="x">daugianario argumentas</param>  
        /// <returns></returns>  
        private double F(double x)  
        {  
            return (double)(a[4] * Math.Pow(x, 4) + a[3] * Math.Pow(x, 3) + a[2] * Math.Pow(x,  
2) + a[1] * x + a[0]);  
        }  
  
        /// <summary>
```

```

/// Sprendžiama  $G(x) = e^{-x} (\cos(x)/(x-6))$ ;  $-5 \leq x \leq 5$ 
/// </summary>
/// <param name="x">funkcijos argumentas</param>
/// <returns></returns>
private double G(double x)
{
    return (double)(Math.Pow(Math.E, -x) * (Math.Cos(x) / (x - 6)));
}
private float PreciseInterval(float[] array)
{
    int k = array.Length - 1;
    float B = Math.Abs(array[0]);
    for (int i = 1; i < array.Length; i++)
    {
        if (Math.Abs(array[i]) > B && array[i] < 0)
        {
            B = Math.Abs(array[i]);
        }
    }
    for (int i = array.Length - 1; i >= 0; i--)
    {
        if (array[i] < 0)
        {
            k = i;
            i = -1;
        }
    }
    k = array.Length - k;
    return (float)(1 + Math.Pow((B / a[a.Length - 1]), 1.0 / k));
}

private void BigInterval(ref float x1, ref float x2)
{
    float R, R_positive, R_negative;
    R = 1 + a.MaximumAbsolute() / a[a.Length - 1];
    R_positive = PreciseInterval(a);
    R_negative = PreciseInterval(a_neg);
    x2 = Math.Min(R, R_positive);
    x1 = -Math.Min(R, R_negative);
}

private void button2_Click(object sender, EventArgs e)
{
    ClearForm1();
    PreparareForm(-15, 10, -40, 10);
    BigInterval(ref x1, ref x2);
    iii = 0;
    richTextBox1.AppendText("Sprendžiamas dauginaris  $F(x) = 0.89x^4 + 0.07x^3 -$   

23.05x2 + 4.03x + 128.68\n\n");
    richTextBox1.AppendText(string.Format("Nustatytas dauginario šaknų intervalas  

grubių ir tikslesnių įverčių: ({0}, {1})\n\n", x1, x2));
    Fx = chart1.Series.Add("F(x)");
    Fx.ChartType = SeriesChartType.Line;
    double x = -10;
    for (int i = 0; i < 250; i++)
    {
        Fx.Points.AddXY(x, F(x));
        x = x + (2 * Math.PI) / 50;
    }
    Fx.BorderWidth = 3;

    X1X2 = chart1.Series.Add("X1X2");
    X1X2.MarkerStyle = MarkerStyle.Circle;
    X1X2.MarkerSize = 8;
    X1X2.ChartType = SeriesChartType.Point;
    X1X2.ChartType = SeriesChartType.Line;

```

```

XMid = chart1.Series.Add("Šaknies pozicija");
XMid.MarkerStyle = MarkerStyle.Circle;
X1X2.ChartType = SeriesChartType.Point;
X1X2.ChartType = SeriesChartType.Line;
XMid.MarkerSize = 8;

Iv = chart1.Series.Add("Šaknų intervalas");
Iv.MarkerStyle = MarkerStyle.Cross;
Iv.MarkerSize = 8;
Iv.ChartType = SeriesChartType.Point;
Iv.Points.AddXY(x1, 0);
Iv.Points.AddXY(x2, 0);

intervalsArray = new List<Interval>();
x_nueita = x1;
zingsnis = 0.5f;
i_interval = 0;
richTextBox1.AppendText(string.Format("Skenuojamas intervalas ({0}, {1}) su
žingsniu: {2} atskirti šaknų intervalus\n", x1, x2, zingsnis));
timer4.Enabled = true;
timer4.Interval = 500;
timer4.Start();

