



**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

### Lygčių sistemų sprendimas.

#### 1 Tiesinių lygčių sistemų sprendimas

Duota tiesinių lygčių sistema  $[A][X] = [B]$  ir jos sprendimui nurodytas metodas (1 lentelė).

1. Išspręskite tiesinių lygčių sistemą. Jeigu sprendinių be galo daug, raskite bent vieną iš jų. Jeigu sprendinių nėra, pagrįskite, kodėl taip yra.  
*Jei metodas paremtas matricos pertvarkymu, pateikite matricų išraiškas kiekviename žingsnyje. Jei metodas iteracinis, grafiškai pavaizduokite, kaip atliekant iteracijas kinta santykinis sprendinio tikslumas esant kelioms skirtingoms konvergavimo daugiklio reikšmėms.*
2. Patikrinkite gautus sprendinius ir skaidas, įrašydami juos į pradinę lygčių sistemą.
3. Gautą sprendinį patikrinkite naudodami išorinius išteklius (pvz., standartines MATLAB funkcijas).

#### 2 Netiesinių lygčių sistemų sprendimas

1. Duota netiesinių lygčių sistema (2 lentelė. I lygčių sistema):

$$\begin{cases} Z_1(x_1, x_2) = 0 \\ Z_2(x_1, x_2) = 0 \end{cases}$$

- a. Skirtinguose grafikuose pavaizduokite paviršius  $Z_1(x_1, x_2)$  ir  $Z_2(x_1, x_2)$ .
  - b. Užduotyje pateiktą netiesinių lygčių sistemą išspręskite grafiniu būdu.
  - c. Užduotyje pateiktą netiesinių lygčių sistemą išspręskite naudodami užduotyje nurodytą metodą su laisvai pasirinktu pradiniu artiniu (išbandykite bent keturis pradinius artinius). Nurodykite iteracijų pabaigos sąlygas. Lentelėje pateikite pradinį artinį, tikslumą, iteracijų skaičių.
  - d. Gautus sprendinius patikrinkite naudodami išorinius išteklius (pvz., standartines MATLAB funkcijas).
2. Duota netiesinių lygčių sistema (2 lentelė. II lygčių sistema):

$$\begin{cases} Z_1(x_1, x_2, x_3, x_4) = 0 \\ Z_2(x_1, x_2, x_3, x_4) = 0 \\ Z_3(x_1, x_2, x_3, x_4) = 0 \\ Z_4(x_1, x_2, x_3, x_4) = 0 \end{cases}$$

- a. Užduotyje nurodytu metodu išspręskite netiesinių lygčių sistemą su laisvai pasirinktu pradiniu artiniu.
- b. Gautą sprendinį patikrinkite naudodami išorinius išteklius (pvz., standartines MATLAB funkcijas).

#### 3 Optimizavimas

Pagal pateiktą uždavinio sąlygą (3 lentelė) sudarykite tikslo funkciją ir išspręskite ją vienu iš gradientinių metodų (gradientiniu, greičiausio nusileidimo, kvazi-gradientiniu, ar pan.). Gautą taškų konfigūraciją pavaizduokite programoje, skirtingais ženklais pavaizduokite duotus ir pridėtus (jei sąlygoje tokių yra) taškus. Ataskaitoje pateikite pradinę ir gautą taškų konfigūracijas, taikytos tikslo funkcijos aprašymą, taikyto metodo pavadinimą ir parametrus, iteracijų skaičių, iteracijų pabaigos sąlygas ir tikslo funkcijos priklausomybės nuo iteracijų skaičiaus grafiką.

