

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**COMPUTER DEPARTMENT**

### Skaitinių metodų ir algoritmų 4-a projektinė užduotis

**Darbą atliko:**

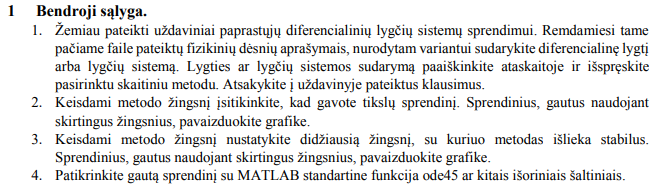
IFF 6/8 grupės studentas

Tadas Laurinaitis

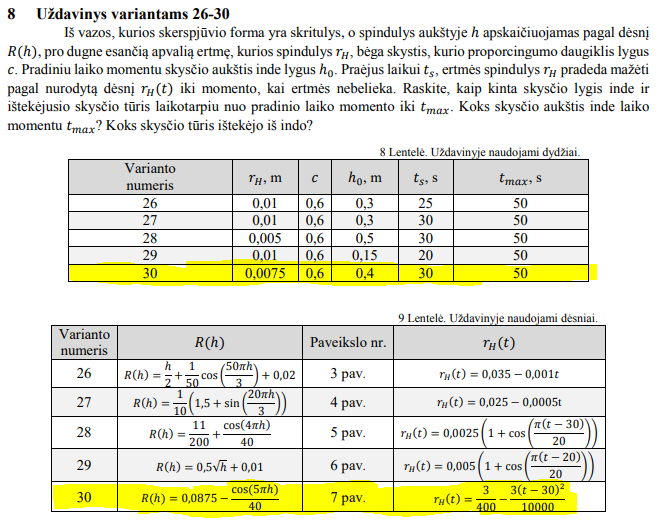
**Darbą vertino**:

Lekt. Dalia Čalnerytė

**Užduotys**



Pav. #1 Užduočių sąrašas

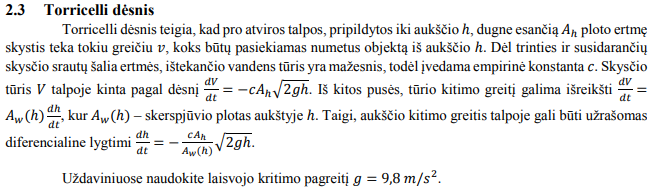


Pav. #2 Užduočių variantų sąrašas

**Užduočių sprendimai:**

Spręstas variantas: 30 (Uždavinys, kintamieji ir formulės matomi Pav. #2)

Diferencialinėms lygtims sudaryti buvo naudojamas Torricelli dėsnis, į jį buvo įsistatyti man duotos funkcijos R(h) ir rH(t) :



Diferencialinės lygties realizacija:

public double dhdt(double h, double t, double c)

{

double g = 9.8;

double dhdt = -1 \* ((c \* Math.Pow(rHt(t), 2)) / Math.Pow(Rh(h), 2)) \* (Math.Sqrt((2\*g\*h)));

return dhdt;

}

R(h) formulės realizacija:

public double Rh(double h)

{

double Rh = 0;

Rh = 0.0875 - (Math.Cos(5 \* Math.PI \* h) / 40);

return Rh;

}

𝑟𝐻(𝑡) formulės realizacija:

public double rHt(double t)

{

double rHt = 0;

if(t <= 30)

{

rHt = 0.0075;

richTextBox1.AppendText(rHt.ToString());

}

else if (t > 30 && t <= 50)

{

rHt = (3.0 / 400) - ((3 \* Math.Pow(t - 30, 2)) / 10000);

if(rHt < 0)

{

rHt = 0;

