

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**COMPUTER DEPARTMENT**

### Skaitinių metodų ir algoritmų 2-a projektinė užduotis

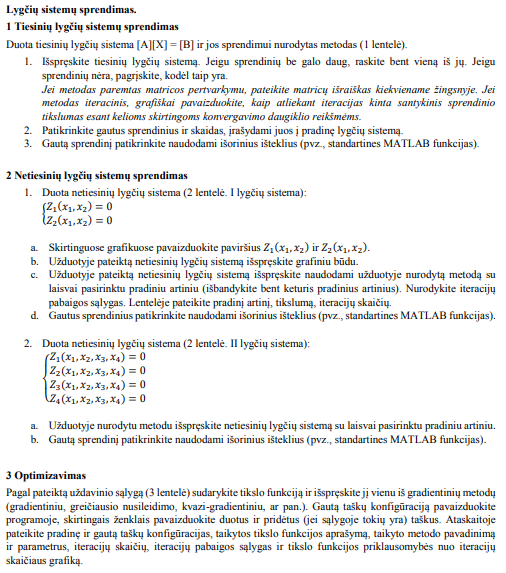
**Darbą atliko:**

IFF 6/8 grupės studentas

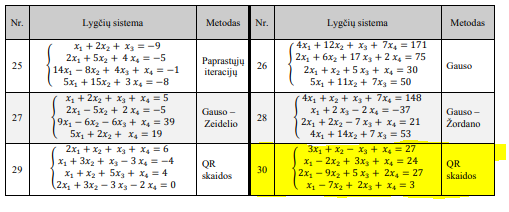
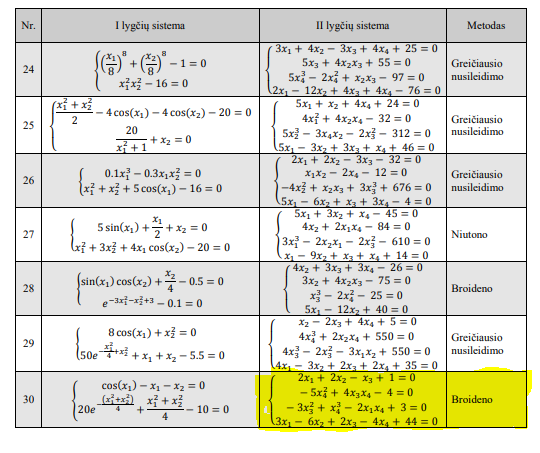
Tadas Laurinaitis

