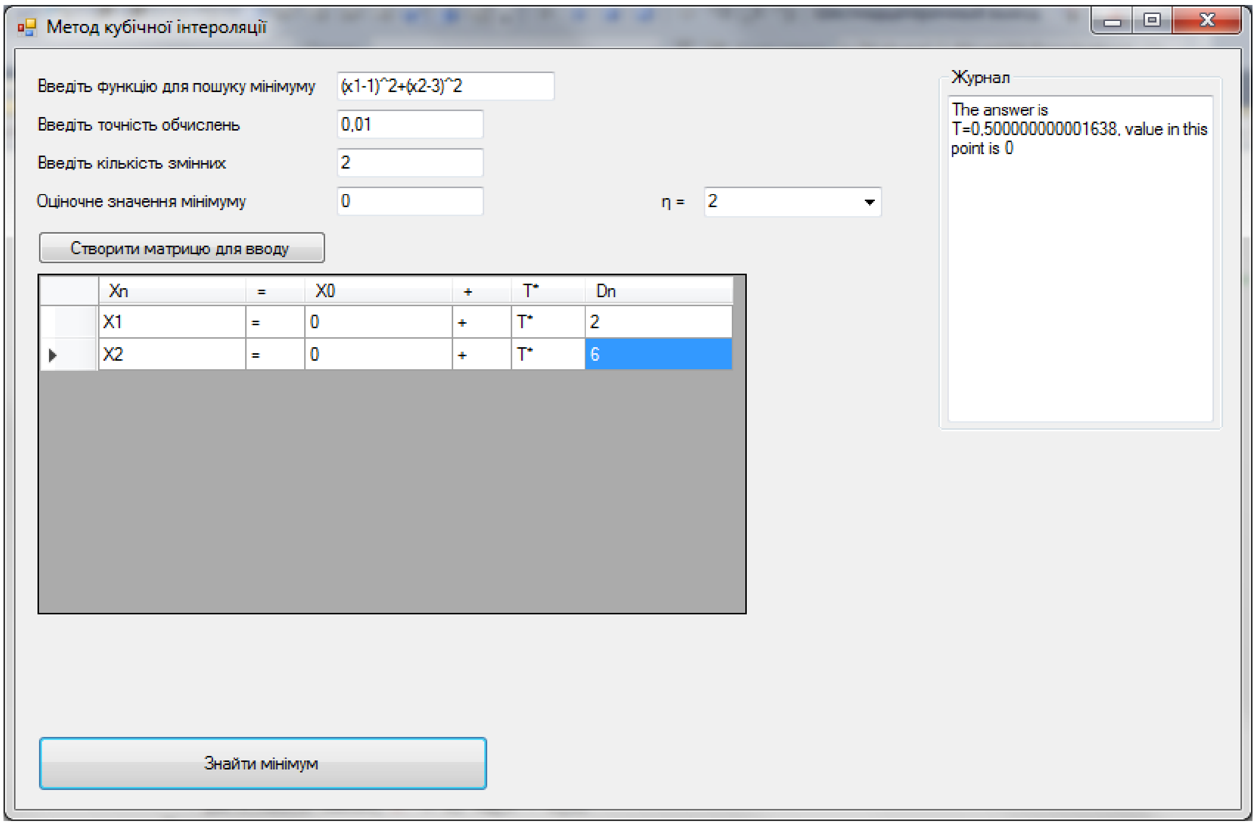


Введіть функцію для пошуку мінімуму (x1-1)^2+(x2-3)^2

Введіть точність обчислень 0,01

Введіть кількість змінних 2

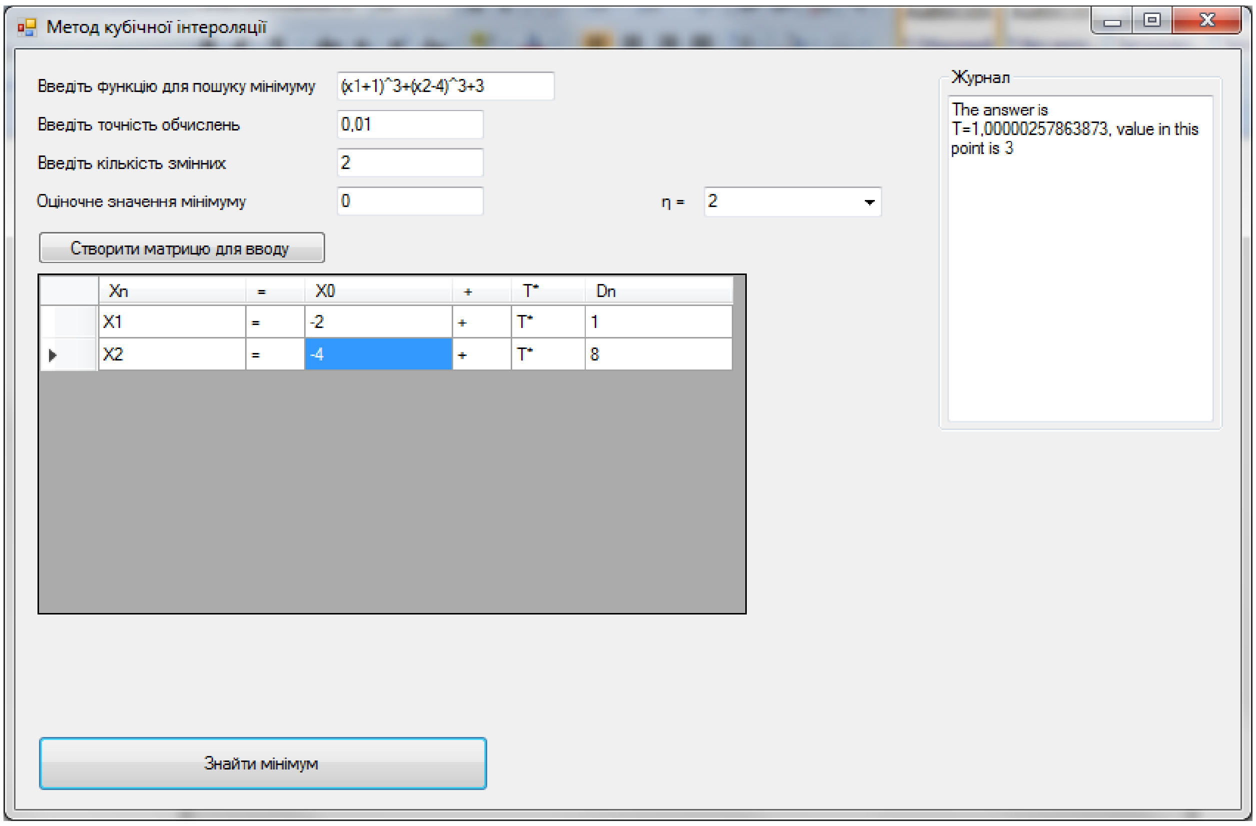
Оціночне значення мінімуму 0

η = 2

Xn	=	X0	+	T*	Dn
X1	=	0	+	T*	2
X2	=	0	+	T*	6

Журнал

The answer is T=0,500000000001638, value in this point is 0



Введіть функцію для пошуку мінімуму (x1+1)^3+(x2-4)^3+3

Введіть точність обчислень 0,01

Введіть кількість змінних 2

Оціночне значення мінімуму 0

η = 2

Xn	=	X0	+	T*	Dn
X1	=	-2	+	T*	1
X2	=	-4	+	T*	8

Журнал

The answer is T=1,00000257863873, value in this point is 3

Код програми:

```
if (IncomeCreated = false)
            {
                textBoxLog.Text += "Start from creating matrix"+Environment.NewLine;
                return;
            }
            int i, j, k;
            string function;
            Parser pars = new Parser();
            function = textBox1.Text;
            double epsilon, QM, ita;
            try
            {
                epsilon = Convert.ToDouble(textBox3.Text);
                if (epsilon <= 0)
                {
                    throw new Exception();
                }
            }
            catch
            {
                textBoxLog.Text += "Wrong epsilon" + Environment.NewLine;
                return;
            }
            try
            {
                QM = Convert.ToDouble(textBox4.Text);
            }
            catch
            {
                textBoxLog.Text += "Wrong QM" + Environment.NewLine;
                return;
            }
            try
            {
                ita = Convert.ToDouble(comboBox1.Text);
                if (ita != 1 && ita != 2)
                {
                    throw new Exception();
                }
            }
            catch
            {
                textBoxLog.Text += "Wrong ita" + Environment.NewLine;
                return;
            }
            double[] X0 = new double[dataGridView1.Rows.Count];
            double[] D = new double[dataGridView1.Rows.Count];
            try
            {
                for (i = 0; i < dataGridView1.Rows.Count; i++)
                {
                    X0[i] = Convert.ToDouble(dataGridView1.Rows[i].Cells[2].Value);
                    D[i] = Convert.ToDouble(dataGridView1.Rows[i].Cells[5].Value);
                }
            }
            catch
            {
                textBoxLog.Text += "Wrong input" + Environment.NewLine;
                return;
            }
            ////
            try
            {
                pars.AddVariable("e", Math.E);
                pars.AddVariable("pi", Math.PI);
                for (i = 1; i <= dataGridView1.Rows.Count; i++)
                {
                    pars.AddVariable("x" + i, 0);