}

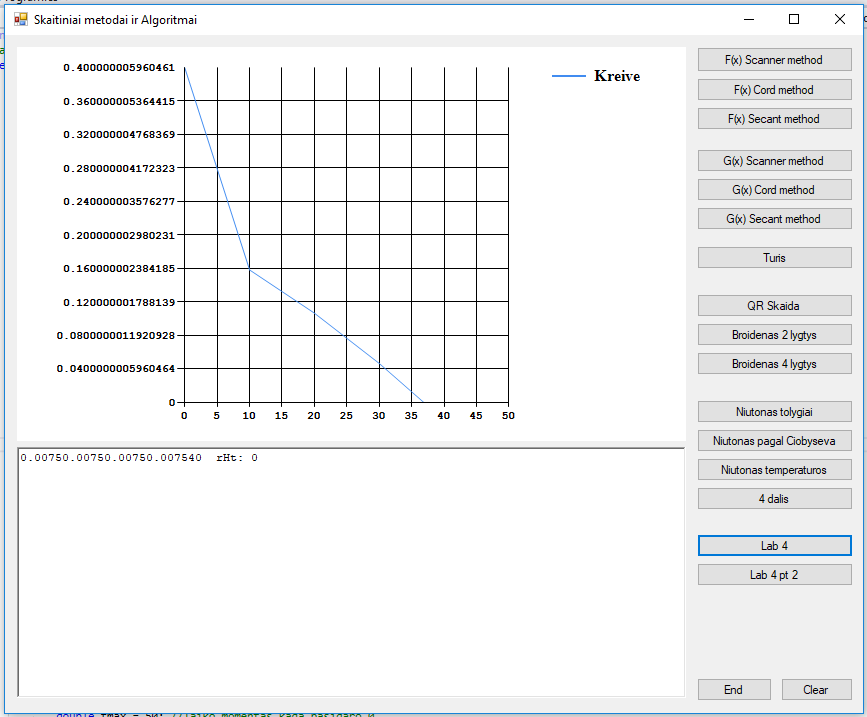
richTextBox1.AppendText(t.ToString() + " rHt: " +rHt.ToString() +"\n");

}

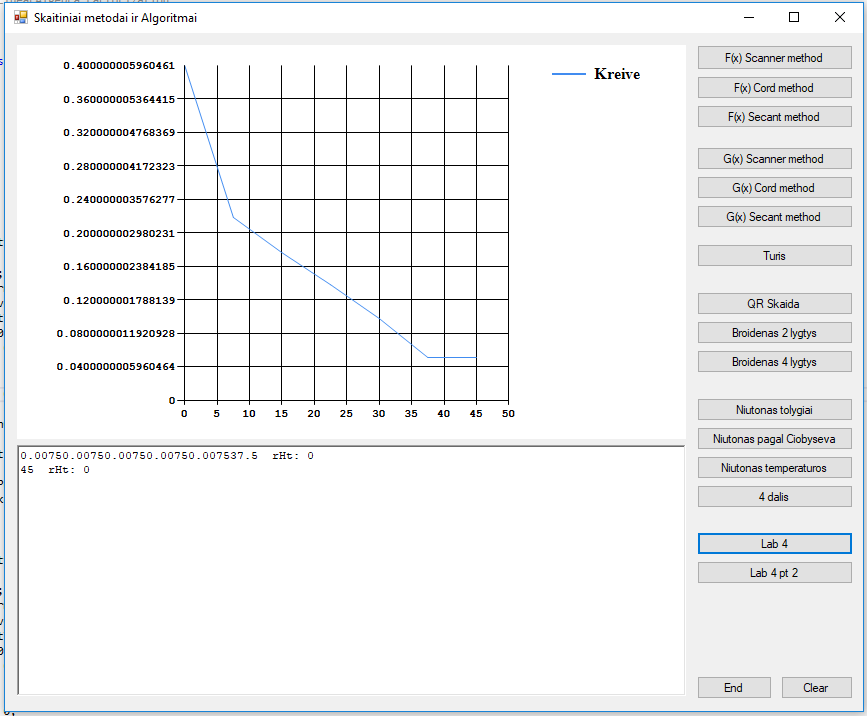
return rHt;