**Darbą vertino**:

Lekt. Dalia Čalnerytė

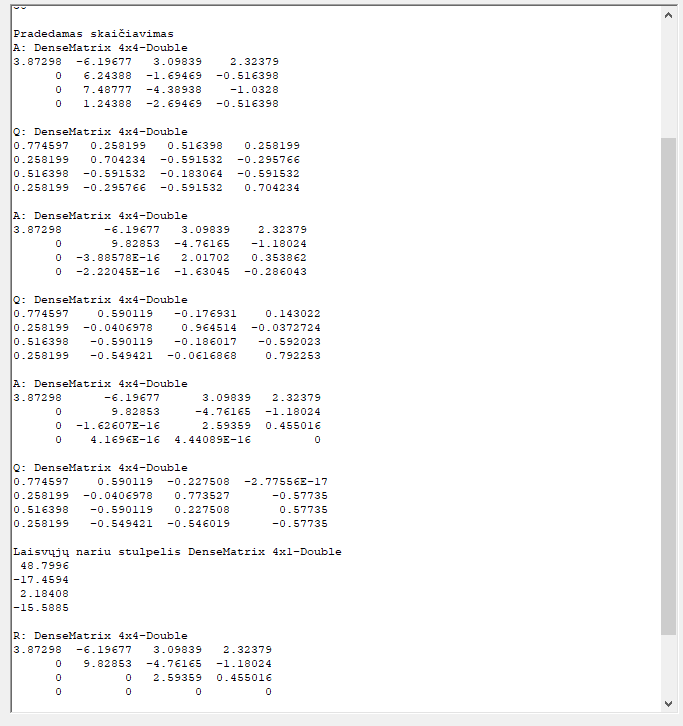
**Užduotys**

Pav. #1 uzduociu sarasas

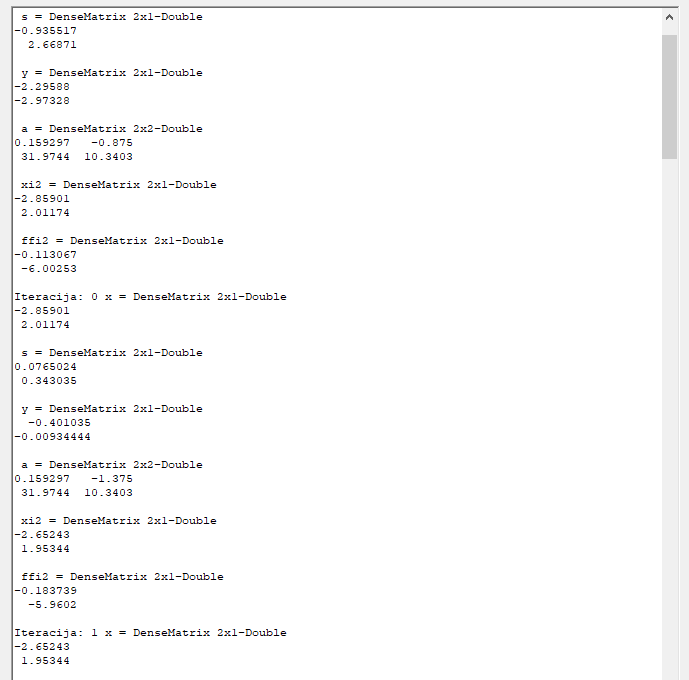
Pav. #2 uzduociu variantu sarasas

**Užduočių sprendimai:**

Spręstas variantas: 30 (užduočių funkcijos bei metodai kuriais jas spręsti pažymėtos geltonai)



Pav. #3 Lygčių sistema QR skaidos metodu



Pav. #4 Netiesinių lygčių sistemos sprendimas Broideno metodu

**Programos kodo fragmentai:**

// ---------------------------- QR skaida --------------------------------------

public void QR()

{

ClearForm();

//Pradine x matrica

Matrix<double> A = DenseMatrix.OfArray(new double[,]

{

{ 3, 1, -1, 1 },

{ 1, -2, 3, 1 },

{ 2, -9, 5, 2 },

{ 1, -7, 2, 1 }

});

//po lygybes matrica

Matrix<double> B = DenseMatrix.OfArray(new double[,]

{

{27},

{24},

{27},

{30}

});

richTextBox1.AppendText("Pradinės matricos A ir B: \n");

richTextBox1.AppendText("A : " + A + "\n");

richTextBox1.AppendText("B : " + B + "\n");

Matrix<double> Q = DenseMatrix.OfArray(new double[,]

{

{ 1, 0, 0, 0 },

{ 0, 1, 0, 0 },

{ 0, 0, 1, 0 },

{ 0, 0, 0, 1 }

});

richTextBox1.AppendText("Pradedamas skaičiavimas \n");

for (int i = 0; i < 3; i++)

{

//Paimam i-taji A matricos stulpeli

var tempz = A.Column(i).ToArray();

double[,] temp = new double[tempz.Length - i, 1];

Matrix<double> z;

for (int j = 0; j < temp.Length; j++)

{

temp[j, 0] = tempz[j + i];

}

//paverciam i matrica

z = DenseMatrix.OfArray(temp);

//richTextBox1.AppendText("z: " + z + "\n");

var zp = DenseMatrix.Build.Dense(z.RowCount, 1);

zp[0, 0] = Math.Sign(z[0, 0]) \* norm(z.Column(0).AsArray());

//richTextBox1.AppendText("zp : " + zp + "\n");

var omega = (z - zp);

omega = omega / omega.FrobeniusNorm();

var Qi = DenseMatrix.Build.DenseIdentity(4 - i) - 2 \* omega \* omega.Transpose();

//richTextBox1.AppendText("Qi: " + Qi + "\n");

//pirmas ciklas

if (i == 0)

{

A = Qi \* A;

}

else

{

var tp = DenseMatrix.Build.Dense(A.RowCount - i, A.RowCount);

for (int q = 0; q < 4 - i; q++)

{

for (int w = 0; w < 4; w++)

{

tp[q, w] = A[q + i, w];

}

}

tp = Qi \* tp;

//Atliekami pakeitimai A matricoje

for (int q = i; q < 4; q++)

{

for (int w = 0; w < 4; w++)

{

A[q, w] = tp[q - i, w];

}

}

}

richTextBox1.AppendText("A: " + A + "\n");

//pirmas ciklas

if (i == 0)

{

Q = Q \* Qi;

}

else

{

var tp = DenseMatrix.Build.DenseIdentity(4, 4);

//Atliekami pakeitimai Q matricai

for (int q = i; q < 4; q++)