}

/// <summary>
/// timer2 iteracijoje atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void timer4_Tick(object sender, EventArgs e)
{
    xtemp = x_nueita + zingsnis;

    X1X2.Points.Clear();
    XMid.Points.Clear();

    X1X2.Points.AddXY(x_nueita, 0);
    X1X2.Points.AddXY(xtemp, 0);

    if (Math.Sign((double)F(x_nueita)) != Math.Sign((double)F(xtemp)))
    {
        richTextBox1.AppendText(string.Format("{0}-asis intervalas: ({1}; {2}) \n",
++i_interval, x_nueita, xtemp));
        intervalsArray.Add(new Interval() { x1 = x_nueita, x2 = xtemp, done = false
});
    }

    x_nueita = xtemp;

    if (xtemp >= x2)
    {
        timer4.Stop();
        current = 0;
        richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
        if (stygu.Checked)
        {
            richTextBox1.AppendText("Iteracija          x          F(x)          x1
x2          F(x1)          F(x2)          \n");
        }
        else if (iteraciju.Checked)
        {
            richTextBox1.AppendText("Iteracija          x1          F(x1)\n");
        }
        else if (skenavimo.Checked)
        {

```

```

        richTextBox1.AppendText("Iteracija          x1          x2          F(x1)
F(x2)          \n");
    }
    intervals = intervalsArray.ToArray();
    x1prad = intervals[current].x1;
    x2prad = intervals[current].x2;
    timer2.Enabled = true;
    timer2.Interval = 500;
    timer2.Start();
    }
}
private void StyguMetodas()
{
    x1 = intervals[current].x1;
    x2 = intervals[current].x2;

    float k = Math.Abs((float)(F(x1) / F(x2)));
    xtemp = (x1 + k * x2) / (1 + k);

    double tikslumas = 1e-5;

    if (Math.Abs(F(xtemp)) > tikslumas & iii <= N)
    {
        X1X2.Points.Clear();
        XMid.Points.Clear();

        X1X2.Points.AddXY(x1, 0);
        X1X2.Points.AddXY(x2, 0);
        XMid.Points.AddXY(xtemp, 0);

        richTextBox1.AppendText(String.Format(" {0,6:d}   {1,12:f7}   {2,12:f7}
{3,12:f7} {4,12:f7} {5,12:f7} {6,12:f7}\n",
        iii, xtemp, F(xtemp), x1, x2, F(x1), F(x2)));
        if (Math.Sign((double)F(x1)) != Math.Sign((double)F(xtemp)))
        {
            intervals[current].x2 = xtemp;
        }
        else
        {
            intervals[current].x1 = xtemp;
        }
        iii = iii + 1;
    }
    else if (current < intervals.Length - 1)
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
        " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
        1 + current, "Stygu", x1prad, x2prad, xtemp, tikslumas, iii));
        current++;
        iii = 0;
        x1prad = intervals[current].x1;
        x2prad = intervals[current].x2;
        richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
        richTextBox1.AppendText("Iteracija          x          F(x)          x1
x2          F(x1)          F(x2)          \n");
    }
    else
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
        " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
        1 + current, "Stygu", x1prad, x2prad, xtemp, tikslumas, iii));

        richTextBox1.AppendText("\nSkaičiavimai baigti.");
    }
}

```

```

        timer2.Stop();
    }
}
private void IteracijuMetodas()
{
    x1 = intervals[current].x1;
    x2 = intervals[current].x2;

    float[] alfa = { 70f, -48f, 22f, -24f };
    //float alfa = 70f; //-4.7 (1)
    //float alfa = -48f; //2.62 (2)
    //float alfa = 22f; //3.25 (3)
    //float alfa = -24f; //3.77 (4)

    xtemp = (float)(F(x1) / alfa[current]) + x1;

    double tikslumas = 1e-4;

    if (Math.Abs(F(x1)) > tikslumas & iii <= N)
    {
        X1X2.Points.Clear();
        XMid.Points.Clear();

