

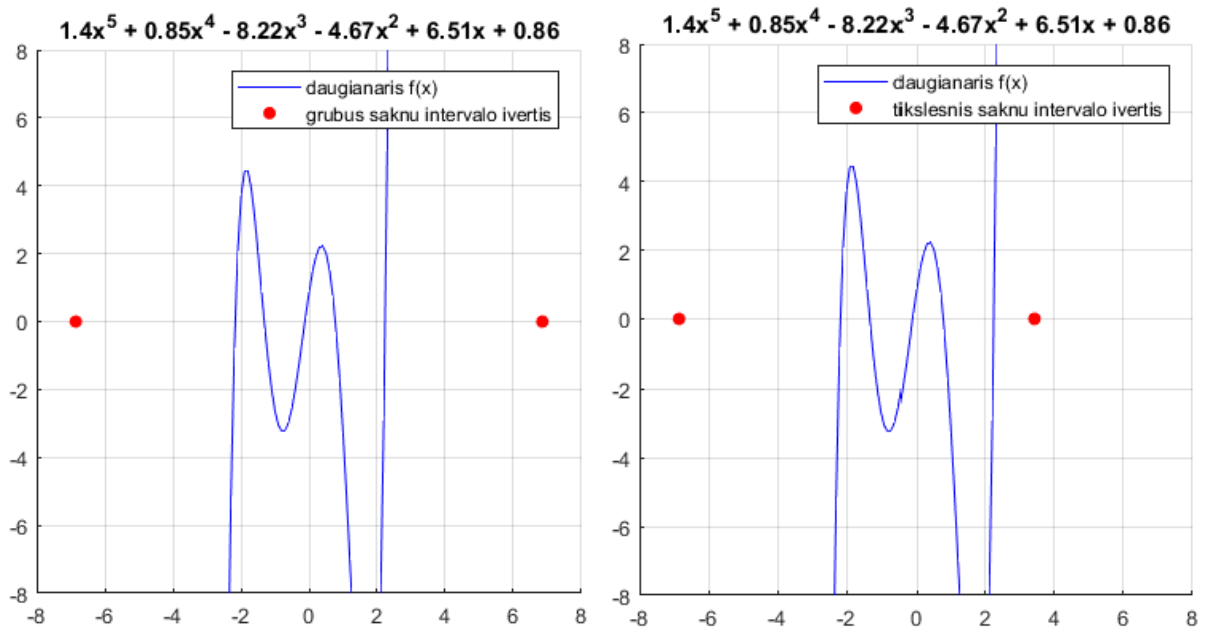
1. Netiesinių lygčių sprendimas

Duotos dvi netiesinės lygtys: daugianaris $f(x) = 0$ ir transcendentinė funkcija $g(x) = 0$.

Nr.	Daugianaris $f(x)$	Funkcija $g(x)$
23	$1.4x^5 + 0.85x^4 - 8.22x^3 - 4.67x^2 + 6.51x + 0.86$	$\cos(2x) e^{-\left(\frac{x}{2}\right)^2}; -6 \leq x \leq 6$
Sprendimo metodai: skenavimo, paprastųjų iteracijų, Kvazi-Niutono (kirstinių)		

1.1. Lygties $f(x) = 0$ ($f(x)$ – daugianaris) sprendimas

- Daugianario šaknų intervalo įverčiai**



1 pav. Daugianario šaknų intervalo grubūs įverčiai (1 grafas) ir tikslesni įverčiai (2 grafas)

Grubus lygties $f(x) = 0$ šaknų intervalo įvertis	$[-6.8714; 6.8714]$
Tikslesnis lygties $f(x) = 0$ šaknų intervalo įvertis	$[-6.8714; 3.4231]$

Lentelė 1. Šaknų intervalo įverčiai.

- Komentarai**

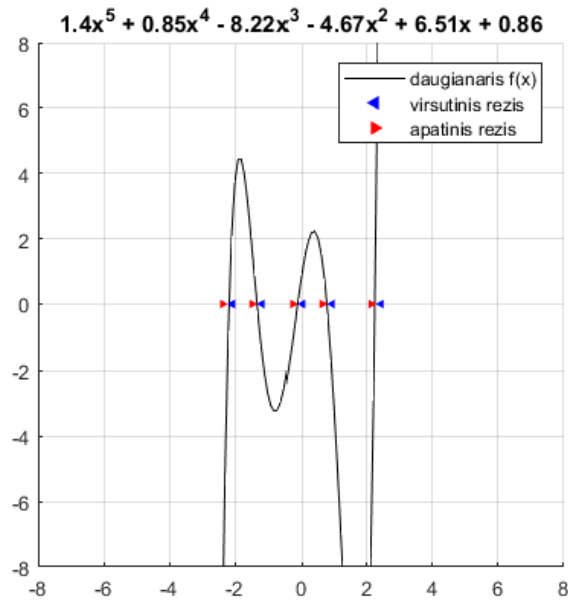
Grubus įvertis gavosi gana tikslus.