Pav. #1 uždaviniu sarasas

Nr.	Lygčių sistema	Metodas	Nr.	Lygčių sistema	Metodas
25	$\begin{cases} x_1 + 2x_2 + x_3 = -9 \\ 2x_1 + 5x_2 + 4x_4 = -5 \\ 14x_1 - 8x_2 + 4x_3 + x_4 = -1 \\ 5x_1 + 15x_2 + 3x_4 = -8 \end{cases}$	Paprastųjų iteracijų	26	$\begin{cases} 4x_1 + 12x_2 + x_3 + 7x_4 = 171 \\ 2x_1 + 6x_2 + 17x_3 + 2x_4 = 75 \\ 2x_1 + x_2 + 5x_3 + x_4 = 30 \\ 5x_1 + 11x_2 + 7x_3 = 50 \end{cases}$	Gauso
27	$\begin{cases} x_1 + 2x_2 + x_3 + x_4 = 5 \\ 2x_1 - 5x_2 + 2x_4 = -5 \\ 9x_1 - 6x_2 - 6x_3 + x_4 = 39 \\ 5x_1 + 2x_2 + x_4 = 19 \end{cases}$	Gauso – Zeidelio	28	$\begin{cases} 4x_1 + x_2 + x_3 + 7x_4 = 148 \\ x_1 + 2x_3 - 2x_4 = -37 \\ 2x_1 + 2x_2 - 7x_3 + x_4 = 21 \\ 4x_1 + 14x_2 + 7x_3 = 53 \end{cases}$	Gauso – Žordano
29	$\begin{cases} 2x_1 + x_2 + x_3 + x_4 = 6 \\ x_1 + 3x_2 + x_3 - 3x_4 = -4 \\ x_1 + x_2 + 5x_3 + x_4 = 4 \\ 2x_1 + 3x_2 - 3x_3 - 2x_4 = 0 \end{cases}$	QR skaidos	30	$\begin{cases} 3x_1 + x_2 - x_3 + x_4 = 27 \\ x_1 - 2x_2 + 3x_3 + x_4 = 24 \\ 2x_1 - 9x_2 + 5x_3 + 2x_4 = 27 \\ x_1 - 7x_2 + 2x_3 + x_4 = 3 \end{cases}$	QR skaidos

Nr.	I lygčių sistema	II lygčių sistema	Metodas
24	$\begin{cases} \left(\frac{x_1}{8}\right)^8 + \left(\frac{x_2}{8}\right)^8 - 1 = 0 \\ x_1^2 x_2^2 - 16 = 0 \end{cases}$	$\begin{cases} 3x_1 + 4x_2 - 3x_3 + 4x_4 + 25 = 0 \\ 5x_3 + 4x_2x_3 + 55 = 0 \\ 5x_4^3 - 2x_4^2 + x_2x_3 - 97 = 0 \\ 2x_1 - 12x_2 + 4x_3 + 4x_4 - 76 = 0 \end{cases}$	Greičiausio nusileidimo
25	$\begin{cases} \frac{x_1^2 + x_2^2}{2} - 4\cos(x_1) - 4\cos(x_2) - 20 = 0 \\ \frac{20}{x_1^2 + 1} + x_2 = 0 \end{cases}$	$\begin{cases} 5x_1 + x_2 + 4x_4 + 24 = 0 \\ 4x_1^2 + 4x_2x_4 - 32 = 0 \\ 5x_3^3 - 3x_4x_2 - 2x_3^2 - 312 = 0 \\ 5x_1 - 3x_2 + 3x_3 + x_4 + 46 = 0 \end{cases}$	Greičiausio nusileidimo
26	$\begin{cases} 0.1x_1^3 - 0.3x_1x_2^2 = 0 \\ x_1^2 + x_2^2 + 5\cos(x_1) - 16 = 0 \end{cases}$	$\begin{cases} 2x_1 + 2x_2 - 3x_3 - 32 = 0 \\ x_1x_2 - 2x_4 - 12 = 0 \\ -4x_2^2 + x_2x_3 + 3x_3^3 + 676 = 0 \\ 5x_1 - 6x_2 + x_3 + 3x_4 - 4 = 0 \end{cases}$	Greičiausio nusileidimo
27	$\begin{cases} 5\sin(x_1) + \frac{x_1}{2} + x_2 = 0 \\ x_1^2 + 3x_2^2 + 4x_1\cos(x_2) - 20 = 0 \end{cases}$	$\begin{cases} 5x_1 + 3x_2 + x_4 - 45 = 0 \\ 4x_2 + 2x_1x_4 - 84 = 0 \\ 3x_1^3 - 2x_2x_1 - 2x_3^2 - 610 = 0 \\ x_1 - 9x_2 + x_3 + x_4 + 14 = 0 \end{cases}$	Niutono
28	$\begin{cases} \sin(x_1)\cos(x_2) + \frac{x_2}{4} - 0.5 = 0 \\ e^{-3x_1^2 - x_2^2 + 3} - 0.1 = 0 \end{cases}$	$\begin{cases} 4x_2 + 3x_3 + 3x_4 - 26 = 0 \\ 3x_2 + 4x_2x_3 - 75 = 0 \\ x_3^3 - 2x_4^2 - 25 = 0 \\ 5x_1 - 12x_2 + 40 = 0 \end{cases}$	Broideno
29	$\begin{cases} 8\cos(x_1) + x_2^2 = 0 \\ 50e^{-\frac{x_1^2}{4} + x_2^2} + x_1 + x_2 - 5.5 = 0 \end{cases}$	$\begin{cases} x_2 - 2x_3 + 4x_4 + 5 = 0 \\ 4x_4^3 + 2x_2x_4 + 550 = 0 \\ 4x_3^3 - 2x_3^2 - 3x_1x_2 + 550 = 0 \\ 4x_1 - 3x_2 + 2x_3 + 2x_4 + 35 = 0 \end{cases}$	Greičiausio nusileidimo
30	$\begin{cases} \cos(x_1) - x_1 - x_2 = 0 \\ 20e^{-\frac{(x_1^2 + x_2^2)}{4}} + \frac{x_1^2 + x_2^2}{4} - 10 = 0 \end{cases}$	$\begin{cases} 2x_1 + 2x_2 - x_3 + 1 = 0 \\ -5x_4^2 + 4x_3x_4 - 4 = 0 \\ -3x_3^2 + x_4^3 - 2x_1x_4 + 3 = 0 \\ 3x_1 - 6x_2 + 2x_3 - 4x_4 + 44 = 0 \end{cases}$	Broideno