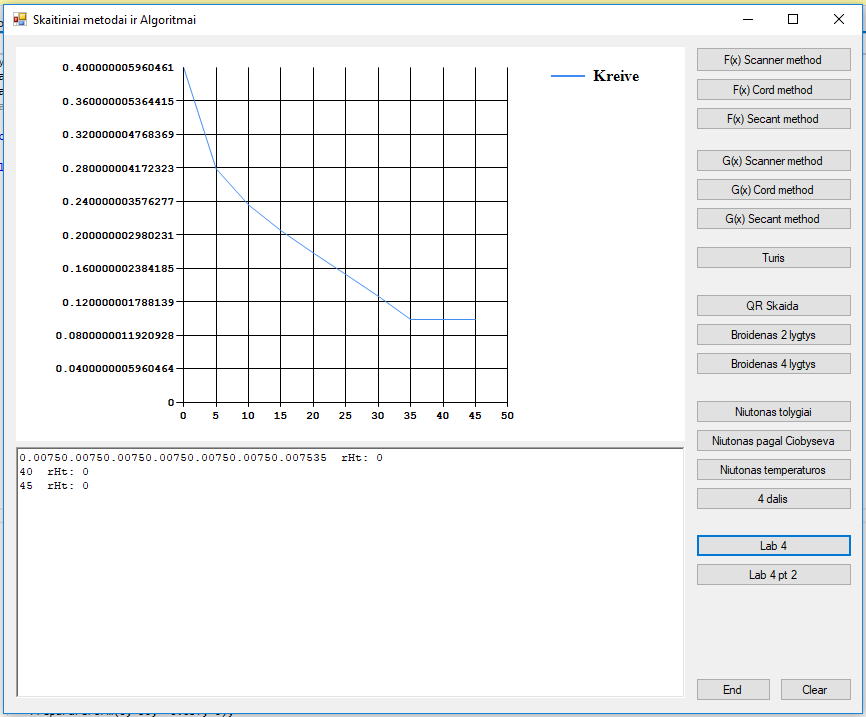
}

-------------------------------------------------------------------------------------------------------------------------------

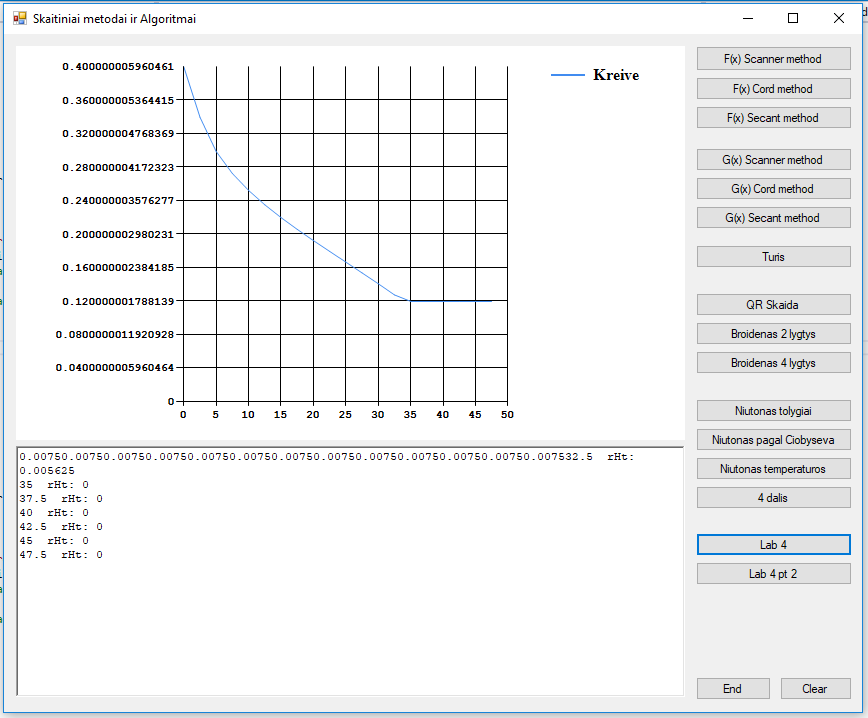


Pav. #3 Grafikas parodantis skysčio lygį inde naudojant žingsnį, kurio dydis yra 10