Apatinis grubaus įverčio režis net gavosi toks pat kaip ir tikslesnio.

Abu įverčiai apima visas šaknis.

- **Šaknų atskyrimas skenavimo metodu**

Skenavimas atliekamas intervale $[-6,8714; 3,4231]$, skenavimo žingsnis lygus 0,3.



2 pav. Daugianario šaknų atskyrimo intervalai

Intervalo Nr.	Intervalas
1	$[-2.3714000; -2.0714000]$
2	$[-1.4714000; -1.1714000]$
3	$[-0.2714000; 0.0286000]$
4	$[0.6286000; 0.9286000]$
5	$[2.1286000; 2.4286000]$

Lentelė 2. Šaknies atskyrimo intervalai.

- **Komentarai**

Skenavimo žingsnis parinktas 0,3, kad sumažinti šaknies peršokimo tikimybę. Šaknų neperšoko ir rado visas 5 šaknis.

Skaitiniai metodai ir algoritmai (P170B115). Lukas Šivickas (IFF-6/8). Varianto Nr. 23

- **Šaknų tikslinimas skenavimo, paprastųjų iteracijų ir Kvazi-Niutono(kirstinių) metodais**
Tariama, kad x_g yra šaknis (stabdomi skaičiavimai), jei $|f(x_g)| < 1e - 9$. Skaičiavimuose naudojamas šaknies tikslumo įvertis $|f(x_g)|$.

Skenavimo metodas	Pradinis intervalas	-	Šaknis (x_g)	Tikslumas ($ f(x_g) $)	Iteracijų skaičius
	[-2.3714000; -2.0714000]	-	-2.17699252109	0.00000000091	46
	[-1.4714000; -1.1714000]	-	-1.31947233755	0.00000000085	41
	[-0.2714000; 0.0286000]	-	-0.12355878785	0.00000000025	40
	[0.6286000; 0.9286000]	-	0.77255218603	0.00000000035	36
	[2.1286000; 2.4286000]	-	2.24032860337	0.00000000046	47
Pap. iteracijų metodas	Pradinis artinys	α reikšmė	Šaknis (x_g)	Tikslumas ($ f(x_g) $)	Iteracijų skaičius
	-2.2214	-30	-2.17699252106	0.00000000012	9
	-1.3214	10	-1.31947233761	0.00000000016	7
	-0.1214	-7.5	-0.12355878784	0.00000000033	5
	0.7786	11.4	0.77255218606	0.00000000001	4
	2.2786	-76.25	2.24032860336	0.00000000000	5
Kirstinių metodas	Pradiniai artiniai	-	Šaknis (x_g)	Tikslumas ($ f(x_g) $)	Iteracijų skaičius
	-2.3714000; -2.0714000	-	-2.17699252107	0.00000000000	7
	-1.4714000; -1.1714000	-	-1.31947233762	0.00000000000	5
	-0.2714000; 0.0286000	-	-0.12355878785	0.00000000019	4
	0.6286000; 0.9286000	-	0.77255218606	0.00000000000	6
	2.1286000; 2.4286000	-	2.24032860336	0.00000000000	6
MATLAB funkcijos	Pradinis artinys	-	Šaknis (fzero)	Šaknis (roots)	Iteracijų skaičius
	-2.2214	-	-2.17699252107	-2.17699252107	-
	-1.3214	-	-1.31947233762	-1.31947233762	-
	-0.1214	-	-0.12355878788	-0.12355878788	-
	0.7786	-	0.77255218606	0.77255218606	-
	2.2786	-	2.24032860336	2.24032860336	-