        XMid.Points.AddXY(x1, 0);

        richTextBox1.AppendText(String.Format(" {0,6:d} {1,12:f7} {2,12:f7}\n",
            iii, x1, F(x1)));
        iii = iii + 1;
        intervals[current].x1 = xtemp;
    }
    else if (current < intervals.Length - 1)
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
            " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}, alfa
koeficientas: {7}\n",
            1 + current, "Iteraciju", x1prad, x2prad, intervals[current].x1, tikslumas,
            iii, alfa[current]));
        current++;
        iii = 0;
        x1prad = intervals[current].x1;
        x2prad = intervals[current].x2;
        richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
        richTextBox1.AppendText("Iteracija          x1          F(x1)\n");
    }
    else
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
            " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}, alfa
koeficientas: {7}\n",
            1 + current, "Iteraciju", x1prad, x2prad, intervals[current].x1, tikslumas,
            iii, alfa[current]));

        richTextBox1.AppendText("\nSkaičiavimai baigti.");
        timer2.Stop();
    }
}
private void SkenavimoMetodas()
{
    x1 = intervals[current].x1;
    x2 = intervals[current].x2;
    double tikslumas = 1e-6;
    xtemp = x1 + zingsnis;

    if (Math.Abs(x2 - x1) > tikslumas & iii <= N)

```

```

        {
            X1X2.Points.Clear();
            XMid.Points.Clear();
            X1X2.Points.AddXY(x1, 0);
            X1X2.Points.AddXY(xtemp, 0);
            richTextBox1.AppendText(String.Format(" {0,6:d}    {1,12:f7} {2,12:f7}
{3,12:f7} {4,12:f7}\n",
                iii, x1, xtemp, F(x1), F(xtemp)));

            if (Math.Sign((double)F(x1)) != Math.Sign((double)F(xtemp)))
            {
                intervals[current].x1 = x1;
                intervals[current].x2 = xtemp;
                zingsnis = (float)(xtemp - x1) / 5;
            }
            else
            {
                intervals[current].x1 = xtemp;
                intervals[current].x2 = xtemp + zingsnis;
            }
            iii++;
        }
        else if (current < intervals.Length - 1)
        {
            richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
                " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
                1 + current, "Skenavimo su mažėjančiu žingsniu", x1prad, x2prad, x1,
tikslumas, iii));
            current++;
            iii = 0;
            x1prad = intervals[current].x1;
            x2prad = intervals[current].x2;
            richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
            richTextBox1.AppendText("Iteracija          x1          x2          F(x1)
F(x2)          \n");
        }
        else
        {
            richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
                " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
                1 + current, "Skenavimo su mažėjančiu žingsniu", x1prad, x2prad, x1,
tikslumas, iii));

            richTextBox1.AppendText("\nSkaičiavimai baigti.");
            timer2.Stop();
        }
    }
}
/// <summary>
/// timer2 iteracijoje atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void timer2_Tick(object sender, EventArgs e)
{
    if (stygu.Checked)
    {
        StyguMetodas();
    }
    else if (iteraciju.Checked)
    {
        IteracijuMetodas();
    }
    else if (skenavimo.Checked)
    {

```

```

        x1 = intervals[current].x1;
        x2 = intervals[current].x2;
        zingsnis = (float)(x2 - x1) / 5;
        SkenavimoMetodas();
    }
}
private void button3_Click(object sender, EventArgs e)
{
    ClearForm1();
    PreparareForm(-10, 10, -5, 5);
    x1 = -5;
    x2 = 5;
    iii = 0;
    richTextBox1.AppendText("Sprendžiama funkcija  $G(x) = e^{-x} (\cos(x)/(x-6))$ \n\n");
    Fx = chart1.Series.Add("G(x)");
    Fx.ChartType = SeriesChartType.Line;
    double x = -5;
    for (int i = 0; i < 250; i++)
    {
        if (x > 5) break;
        Fx.Points.AddXY(x, G(x));
        x = x + (2 * Math.PI) / 50;
    }
    Fx.BorderWidth = 3;