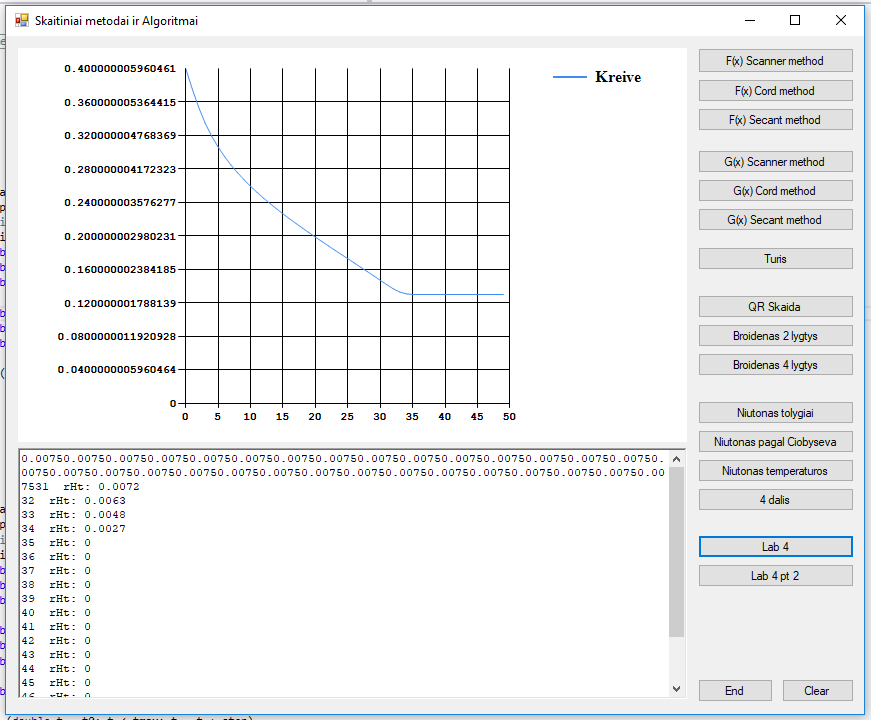
Pav. #4 Grafikas parodantis skysčio lygį inde naudojant žingsnį, kurio dydis yra 7.5



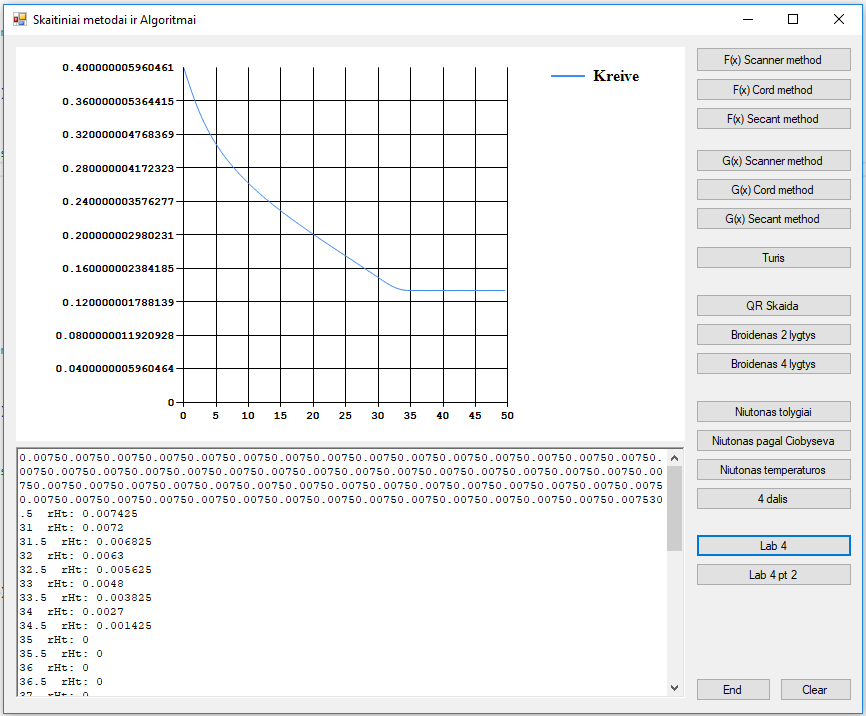
Pav. #5 Grafikas parodantis skysčio lygį inde naudojant žingsnį, kurio dydis yra 5