Lentelė 3. Rezultatų lentelė

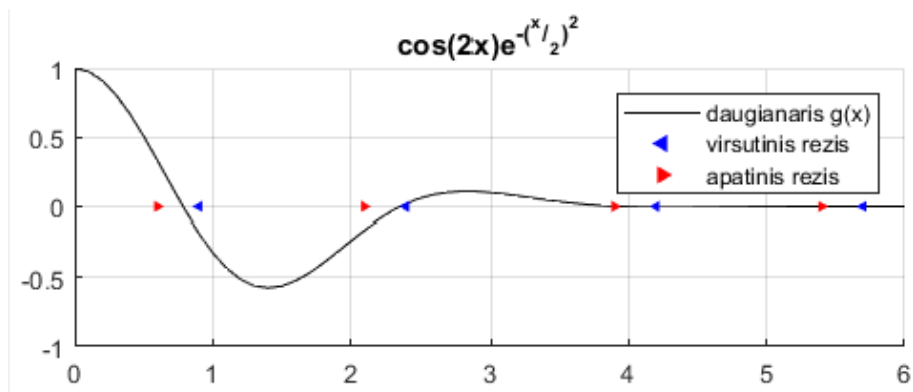
• **Komentarai**

Metodas	Komentaras
Skenavimo metodas	Tiesmukiškiausias metodas iš visų, kuris puikiai veikia, nereikia labai tikslaus intervalo, kad rasti šaknį, tačiau užtrunka gana daug iteracijų, kol randama šaknis, jei intervalas sąlyginai didelis ar paklaida nuo 0 yra sąlyginai maža.
Pap. iteracijų metodas	Šaknis randa greičiau, nei skenavimo metodas, nes reikia mažiau iteracijų, tačiau reikia parinkti tinkamą α reikšmę kiekvienai šakniai, o parinkus netinkamą α reikšmę šaknies nerasta.
Kirstinių metodas	Šaknis randa greičiau, nei skenavimo metodas, panašiai kaip pap. iteracijų metodas. Paprastesnis nei pap. iter. metodas, nes nereikia parinkti α , tačiau šaknies gali nerasti, jei parenkami netinkami pradiniai artiniai. Taip pat šaknys randamos geresniu tikslumu.

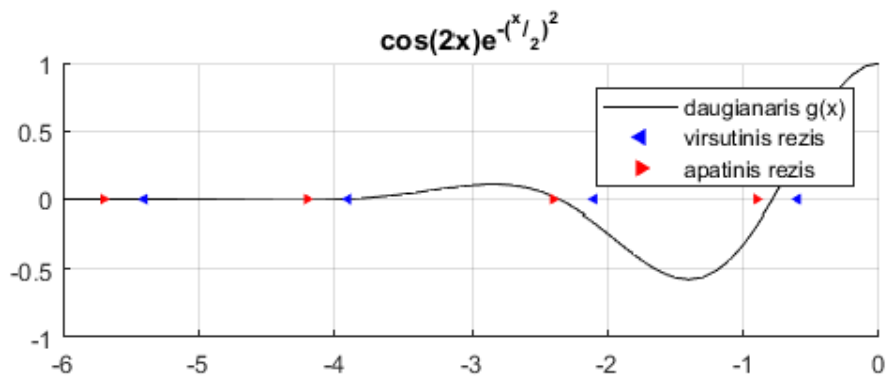
1.2. Lygties $g(x) = 0$ ($g(x)$ – transcendentinė funkcija) sprendimas

- Šaknų atskyrimas skenavimo metodu

Skenavimas atliekamas intervale $[-6; 6]$, skenavimo žingsnis lygus 0,3.



4 pav. Šaknų atskyrimo intervalai, kai $0 < x \leq 6$



3 pav. Šaknų atskyrimo intervalai, kai $-6 \leq x \leq 0$

Intervalo Nr.	Intervalas
1	$[-5.700; -5.400]$
2	$[-4.200; -3.900]$
3	$[-2.400; -2.100]$
4	$[-0.900; -0.600]$
5	$[0.600; 0.900]$
6	$[2.100; 2.400]$
7	$[3.900; 4.200]$
8	$[5.400; 5.700]$

Lentelė 5. Šaknies atskyrimo intervalai.

- Komentarai

Skenavimo žingsnis parinktas 0,3, kad sumažinti šaknies peršokimo tikimybę.

Šaknų neperšoko ir rado visas 8 šaknis.

- **Šaknų tikslinimas skenavimo, paprastųjų iteracijų ir Kvazi-Niutono(kirstinių) metodais**
Tariama, kad x_g yra šaknis (stabdomi skaičiavimai), jei $|f(x_g)| < 1e - 9$. Skaičiavimuose naudojamas šaknies tikslumo įvertis $|f(x_g)|$.