Pav. #2 uždauciu variantu sarasas

## Užduočių sprendimai:

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

```
Pradedamas skaičiavimas
A: DenseMatrix 4x4-Double
3.87298 -6.19677 3.09839 2.32379
0 6.24388 -1.69469 -0.516398
0 7.48777 -4.38938 -1.0328
0 1.24388 -2.69469 -0.516398

Q: DenseMatrix 4x4-Double
0.774597 0.258199 0.516398 0.258199
0.258199 0.704234 -0.591532 -0.295766
0.516398 -0.591532 -0.183064 -0.591532
0.258199 -0.295766 -0.591532 0.704234

A: DenseMatrix 4x4-Double
3.87298 -6.19677 3.09839 2.32379
0 9.82853 -4.76165 -1.18024
0 -3.88578E-16 2.01702 0.353862
0 -2.22045E-16 -1.63045 -0.286043

Q: DenseMatrix 4x4-Double
0.774597 0.590119 -0.176931 0.143022
0.258199 -0.0406978 0.964514 -0.0372724
0.516398 -0.590119 -0.186017 -0.592023
0.258199 -0.549421 -0.0616868 0.792253

A: DenseMatrix 4x4-Double
3.87298 -6.19677 3.09839 2.32379
0 9.82853 -4.76165 -1.18024
0 -1.62607E-16 2.59359 0.455016
0 4.1696E-16 4.44089E-16 0

Q: DenseMatrix 4x4-Double
0.774597 0.590119 -0.227508 -2.77556E-17
0.258199 -0.0406978 0.773527 -0.57735
0.516398 -0.590119 0.227508 0.57735
0.258199 -0.549421 -0.546019 -0.57735

Laisvųjų narių stulpelis DenseMatrix 4x1-Double
48.7996
-17.4594
2.18408
-15.5885

R: DenseMatrix 4x4-Double
3.87298 -6.19677 3.09839 2.32379
0 9.82853 -4.76165 -1.18024
0 0 2.59359 0.455016
0 0 0 0
```

Pav. #3 Lygčių sistema QR skaidos metodu

```

s = DenseMatrix 2x1-Double
-0.935517
 2.66871

y = DenseMatrix 2x1-Double
-2.29588
-2.97328

a = DenseMatrix 2x2-Double
0.159297  -0.875
31.9744  10.3403

xi2 = DenseMatrix 2x1-Double
-2.85901
 2.01174

ffi2 = DenseMatrix 2x1-Double
-0.113067
-6.00253

Iteracija: 0 x = DenseMatrix 2x1-Double
-2.85901
 2.01174

s = DenseMatrix 2x1-Double
0.0765024
0.343035

y = DenseMatrix 2x1-Double
-0.401035
-0.00934444

a = DenseMatrix 2x2-Double
0.159297  -1.375
31.9744  10.3403

xi2 = DenseMatrix 2x1-Double
-2.65243
 1.95344

ffi2 = DenseMatrix 2x1-Double
-0.183739
-5.9602

Iteracija: 1 x = DenseMatrix 2x1-Double
-2.65243
 1.95344

```

Pav. #4 Netiesinių lygčių sistemos sprendimas Broideno metodu

## Programos kodo fragmentai:

```

// ----- QR skaida -----
---
public void QR()
{
    ClearForm();
    //Pradinė 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-tąją A matricos stulpelį
        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];
        }
        //paverčiam 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).ToArray());
        //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 narių 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]) <= tiks1)
        {
            A[i, j] = 0;
        }
    }
}
for (int i = 0; i < 4; i++)
{
    if (Math.Abs(bl[i, 0]) <= tiks1)
        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 sistemas 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 -----
-----
```