                }
                pars.SimplifyDouble(function);
                for (i = 1; i <= dataGridView1.Rows.Count; i++)
                {
                    pars.RemoveVariable("x" + i);
                }
            }
            catch
            {
                textBoxLog.Text += "Функція введена невірно. Можливо, ви ввели розривну
функцію" + Environment.NewLine;
                return;
            }
            double Multiplyer = 1;
            double[] Grad;
            double[] XQ = new double[dataGridView1.Rows.Count], XP = new
double[dataGridView1.Rows.Count], XR = new double[dataGridView1.Rows.Count];
            double p, q, Gp, Gq, fp, fq;
            p = 0;
            for (i = 1; i <= dataGridView1.Rows.Count; i++)
            {
                XP[i - 1] = X0[i - 1];
                pars.AddVariable("x" + i, XP[i - 1]);
            }
            fp = pars.SimplifyDouble(function);
            for (i = 1; i <= dataGridView1.Rows.Count; i++)
            {
                pars.RemoveVariable("x" + i);
            }
            Grad = FindGrad(pars, function, X0);
            Gp = Scalar(Grad,D);
            if (Gp > 0)
            {
                Multiplyer = -1;
            }
            q = p + (Math.Min(ita,(-2*(fp-QM))/Gp));
            // double
            for (i = 1; i <= dataGridView1.Rows.Count; i++)
            {
                XQ[i-1] = X0[i - 1]+q*D[i-1];
                pars.AddVariable("x" + i, XQ[i-1]);
            }
            fq = pars.SimplifyDouble(function);
            for (i = 1; i <= dataGridView1.Rows.Count; i++)
            {
                pars.RemoveVariable("x" + i);
            }
            Grad = FindGrad(pars, function, XQ);
            Gq = Scalar(Grad, D);
            while ((fq <= fp) && (Multiplyer * Gq < 0.000001))
            {
                q *= 2;
                for (i = 1; i <= dataGridView1.Rows.Count; i++)
                {
                    XQ[i-1] = X0[i - 1]+q*D[i-1];
                    pars.AddVariable("x" + i, XQ[i-1]);
                }
                fq = pars.SimplifyDouble(function);
                for (i = 1; i <= dataGridView1.Rows.Count; i++)
                {
                    pars.RemoveVariable("x" + i);
                }
                Grad = FindGrad(pars, function, XQ);
                Gq = Scalar(Grad, D);
            }
            /////
            double z, w;
            z = ((3*(fp-fq))/q)+Gp+Gq;
            w = Math.Sqrt(z * z - Gp * Gq);
            double r, fr, Gr;
            r = (q * (z + w - Gp)) / (Gq - Gp + 2 * w);
            //r = (q * (((3 * (fp - fq)) / q) + Math.Sqrt(Math.Pow(((3 * (fp - fq)) / q),
2) - Gp * Gq)-Gp)) / (Gq - Gp + 2 * (Math.Sqrt(Math.Pow(((3 * (fp - fq)) / q), 2) - Gp *
Gq)));