Skenavimo metodas	Pradinis intervalas	-----	Šaknis (x_g)	Tikslumas ($ f(x_g) $)	Iteracijų skaičius
	[-5.700; -5.400]	-	-5.49778747559	0.00000000035	22
	[-4.200; -3.900]	-	-3.92699079514	0.00000000093	34
	[-2.400; -2.100]	-	-2.35619449019	0.00000000000	31
	[-0.900; -0.600]	-	-0.78539816290	0.00000000085	38
	[0.600; 0.900]	-	0.78539816290	0.00000000085	40
	[2.100; 2.400]	-	2.35619449019	0.00000000000	41
	[3.900; 4.200]	-	3.92699079514	0.00000000093	26
	[5.400; 5.700]	-	5.49778747559	0.00000000035	23
Pap. iteracijų metodas	Pradinis artinys	α reikšmė	Šaknis (x_g)	Tikslumas ($ f(x_g) $)	Iteracijų skaičius
	-5.550	0.001	-5.49778681649	0.00000000034	4
	-4.050	-0.041	-3.92699080313	0.00000000059	5
	-2.250	0.5	-2.35619449083	0.00000000032	4
	-0.750	-1.75	-0.78539816335	0.00000000008	5
	0.750	1.75	0.78539816335	0.00000000008	5
	2.250	-0.5	2.35619449083	0.00000000032	4
	4.050	0.041	3.92699080313	0.00000000059	5
	5.550	-0.001	5.49778681649	0.00000000034	4
Kirstinių metodas	Pradiniai artiniai	-----	Šaknis (x_g)	Tikslumas ($ f(x_g) $)	Iteracijų skaičius
	-5.700; -5.400	-	-5.49778714340	0.00000000000	6
	-4.200; -3.900	-	-3.92699081700	0.00000000000	5
	-2.400; -2.100	-	-2.35619449019	0.00000000000	5
	-0.900; -0.600	-	-0.78539816349	0.00000000015	4
	0.600; 0.900	-	0.78539816358	0.00000000031	4
	2.100; 2.400	-	2.35619449019	0.00000000000	5
	3.900; 4.200	-	3.92699081699	0.00000000000	6
	5.400; 5.700	-	5.49778733092	0.00000000020	6
MATLAB funkcijos	Pradinis artinys	-----	Šaknis (fzero)	-----	Iteracijų skaičius
	-5.550	-	-5.49778714378	-	-
	-4.050	-	-3.92699081699	-	-
	-2.250	-	-2.35619449019	-	-
	-0.750	-	-0.78539816340	-	-
	0.750	-	0.78539816340	-	-
	2.250	-	2.35619449019	-	-
	4.050	-	3.92699081699	-	-
	5.550	-	5.49778714378	-	-

• **Komentarai**

Metodas	Komentaras
Skenavimo metodas	Tiesmukiškiausias metodas iš visų, kuris puikiai veikia, nereikia labai tikslaus intervalo, kad rasti šaknį, tačiau užtrunka gana daug iteracijų, kol randama šaknis, jei intervalas sąlyginai didelis ar paklaida nuo 0 yra sąlyginai maža.
Pap. iteracijų metodas	Šaknis randa greičiau, nei skenavimo metodas, nes reikia mažiau iteracijų, tačiau reikia parinkti tinkamą α reikšmę kiekvienai šakniai, o parinkus netinkamą α reikšmę šaknies nerasta.
Kirstinių metodas	Šaknis randa greičiau, nei skenavimo metodas, panašiai kaip pap. iteracijų metodas. Paprastesnis nei pap. iter. metodas, nes nereikia parinkti α , tačiau šaknies gali nerasti, jei parenkami netinkami pradiniai artiniai. Taip pat šaknis randamos geresniu tikslumu.

Lentelė 7. Komentarai

2. Sąlyginio uždavinio sprendimas

2.1. Uždavinio sąlyga

Skysčio tūris V sferos formos talpoje priklauso nuo skysčio lygio (aukščio) h pagal dėsnį:

$$V(h) = \frac{\pi h^2(3r - h)}{3}$$

Koks yra skysčio aukštis h talpoje, jeigu žinomas r , V ?

Varianto Nr.	r, m	V, m^3
23	2	2

Lentelė 8. Variantų duomenys

2.2. Sprendimas

Sprendžiama lygtis $2 = \frac{\pi h^2(6-h)}{3}$, čia t – kintamasis.

Sudaroma funkcija $f(h) = \pi h^2(6 - h) - 6$ ir skenavimo metodu ieškoma, kur funkcija kerta X ašį.