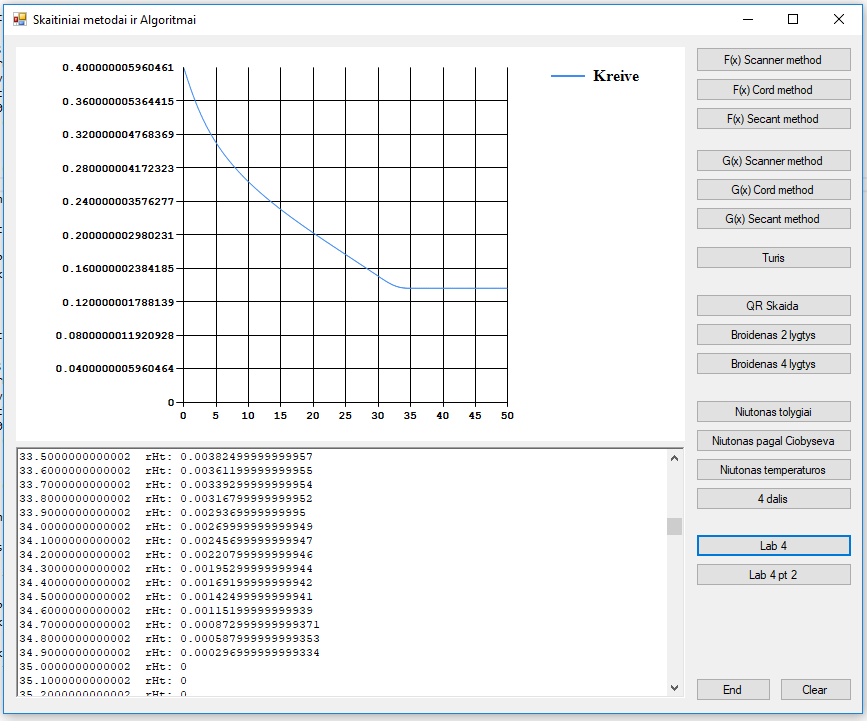
Pav. #6 Grafikas parodantis skysčio lygį inde naudojant žingsnį, kurio dydis yra 2.5



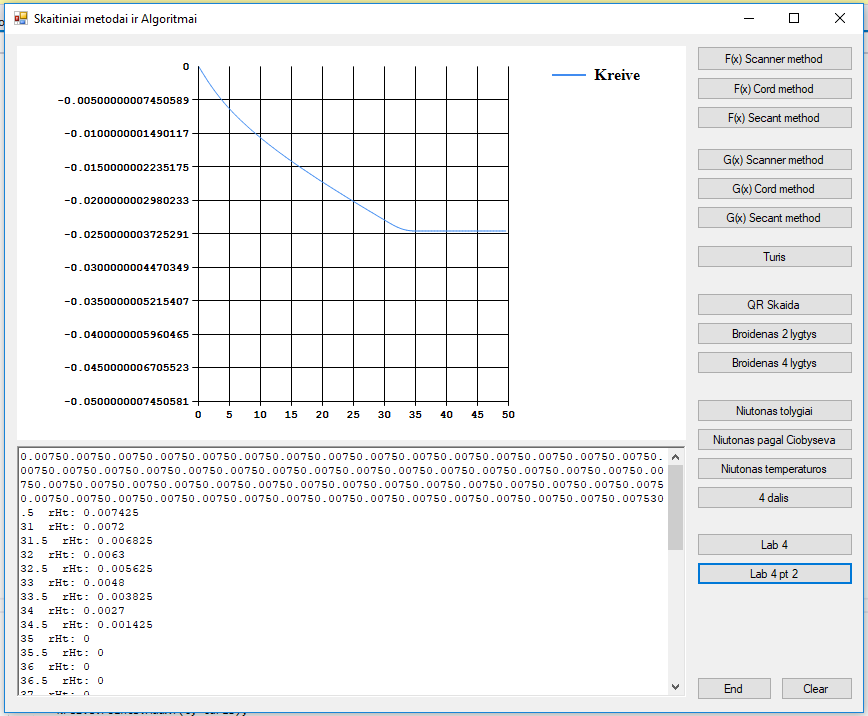
Pav. #7 Grafikas parodantis skysčio lygį inde naudojant žingsnį, kurio dydis yra 1