{

for (int w = i; w < 4; w++)

{

tp[q, w] = Qi[q - i, w - i];

}

}

Q = Q \* tp;

}

richTextBox1.AppendText("Q: " + Q + "\n");

}

//y=Qt\*b

var bl = Q.Transpose() \* B;

var x = DenseMatrix.Build.Dense(A.ColumnCount, 1);

// Rx = y => x=y/R

x[A.ColumnCount - 1, 0] = bl[A.ColumnCount - 1, 0] / A[A.ColumnCount - 1, A.RowCount - 1];

richTextBox1.AppendText("Laisvųjų nariu stulpelis " + bl + "\n");

double tiksl = 0.00000001;

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

if (Math.Abs(A[i, j]) <= tiksl)

{

A[i, j] = 0;

}

}

}

for (int i = 0; i < 4; i++)

{

if (Math.Abs(bl[i, 0]) <= tiksl)

bl[i, 0] = 0;

}

richTextBox1.AppendText("R: " + A + "\n");

double[] ats = new double[4];

double past = 0;

bool yra = true;

for (int i = 3; i >= 0; i--)

{

if (A[i, i] == 0 && B[i, 0] - past > 0)

{

richTextBox1.AppendText("SPRENDINIU NERA \n");

yra = false;

break;

}

else if (A[i, i] == 0 && B[i, 0] - past == 0)

{

richTextBox1.AppendText("LABAI DAUG SPRENDINIU \n");

yra = false;

break;

}

if (i == 3)

{

ats[i] = B[i, 0] / A[i, i];

past += ats[i];

}

else

{

ats[i] = (B[i, 0] - past) / A[i, i];

past += ats[i];

}

}

for (int i = 0; i < ats.Length; i++)

{

richTextBox1.AppendText("ats: " +ats[i] +"\n");

}

if (yra)

foreach (double atsx in ats)

richTextBox1.AppendText(atsx + " \n");

}

//Frobenius norm/Euclidean norm of array

double norm(double[] z)

{

double norm = 0;

for (int i = 0; i < z.Length; i++)

{

norm += Math.Pow(z[i], 2);

}

norm = Math.Sqrt(norm);

return norm;

}

public void BroydenForTwo()

{

ClearForm();

//max iteraciju skaicius

int itmax = 10;

//pozymis rodantis ar sprendinys geras

bool isGood = true;

//pradinis x1 ir x2 artinys

Matrix<double> x0 = DenseMatrix.OfArray(new double[,]

{

{ -2},//x1

{ -1} //x2

});

var xi = x0;

richTextBox1.AppendText("Pradinis artinys : " + x0 + "\n");

//Lygciu sistemos matrica

var ffi = SystemOfEquations(x0);

//Jakobio matrica

var a = (Matrix<double>) Jacobian(x0);

//sprendinio apskaiciavimo tikslumas

var xi1 = xi - a.Inverse() \* ffi;

var ffi1 = SystemOfEquations(xi1);

// kam tas z????

//var z = Math.Abs(ffi1.);

for (int i = 0; i <= itmax; i++)

{

var s = xi1 - xi;

richTextBox1.AppendText(" s = " + s + "\n");

var y = ffi1 - ffi;

richTextBox1.AppendText(" y = " + y + "\n");

a = a + (y - a \* s) \* s.Transpose() \* (s \* s.Transpose()).Inverse();

richTextBox1.AppendText(" a = " + a + "\n");

var xi2 = xi1 - a.Inverse() \* ffi1;

richTextBox1.AppendText(" xi2 = " + xi2 + "\n");

var ffi2 = SystemOfEquations(xi2);

richTextBox1.AppendText(" ffi2 = " + ffi2 + "\n");

xi = xi1;

xi1 = xi2;

ffi = ffi1;

ffi1 = ffi2;

richTextBox1.AppendText("Iteracija: " +i +" x = " + xi1 + "\n");

if (i == itmax)

{

s = xi1;

richTextBox1.AppendText("Tikslumas pasiektas, Paskutinis artinys x = " + s + "\n");

isGood = true;

break;

}

}

}

public DenseMatrix SystemOfEquations(Matrix<double> x)

{

var f = DenseMatrix.Build.Dense(2, 1);

f[0, 0] = (Math.Cos(x[0, 0]) - x[0, 0] - x[1, 0]);

f[1, 0] = (20 \* Math.Pow(Math.E, -1 \* (Math.Pow(x[0, 0], 2) + Math.Pow(x[1, 0], 2)) / 4) + (Math.Pow(x[0, 0], 2) + Math.Pow(x[1, 0], 2)) / 4 - 10);

return f as DenseMatrix;