    X1X2 = chart1.Series.Add("X1X2");
    X1X2.MarkerStyle = MarkerStyle.Circle;
    X1X2.MarkerSize = 8;
    X1X2.ChartType = SeriesChartType.Point;
    X1X2.ChartType = SeriesChartType.Line;

    XMid = chart1.Series.Add("Šaknies pozicija");
    XMid.MarkerStyle = MarkerStyle.Circle;
    X1X2.ChartType = SeriesChartType.Point;
    X1X2.ChartType = SeriesChartType.Line;
    XMid.MarkerSize = 8;

    Iv = chart1.Series.Add("Šaknų intervalas");
    Iv.MarkerStyle = MarkerStyle.Cross;
    Iv.MarkerSize = 8;
    Iv.ChartType = SeriesChartType.Point;
    Iv.Points.AddXY(x1, 0);
    Iv.Points.AddXY(x2, 0);

    intervalsArray = new List<Interval>();
    x_nueita = x1;
    zingsnis = 0.5f;
    i_interval = 0;
    richTextBox1.AppendText(string.Format("Skenuojamas intervalas ({0}, {1}) su žingsniu: {2} atskirti šaknų intervalus\n", x1, x2, zingsnis));
    timer3.Enabled = true;
    timer3.Interval = 500;
    timer3.Start();
}
/// <summary>
/// timer2 iteracijoje atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void timer3_Tick(object sender, EventArgs e)
{
    xtemp = x_nueita + zingsnis;

    X1X2.Points.Clear();
    XMid.Points.Clear();

    X1X2.Points.AddXY(x_nueita, 0);
    X1X2.Points.AddXY(xtemp, 0);
}

```

```

        if (Math.Sign((double)G(x_nueita)) != Math.Sign((double)G(xtemp)))
        {
            richTextBox1.AppendText(string.Format("{0}-asis intervalas: ({1}; {2}) \n",
++i_interval, x_nueita, xtemp));
            intervalsArray.Add(new Interval() { x1 = x_nueita, x2 = xtemp, done = false
        });
    }

    x_nueita = xtemp;

    if (xtemp >= x2)
    {
        timer3.Stop();
        current = 0;
        richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
        if (stygu.Checked)
        {
            richTextBox1.AppendText("Iteracija          x          G(x)          x1
x2          G(x1)          G(x2)          \n");
        }
        else if (iteraciju.Checked)
        {
            richTextBox1.AppendText("Iteracija          x1          G(x1)\n");
        }
        else if (skenavimo.Checked)
        {
            richTextBox1.AppendText("Iteracija          x1          x2          G(x1)
G(x2)          \n");
        }
        intervals = intervalsArray.ToArray();
        x1prad = intervals[current].x1;
        x2prad = intervals[current].x2;
        timer5.Enabled = true;
        timer5.Interval = 500;
        timer5.Start();
    }
}

/// <summary>
/// timer2 iteracijoje atliekami veiksmi
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void timer5_Tick(object sender, EventArgs e)
{
    if (stygu.Checked)
    {
        StyguMetodasF();
    }
    else if (iteraciju.Checked)
    {
        IteracijuMetodasF();
    }
    else if (skenavimo.Checked)
    {
        x1 = intervals[current].x1;
        x2 = intervals[current].x2;
        zingsnis = (float)(x2 - x1) / 5;
        SkenavimoMetodasF();
    }
}

private void StyguMetodasF()
{
    x1 = intervals[current].x1;
    x2 = intervals[current].x2;

```



```

float k = Math.Abs((float)(G(x1) / G(x2)));
xtemp = (x1 + k * x2) / (1 + k);

double tikslumas = 1e-5;

if (Math.Abs(G(xtemp)) > tikslumas & iii <= N)
{
    X1X2.Points.Clear();
    XMid.Points.Clear();