            for (i = 1; i <= dataGridView1.Rows.Count; i++)
            {
                XR[i - 1] = XP[i - 1] + r * D[i - 1];
                pars.AddVariable("x" + i, XR[i - 1]);
            }
            fr = pars.SimplifyDouble(function);
            for (i = 1; i <= dataGridView1.Rows.Count; i++)
            {
                pars.RemoveVariable("x" + i);
            }
            Grad = FindGrad(pars, function, XR);
            Gr = Scalar(Grad, D);
            while (Math.Abs(Gr) > epsilon)
            {
                if (Gr > 0)
                {
                    q = r;
                    // double
                    for (i = 1; i <= dataGridView1.Rows.Count; i++)
                    {
                        XQ[i - 1] = XR[i - 1];
                        pars.AddVariable("x" + i, XQ[i - 1]);
                    }
                    fq = pars.SimplifyDouble(function);
                    for (i = 1; i <= dataGridView1.Rows.Count; i++)
                    {
                        pars.RemoveVariable("x" + i);
                    }
                    Grad = FindGrad(pars, function, XQ);
                    Gq = Scalar(Grad, D);
                }
                else
                {
                    p = r;
                    for (i = 1; i <= dataGridView1.Rows.Count; i++)
                    {
                        XP[i - 1] = XR[i - 1];
                        pars.AddVariable("x" + i, XP[i - 1]);
                    }
                    fp = pars.SimplifyDouble(function);
                    for (i = 1; i <= dataGridView1.Rows.Count; i++)
                    {
                        pars.RemoveVariable("x" + i);
                    }
                    Grad = FindGrad(pars, function, X0);
                    Gp = Scalar(Grad, D);
                }
                z = (3 * (fp - fq)) / q + Gp + Gq;
                w = Math.Sqrt(z * z - Gp * Gq);
                r = (q * (z + w - Gp)) / (Gq - Gp + 2 * w);
                //r = (q * (((3 * (fp - fq)) / q) + Math.Sqrt(Math.Pow(((3 * (fp - fq)) /
q), 2) - Gp * Gq) - Gp)) / (Gq - Gp + 2 * (Math.Sqrt(Math.Pow(((3 * (fp - fq)) / q), 2) -
Gp * Gq)));
                for (i = 1; i <= dataGridView1.Rows.Count; i++)
                {
                    XR[i - 1] = XP[i - 1] + r * D[i - 1];
                    pars.AddVariable("x" + i, XR[i - 1]);
                }
                fr = pars.SimplifyDouble(function);
                for (i = 1; i <= dataGridView1.Rows.Count; i++)
                {
                    pars.RemoveVariable("x" + i);
                }
                Grad = FindGrad(pars, function, XR);
                Gr = Scalar(Grad, D);
            }
            textBoxLog.Text += "The answer is T=" + r  + ", value in this point is " +
Math.Round(fr,6) + Environment.NewLine;
```