Pav. #8 Grafikas parodantis skysčio lygį inde naudojant žingsnį, kurio dydis yra 0.5



Iš grafikų pastebime, kad šiam uždaviniui spręsti optimalus žingsnis yra 0.5, kadangi po šio žingsnio, grafiko pasikeitimai yra beveik nepastebimi, o rezultatas t.y. skysčio lygis išlieka toks pats.



Pav. #8 Grafikas, kuriame parodomas išbėgusio skysčio tūris priklausomai nuo laiko, naudojant žingsnį, kurio dydis 0.5

Skysčio aukštis momentu tmax ~ 0.139

Išbėgusio skysčio tūris ~ 0.245

**Darbo apibendrinimas:**

Darbo metu užduočiai buvo pritaikytas Torricelli dėsnis ir pagal jį sudaryta diferencialinė lygtis. Į šią lygtį buvo įstatytos turimos funkcijos, o vėliau duotos reikšmės ir galiausiai sukurti grafikai. Iš grafikų pastebėjome, kad optimaliausias šiam uždaviniui žingsnis yra 0.5, kadangi jį didinant nepastebimi jokie žymūs pokyčiai grafike ar atsakyme.

**Programos kodo fragmentai:**

private void Button17\_click(object sender, EventArgs e)

{

ClearForm();

PreparareForm(0, 50, 0f, 0.4f);

Series kreive = chart1.Series.Add("Kreive");

kreive.ChartType = SeriesChartType.Line;

double c = 0.6; // kazkokia konstanta

double h0 = 0.4; //pradinis aukstis

double tmax = 50; //laiko momentas kada pasidaro 0

double t0 = 0;

double step = 0.5;

double h = h0;

for(double t = t0; t < tmax; t = t + step)

{

kreive.Points.AddXY(t, h);

double kazkas = dhdt(h, t, c);

h = h + step \* kazkas;

}

}

private void Button18\_click(object sender, EventArgs e)

{

ClearForm();

PreparareForm(0, 50, -0.05f, 0);

Series kreive = chart1.Series.Add("Kreive");

kreive.ChartType = SeriesChartType.Line;

double c = 0.6; // kazkokia konstanta

double h0 = 0.4; //pradinis aukstis

double tmax = 50; //laiko momentas kada pasidaro 0

double t0 = 0;

double step = 0.5;

double h = h0;

double turis = 0;

for (double t = t0; t < tmax; t = t + step)

{

kreive.Points.AddXY(t, turis);

double kazkas = dhdt(h, t, c);

double kazkas2 = Rh(h) \* kazkas;

turis = turis + kazkas2 \* step;

h = h + step \* kazkas;

}

}

public double Rh(double h)

{

double Rh = 0;

Rh = 0.0875 - (Math.Cos(5 \* Math.PI \* h) / 40);

return Rh;

}

public double rHt(double t)

{

double rHt = 0;

if(t <= 30)

{

rHt = 0.0075;

richTextBox1.AppendText(rHt.ToString());

}

else if (t > 30 && t <= 50)

{

rHt = (3.0 / 400) - ((3 \* Math.Pow(t - 30, 2)) / 10000);

if(rHt < 0)

{

rHt = 0;

}

richTextBox1.AppendText(t.ToString() + " rHt: " +rHt.ToString() +"\n");

}

return rHt;

}

public double dhdt(double h, double t, double c)

{

double g = 9.8;

double dhdt = -1 \* ((c \* Math.Pow(rHt(t), 2)) / Math.Pow(Rh(h), 2)) \* (Math.Sqrt((2\*g\*h)));

return dhdt;

}

#endregion

}