    X1X2.Points.AddXY(x1, 0);
    X1X2.Points.AddXY(x2, 0);
    XMid.Points.AddXY(xtemp, 0);

    richTextBox1.AppendText(String.Format(" {0,6:d}   {1,12:f7}   {2,12:f7}
{3,12:f7} {4,12:f7} {5,12:f7} {6,12:f7}\n",
        iii, xtemp, G(xtemp), x1, x2, G(x1), G(x2)));
    if (Math.Sign((double)G(x1)) != Math.Sign((double)G(xtemp)))
    {
        intervals[current].x2 = xtemp;
    }
    else
    {
        intervals[current].x1 = xtemp;
    }
    iii = iii + 1;
}
else if (current < intervals.Length - 1)
{
    richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
        " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
        1 + current, "Stygu", x1prad, x2prad, xtemp, tikslumas, iii));
    current++;
    iii = 0;
    x1prad = intervals[current].x1;
    x2prad = intervals[current].x2;
    richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
    richTextBox1.AppendText("Iteracija      x      G(x)      x1
x2      G(x1)      G(x2)      \n");
}
else
{
    richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
        " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
        1 + current, "Stygu", x1prad, x2prad, xtemp, tikslumas, iii));

    richTextBox1.AppendText("\nSkaičiavimai baigti.");
    timer5.Stop();
}
}
private void IteracijuMetodasF()
{
    x1 = intervals[current].x1;
    x2 = intervals[current].x2;

    float[] alfa = { -7.5f, 0.5f, -0.2f, 0.01f };

    //float alfa = 7.5f; //-4.7 (1)
    //float alfa = 0.5f; //-1.57 (2)
    //float alfa = -0.2f; //1.57 (3)
    //float alfa = 0.01f; //4.7 (4)

    xtemp = (float)(G(x1) / alfa[current]) + x1;

```

```

double tikslumas = 1e-4;

if (Math.Abs(G(x1)) > tikslumas & iii <= N)
{
    X1X2.Points.Clear();
    XMid.Points.Clear();

    XMid.Points.AddXY(x1, 0);

    richTextBox1.AppendText(String.Format(" {0,6:d} {1,12:f7} {2,12:f7}\n",
        iii, x1, G(x1)));
    iii = iii + 1;
    intervals[current].x1 = xtemp;
}
else if (current < intervals.Length - 1)
{
    richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodos: {1},
pradinis šaknies intervalas: ({2}; {3})," +
        " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}, alfa
koeficientas: {7}\n",
        1 + current, "Iteraciju", x1prad, x2prad, intervals[current].x1,
tikslumas, iii, alfa[current]));
    current++;
    iii = 0;
    x1prad = intervals[current].x1;
    x2prad = intervals[current].x2;
    richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
    richTextBox1.AppendText("Iteracija          x1          G(x1)\n");
}
else
{
    richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodos: {1},
pradinis šaknies intervalas: ({2}; {3})," +
        " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}, alfa
koeficientas: {7}\n",
        1 + current, "Iteraciju", x1prad, x2prad, intervals[current].x1, tikslumas,
iii, alfa[current]));

    richTextBox1.AppendText("\nSkaičiavimai baigti.");
    timer5.Stop();
}
}
private void SkenavimoMetodasF()
{
    x1 = intervals[current].x1;
    x2 = intervals[current].x2;
    double tikslumas = 1e-6;
    xtemp = x1 + zingsnis;

    if (Math.Abs(x2 - x1) > tikslumas & iii <= N)
    {
        X1X2.Points.Clear();
        XMid.Points.Clear();
        X1X2.Points.AddXY(x1, 0);
        X1X2.Points.AddXY(xtemp, 0);
        richTextBox1.AppendText(String.Format(" {0,6:d} {1,12:f7} {2,12:f7}
{3,12:f7} {4,12:f7}\n",
            iii, x1, xtemp, G(x1), G(xtemp)));

        if (Math.Sign((double)G(x1)) != Math.Sign((double)G(xtemp)))
        {
            intervals[current].x1 = x1;
            intervals[current].x2 = xtemp;
            zingsnis = (float)(xtemp - x1) / 5;
        }
    }
}

```