• **Funkcijos $f(h)$ sudarymas**

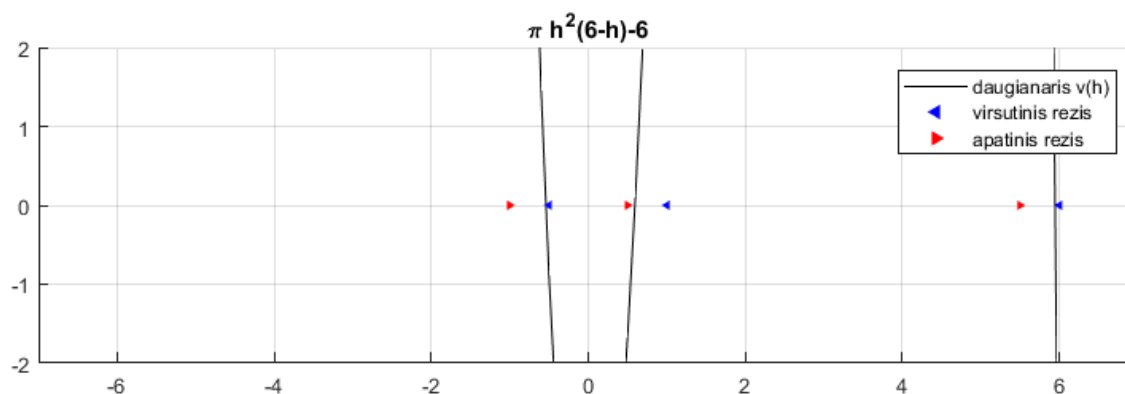
- $2 = \frac{\pi h^2(6-h)}{3} \mid * 3$
- $6 = \pi h^2(6 - h) \mid - 6$
- $\pi h^2(6 - h) - 6 = 0$

• **Skenavimo metodo pasirinkimo argumentavimas**

Skenavimo metodą pasirinkau, nes su juo paprasčiausia dirbti, nereikia rinktis papildomų metodo parametrų. Taip, užtruks ilgiau, bet paprasčiau.

• **Šaknų atskyrimas skenavimo metodu**

Skenavimas atliekamas intervale $[-10; 10]$, skenavimo žingsnis lygus 0,5.



5 pav. Šaknų atskyrimo intervalai

Intervalo Nr.	Intervalas
1	$[-1.000; -0.500]$
2	$[0.500; 1.000]$
3	$[5.500; 6.000]$

Lentelė 9. Šaknies atskyrimo intervalai.

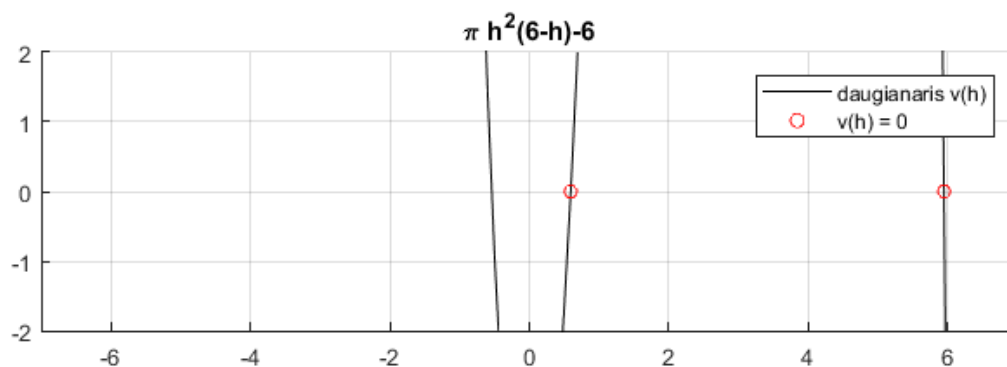
• **Šaknų tikslinimas skenavimo metodu**

Tariama, kad x_g yra šaknis (stabdomi skaičiavimai), jei $|f(x_g)| < 1e - 9$. Skaičiavimuose naudojamas šaknies tikslumo įvertis $|f(x_g)|$.

Skenavimo metodas	Pradinis intervalas	Šaknis (x_g)	Tikslumas ($ f(x_g) $)	Iteracijų skaičius
	$[-1.000; -0.500]$	-0.54037991753	0.00000000076	44
	$[0.500; 1.000]$	0.59439990604	-0.00000000024	45
	$[5.500; 6.000]$	5.94598001144	0.00000000016	53

Lentelė 10. Rezultatai

Kadangi aukštis gali būti tik teigiamas pirmos šaknies nenaudojame.