}

public DenseMatrix Jacobian(Matrix<double> x)

{

var f = DenseMatrix.Build.Dense(2, 2);

f[0, 0] = -1\*Math.Sin(x[0,0])-1;

f[0, 1] = -1;

f[1, 0] = (x[0,0]/2) - 10 \* x[0,0] \* Math.Pow(Math.E, -1 \* (x[0,0] / 4));

f[1, 1] = (x[1,0]/2) - 10 \* x[1,0] \* Math.Pow(Math.E, -1 \* (x[1,0] / 4));

return f as DenseMatrix;

}

private void Button11\_click(object sender, EventArgs e)

{

ClearForm();

PreparareForm(-10, 10, -5, 5);

BroydenForTwo();

}

// --------------------------------------- Broydenas 2 lygtims pabaiga ---------------------------------------------------------------

// --------------------------------------- Broydenas 4 lygtims -----------------------------------------------------------------------

public void BroydenForFour()

{

ClearForm();

//max iteraciju skaicius

int itmax = 10;

//pozymis rodantis ar sprendinys geras

bool isGood = true;

//pradinis x1 ir x2 artinys

Matrix<double> x0 = DenseMatrix.OfArray(new double[,]

{

{ 2},//x1

{ 3},//x2

{ 2},//x3

{ 4} //x2

});

var xi = x0;

richTextBox1.AppendText("Pradinis artinys : " + x0 + "\n");

//Lygciu sistemos matrica

var ffi = SystemOfEquations2(x0);

//Jakobio matrica

var a = (Matrix<double>)Jacobian2(x0);

//sprendinio apskaiciavimo tikslumas

var xi1 = xi - a.Inverse() \* ffi;

var ffi1 = SystemOfEquations2(xi1);

for (int i = 0; i <= itmax; i++)

{

var s = xi1 - xi;

richTextBox1.AppendText(" s = " + s + "\n");

var y = ffi1 - ffi;

richTextBox1.AppendText(" y = " + y + "\n");

a = a + (y - a \* s) \* s.Transpose() \* (s \* s.Transpose()).Inverse();

richTextBox1.AppendText(" a = " + a + "\n");

var xi2 = xi1 - a.Inverse() \* ffi1;

richTextBox1.AppendText(" xi2 = " + xi2 + "\n");

var ffi2 = SystemOfEquations(xi2);

richTextBox1.AppendText(" ffi2 = " + ffi2 + "\n");

xi = xi1;

xi1 = xi2;

ffi = ffi1;

ffi1 = ffi2;

richTextBox1.AppendText("Iteracija: " + i + " x = " + xi1 + "\n");

if (i == itmax)

{

s = xi1;

richTextBox1.AppendText("Tikslumas pasiektas, Paskutinis artinys x = " + s + "\n");

isGood = true;

break;

}

}

}

public DenseMatrix SystemOfEquations2(Matrix<double> x)

{

var x1 = x[0, 0];

var x2 = x[1, 0];

var x3 = x[2, 0];

var x4 = x[3, 0];

var f = DenseMatrix.Build.Dense(4, 1);

f[0, 0] = 2 \* x1 + 2 \* x2 - x3 + 1;

f[1, 0] = -5 \* Math.Pow(x4, 2) + 4 \* x3 \* x4 - 4;

f[2, 0] = -3 \* Math.Pow(x3, 2) + Math.Pow(x4, 3) - 2 \* x1 \* x4 + 3;

f[3, 0] = 3 \* x1 - 6 \* x2 + 2 \* x3 - 4 \* x4 + 44;

return f as DenseMatrix;

}

public DenseMatrix Jacobian2(Matrix<double> x)

{

var x1 = x[0, 0];

var x2 = x[1, 0];

var x3 = x[2, 0];

var x4 = x[3, 0];

Matrix<double> Jacobian = DenseMatrix.OfArray(new double[,]

{

{ 2 , 2 , -1 , 0 },

{ 0 , 0, 4 \* x4, -10 \* x4 + 4 \* x3 },

{ -2 \* x4 , 0, -6 \* x3, 12\*Math.Pow(x4, 2)},

{ 3, -6, 2, -4 }

});

return Jacobian as DenseMatrix;

}

private void Button12\_click(object sender, EventArgs e)

{

ClearForm();

PreparareForm(-10, 10, -5, 5);

BroydenForFour();

}

// --------------------------------------- Broydenas 4 lygtims pabaiga ---------------------------------------------------------------