```

        else
        {
            intervals[current].x1 = xtemp;
            intervals[current].x2 = xtemp + zingsnis;
        }
        iii++;
    }
    else if (current < intervals.Length - 1)
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
            " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
            1 + current, "Skenavimo su mažėjančiu žingsniu", x1prad, x2prad, x1,
tikslumas, iii));
        current++;
        iii = 0;
        x1prad = intervals[current].x1;
        x2prad = intervals[current].x2;
        richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
        richTextBox1.AppendText("Iteracija          x1          x2          G(x1)
G(x2)          \n");
    }
    else
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
            " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
            1 + current, "Skenavimo su mažėjančiu žingsniu", x1prad, x2prad, x1,
tikslumas, iii));

        richTextBox1.AppendText("\nSkaičiavimai baigti.");
        timer5.Stop();
    }
}
///
/// Antra uzduotis
/// <summary>
///
/// Antra uzduotis
/// <summary>

private double V(double c)
{
    double m = 80;
    double t = 4;
    double v = 36;
    double g = 9.8;
    if (c == 0) c = 0.00000001;
    return ((m * g) / c) * (1 - Math.Pow(Math.E, -(c / m) * t))-v;
}

private void button5_Click(object sender, EventArgs e)
{
    ClearForm1();
    PreparareForm(-10, 10, -10, 10);
    x1 = -5;
    x2 = 5;
    iii = 0;
    Fx = chart1.Series.Add("V(c)");
    Fx.ChartType = SeriesChartType.Line;
    double x = -5;
    for (int i = 0; i < 250; i++)
    {
        if (x > 5) break;
        Fx.Points.AddXY(x, V(x));
        x = x + (2 * Math.PI) / 50;
    }
}

```

```

    }
    Fx.BorderWidth = 3;

    X1X2 = chart1.Series.Add("X1X2");
    X1X2.MarkerStyle = MarkerStyle.Circle;
    X1X2.MarkerSize = 8;
    X1X2.ChartType = SeriesChartType.Point;
    X1X2.ChartType = SeriesChartType.Line;

    XMid = chart1.Series.Add("Šaknies pozicija");
    XMid.MarkerStyle = MarkerStyle.Circle;
    X1X2.ChartType = SeriesChartType.Point;
    X1X2.ChartType = SeriesChartType.Line;
    XMid.MarkerSize = 8;

    Iv = chart1.Series.Add("Šaknų intervalas");
    Iv.MarkerStyle = MarkerStyle.Cross;
    Iv.MarkerSize = 8;
    Iv.ChartType = SeriesChartType.Point;
    Iv.Points.AddXY(x1, 0);
    Iv.Points.AddXY(x2, 0);

    intervalsArray = new List<Interval>();
    x_nueita = x1;
    zingsnis = 1f;
    i_interval = 0;
    richTextBox1.AppendText(string.Format("Skenuojamas intervalas ({0}, {1}) su
žingsniu: {2} atskirti šaknų intervalus\n", x1, x2, zingsnis));
    timer6.Enabled = true;
    timer6.Interval = 500;
    timer6.Start();
}
private void timer6_Tick(object sender, EventArgs e)
{
    xtemp = x_nueita + zingsnis;

    X1X2.Points.Clear();
    XMid.Points.Clear();