6 pav. Sąlyginio uždavinio rezultatai pavaizduoti grafiškai.

3. Išvados

Nagrinėjome 3 netiesinių lygčių šaknų radimo metodus – skenavimo, paprastųjų iteracijų ir Kvazi-Niutono (kirstinių). Rezultatų lentelės parodė, kad visi 3 metodai randa tas pačias šaknis, visi veikia teisingai, tik skiriasi jų greitis (iteracijų kiekis) ir šaknų tikslumas. Iš visų šių 3-jų metodų galima teigti, kad Kvazi-Niutono (kirstinių) metodas yra tiksliausias ir greičiausias (mažiausiai iteracijų).

4. Programų tekstai

4.1. Daugianario šaknų intervalo įverčių nustatymas

```
// Atskyrimas skenujimu
private void button2_Click_1(object sender, EventArgs e)
{
    ClearForm(); // išvalomi programos duomenys
    PreparareForm(-7, 7, -5, 5);
    // Nubraižoma f-ja, kuriai ieskome saknies
    Fx = chart1.Series.Add("F(x)");
    Fx.ChartType = SeriesChartType.Line;
    // NUO KOKIO X PIESIA
    double x = -6;
    double atstumas = 12;
    int iter_sk = 2000;
    for (int i = 0; i < iter_sk; i++)
    {
        Fx.Points.AddXY(x, F(x)); x = x + atstumas / iter_sk;
    }
    Fx.BorderWidth = 3;

    chart1.Series.Add("Apatinis rėžis");
    chart1.Series[1].MarkerStyle = MarkerStyle.Circle;
    chart1.Series[1].MarkerSize = 8;
    chart1.Series[1].ChartType = SeriesChartType.Point;

    chart1.Series.Add("Viršutinis rėžis");
    chart1.Series[2].MarkerStyle = MarkerStyle.Circle;
    chart1.Series[2].ChartType = SeriesChartType.Point;
    chart1.Series[2].MarkerSize = 8;

    double[] ar = new double[8];
    double[] vr = new double[8];
    double[] artiniai = new double[8];

    double x0 = double.Parse(tb_skenx0.Text);
    double x1 = double.Parse(tb_skenx1.Text);
    double step = double.Parse(tb_skenstep.Text);
    int saknu_count = 0;
    //richTextBox1.AppendText("      I      x0      F(x0)      x1
F(x1)\n");
    for (int i = 1; i < 1000; i++)
    {
        if (Math.Sign(F(x0)) != Math.Sign(F(x0 + step)))
        {
            //x1 = x0 + step;
            chart1.Series[1].Points.AddXY(x0, 0);
            chart1.Series[2].Points.AddXY(x0 + step, 0);
            ar[saknu_count] = x0;
            vr[saknu_count] = x0 + step;
            artiniai[saknu_count] = (x0 + step + x0) / 2;
            saknu_count++;

            richTextBox1.AppendText(String.Format("{0}. Rasta {1}-a šaknis intervale
[{2,12:f9}; {3,12:f9}]; Artinys = [{4,12:f9}]\n", i, saknu_count, x0, x0+step, artiniai[saknu_count-
1]));
            x0 += step;
        }
        else
        {
            x0 += step;
        }
    }
}
```

```

    }
    if (x0 >= x1)
    {
        richTextBox1.AppendText(String.Format("{0}. Pabaiga nes x0(={1}) >= x1(={2})\n",
i, x0, x1));
        richTextBox1.AppendText(String.Format("vr = ["));
        for (int d = 0; d < vr.Length; d++)
        {
            richTextBox1.AppendText(vr[d].ToString());
            if(d+1 < vr.Length)
            {
                richTextBox1.AppendText(", ");
            }
        }
        richTextBox1.AppendText(String.Format("];\n"));
        {
        }
        richTextBox1.AppendText(String.Format("ar = ["));
        for (int d = 0; d < ar.Length; d++)
        {
            richTextBox1.AppendText(ar[d].ToString());
            if (d + 1 < vr.Length)
            {
                richTextBox1.AppendText(", ");
            }
        }
        richTextBox1.AppendText(String.Format("];\n"));
        richTextBox1.AppendText(String.Format("artin = ["));
        for (int d = 0; d < artiniai.Length; d++)
        {
            richTextBox1.AppendText(artiniai[d].ToString());
            if (d + 1 < vr.Length)
            {
                richTextBox1.AppendText(", ");
            }
        }
        richTextBox1.AppendText(String.Format("];"));
        break;
    }
}
}

```

Lentelė 11

4.2. Skenavimo metodas

```

// Skenavimas
private void button7_Click(object sender, EventArgs e)
{
    ClearForm(); // išvalomi programos duomenys
    PreparareForm(-7, 7, -5, 5);
    // Nubraižoma f-ja, kuriai ieskome saknies
    Fx = chart1.Series.Add("F(x)");
    Fx.ChartType = SeriesChartType.Line;
    Fx.Color = Color.Black;
    // NUO KOKIO X PIESIA
    double x = -6;
    double atstumas = 12;
    int iter_sk = 2000;
    for (int i = 0; i < iter_sk; i++)
    {

```

```

        Fx.Points.AddXY(x, F(x)); x = x + atstumas / iter_sk;
    }
    Fx.BorderWidth = 3;

    chart1.Series.Add("x0");
    chart1.Series[1].MarkerStyle = MarkerStyle.Circle;
    chart1.Series[1].MarkerSize = 8;
    chart1.Series[1].ChartType = SeriesChartType.Point;
    chart1.Series[1].ChartType = SeriesChartType.Line;
    chart1.Series[1].Color = Color.LightGray;

    chart1.Series.Add("x1");
    chart1.Series[2].MarkerStyle = MarkerStyle.Circle;
    chart1.Series[2].ChartType = SeriesChartType.Point;
    chart1.Series[2].ChartType = SeriesChartType.Line;
    chart1.Series[2].Color = Color.Gray;
    chart1.Series[2].MarkerSize = 8;

    chart1.Series.Add("x_mid");
    chart1.Series[3].MarkerStyle = MarkerStyle.Circle;
    chart1.Series[3].ChartType = SeriesChartType.Point;
    chart1.Series[3].ChartType = SeriesChartType.Line;
    chart1.Series[3].Color = Color.Red;
    chart1.Series[3].MarkerSize = 8;

    double x0 = double.Parse(tb_x0.Text.Replace(',', '.'));
    double x1 = double.Parse(tb_x1.Text.Replace(',', '.'));
    double step = 0.10;
    double stepReductionCoef = 2;
    richTextBox1.AppendText("      I      x0      F(x0)      x1
F(x1)\n");
    for (int i = 1; i < 1000; i++)
    {
        if(Math.Sign(F(x0)) != Math.Sign(F(x0+step)))
        {
            x1 = x0 + step;
            chart1.Series[2].Points.AddXY(x1, 0);
            step /= stepReductionCoef;
        } else
        {
            x0 += step;
            chart1.Series[1].Points.AddXY(x0, 0);
        }
        richTextBox1.AppendText(String.Format(" {0,6:d}      {1,12:f9}      {2,12:f9}      {3,12:f9}
{4,12:f9}\n", i, x0, F(x0), x1, F(x1)));

        if (Math.Abs(F((x0 + x1)/2)) < 1e-9)
        {
            chart1.Series[2].Points.AddXY(x0, 0);
            richTextBox1.AppendText(String.Format("Pabaiga. Rasta saknis (x = {0:f11}, f(x) =
{1:f11}) per {2:d} iteracijas(-a).\n", (x0 + x1) / 2, F((x0 + x1) / 2), i));
            richTextBox1.AppendText(String.Format("{0:f11}\n{1:f11}\n{2:d}\n", (x0 + x1) / 2,
F((x0 + x1) / 2), i));
            chart1.Series[3].Points.AddXY((x0 + x1) / 2, F((x0 + x1) / 2));
            break;
        }
    }
}