    X1X2.Points.AddXY(x_nueita, 0);
    X1X2.Points.AddXY(xtemp, 0);

    if (Math.Sign((double)V(x_nueita)) != Math.Sign((double)V(xtemp)))
    {
        richTextBox1.AppendText(string.Format("{0}-asis intervalas: ({1}; {2}) \n",
++i_interval, x_nueita, xtemp));
        intervalsArray.Add(new Interval() { x1 = x_nueita, x2 = xtemp, done = false
});
    }

    x_nueita = xtemp;

    if (xtemp >= x2)
    {
        timer6.Stop();
        current = 0;
        richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
        richTextBox1.AppendText("Iteracija          x          G(x)          x1
x2          G(x1)          G(x2)          \n");
        intervals = intervalsArray.ToArray();
        x1prad = intervals[current].x1;
        x2prad = intervals[current].x2;
        timer7.Enabled = true;
        timer7.Interval = 500;
        timer7.Start();
    }
}

```

```

private void timer7_Tick(object sender, EventArgs e)
{
    x1 = intervals[current].x1;
    x2 = intervals[current].x2;

    float k = Math.Abs((float)(V(x1) / V(x2)));
    xtemp = (x1 + k * x2) / (1 + k);

    double tikslumas = 1e-5;

    if (Math.Abs(V(xtemp)) > tikslumas & iii <= N)
    {
        X1X2.Points.Clear();
        XMid.Points.Clear();

        X1X2.Points.AddXY(x1, 0);
        X1X2.Points.AddXY(x2, 0);
        XMid.Points.AddXY(xtemp, 0);

        richTextBox1.AppendText(String.Format(" {0,6:d}   {1,12:f7}   {2,12:f7}
{3,12:f7} {4,12:f7} {5,12:f7} {6,12:f7}\n",
            iii, xtemp, V(xtemp), x1, x2, V(x1), V(x2)));
        if (Math.Sign((double)V(x1)) != Math.Sign((double)V(xtemp)))
        {
            intervals[current].x2 = xtemp;
        }
        else
        {
            intervals[current].x1 = xtemp;
        }
        iii = iii + 1;
    }
    else if (current < intervals.Length - 1)
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
            " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
            1 + current, "Stygų", x1prad, x2prad, xtemp, tikslumas, iii));
        current++;
        iii = 0;
        x1prad = intervals[current].x1;
        x2prad = intervals[current].x2;
        richTextBox1.AppendText(string.Format("\n{0}-tos šaknies paieška\n", 1 +
current));
        richTextBox1.AppendText("Iteracija           x           G(x)           x1
x2           G(x1)           G(x2)           \n");
    }
    else
    {
        richTextBox1.AppendText(string.Format("\n{0}-ta šaknis rasta. Metodas: {1},
pradinis šaknies intervalas: ({2}; {3})," +
            " gauta šaknis: x={4, 2}, tikslumas: {5}, iteracijų skaičius: {6}\n",
            1 + current, "Stygų", x1prad, x2prad, xtemp, tikslumas, iii));

        richTextBox1.AppendText("\nSkaičiavimai baigti.");
        timer7.Stop();
    }
}

// ----- KITI METODAI -----

/// <summary>
/// Uždaroma programa
/// </summary>
private void button1_Click(object sender, EventArgs e)

```

```

    {
        Close();
    }

    /// <summary>
    /// Išvalomas grafikas ir consolė
    /// </summary>
    private void button4_Click(object sender, EventArgs e)
    {
        ClearForm1();
    }

    public void ClearForm1()
    {
        richTextBox1.Clear(); // išvalomas richTextBox1
        // sustabdomi timeriai jei tokiu yra
        foreach (var timer in Timerlist)
        {
            timer.Stop();
        }

        // išvalomos visos nubreztos kreives
        chart1.Series.Clear();
    }
}

```

4. Išvados

Darant šį darbą buvo įsisavinti netiesinių lygčių sprendimo būdai (šaknų intervalo nustatymas grubiu ir tikslesniais įverčiais; skenavimas nekintančiu žingsniu šaknų atskyrimui; stygų, paprastųjų iteracijų, skenavimo su mažėjančiu žingsniu metodai. Buvo išaiškinti tam tikrų metodų privalumai bei trūkumai, pagal tai sudarytas algoritmas antros užduoties išsprendimui.