```

4.3. Paprastųjų iteracijų metodas

```

// Paprastu iteraciju
private void button8_Click(object sender, EventArgs e)
{
    ClearForm(); // išvalomi programos duomenys
    PreparareForm(-7, 7, -5, 5);
    // Nubraižoma f-ja, kuriai ieskome saknies
    Fx = chart1.Series.Add("F(x)");
    Fx.Color = Color.Purple;
    Fx.ChartType = SeriesChartType.Line;
    // NUO KOKIO X PIESIA
    double x = -6;
    double atstumas = 12;
    int iter_sk = 2000;
    for (int i = 0; i < iter_sk; i++)
    {
        Fx.Points.AddXY(x, F(x)); x = x + atstumas / iter_sk;
    }
    Fx.BorderWidth = 3;

    AddYZeroLine();

    Series sk = chart1.Series.Add("x + (F(x) / alpha)");
    sk.MarkerStyle = MarkerStyle.Circle;
    sk.MarkerSize = 8;
    sk.ChartType = SeriesChartType.Point;
    sk.ChartType = SeriesChartType.Line;
    sk.Color = Color.Cyan;

    Series rez = chart1.Series.Add("x_mid");
    rez.MarkerStyle = MarkerStyle.Circle;
    rez.Color = Color.Red;
    rez.MarkerSize = 8;

    double x0 = double.Parse(tb_art.Text);
    double alpha = double.Parse(tb_alpha.Text);
    richTextBox1.AppendText(String.Format(" {0,4:s} {1,16:s} {2,16:s}\n", "I", "x", "F(x)"));
    double xTemp = x0;
    for (int i = 1; i < 1000; i++)
    {
        xTemp = xTemp + (F(xTemp) / alpha);
        sk.Points.AddXY(xTemp, F(xTemp));

        richTextBox1.AppendText(String.Format(" {0,4:d} {1,16:f11} {2,16:f11}\n", i, xTemp,
F(xTemp)));

        if (Math.Abs(F(xTemp)) < 1e-9)
        {
            rez.Points.AddXY(xTemp, 0);
            richTextBox1.AppendText(String.Format("Pabaiga. Rasta saknis (x = {0:f11}; F(x) =
{1:f11}) per {2:d} iteracijas(-a).\n", xTemp, F(xTemp), i));
            richTextBox1.AppendText(String.Format("{0:f11}\n{1:f11}\n{2:d}\n", xTemp,
F(xTemp), i));
            break;
        }
        if (xTemp > 10 || xTemp < -10)
        {
            richTextBox1.AppendText("RIP");
            break;
        }
    }
}

```

```

    }
}

```

Lentelė 13

4.4. Kvazi-Niutono (kirstinių) metodas

```

// Kvazi-Niutono
private void button6_Click(object sender, EventArgs e)
{
    ClearForm(); // išvalomi programos duomenys
    PreparareForm(-7, 7, -5, 5);
    // Nubraižoma f-ja, kuriai ieskome saknies
    Fx = chart1.Series.Add("F(x)");
    Fx.ChartType = SeriesChartType.Line;
    // NUO KOKIO X PIESIA
    double x = -6;
    double atstumas = 12;
    int iter_sk = 2000;
    for (int i = 0; i < iter_sk; i++)
    {
        Fx.Points.AddXY(x, F(x)); x = x + atstumas / iter_sk;
    }
    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("XMid");
    XMid.MarkerStyle = MarkerStyle.Circle;
    X1X2.ChartType = SeriesChartType.Point;
    X1X2.ChartType = SeriesChartType.Line;
    XMid.MarkerSize = 8;

    double x0 = double.Parse(tb_knx0.Text);
    double x1 = double.Parse(tb_knx1.Text);
    richTextBox1.AppendText(String.Format(" {0,4:s}    {1,16:s} {2,16:s} {2,16:s}
{2,16:s}\n", "I", "x1", "F(x1)", "x0", "F(x0)"));
    double xTemp0 = x0;
    double xTemp = x1;
    double xTemp1 = x1;
    for (int i = 1; i < 1000; i++)
    {
        xTemp = xTemp1;
        xTemp1 = xTemp1 - Math.Pow(((F(xTemp1) - F(xTemp0)) / (xTemp1 - xTemp0)), -1) *
F(xTemp1);
        xTemp0 = xTemp;

        chart1.Series[1].Points.AddXY(xTemp1, F(xTemp1));

        richTextBox1.AppendText(String.Format(" {0,4:d}    {1,16:f11} {2,16:f11} {2,16:f11}
{2,16:f11}\n", i, xTemp1, F(xTemp1), xTemp0, F(xTemp0)));

        if (Math.Abs(F(xTemp1)) < 1e-9)
        {
            chart1.Series[2].Points.AddXY(xTemp1, 0);
            richTextBox1.AppendText(String.Format("Pabaiga. Rasta saknis (x = {0:f11}); f(x) =
{1:f11}) per {2:d} iteracijas(-a).\n", xTemp1, F(xTemp1), i));

```

```
richTextBox1.AppendText(String.Format("{0:f11}\n{1:f11}\n{2:d}.\n", xTemp1,  
F(xTemp1), i));  
    break;  
}  
if (xTemp0 > 10 || xTemp0 < -10)  
{  
    richTextBox1.AppendText("RIP");  
    break;  
}  
}
```