

Метод найшвидшого спуску

Введіть тут розмірність простору:

Введіть функцію для пошуку мінімуму:

Введіть точність обчислень:

Введіть початковий крок λ :

Введіть початкову точку пошуку:

	Координата	Значення
	x1	-5
►	x2	-8

Знайти мінімум

Точка мінімуму:

	Координата	Значення
►	x1	0.989167
	x2	0.978403
	Значення -	0.000118

Журнал

Кількість кроків - 4075

Метод найшвидшого спуску

Введіть тут розмірність простору:

Введіть функцію для пошуку мінімуму:

Введіть точність обчислень:

Введіть початковий крок λ :

Введіть початкову точку пошуку:

	Координата	Значення
	x1	5
	x2	-15
	x3	6
►	x4	7

Знайти мінімум

Точка мінімуму:

	Координата	Значення
►	x1	-0.126446
	x2	-0.012673
	x3	-0.062399
	x4	-0.063325
	Значення -	0.000321

Журнал

Кількість кроків - 436

Код програми

```
int i, j, k;
Parser pars = new Parser();
if (textBox2.Enabled == false)
{
    textBox4.Text += "Неможливо почати роботу - для початку введіть усі дані"
+ Environment.NewLine;
    return;
}
double Lambda, LastLambda;
try
{
    Lambda = Convert.ToDouble(textBox6.Text);
}
catch
{
    textBox4.Text += "Початкове значення  $\lambda$  введено невірно" +
Environment.NewLine;
    return;
}
if ((Lambda <= 0) || (Lambda >= 1000000))
{
    textBox4.Text += "Початкове значення  $\lambda$  введено невірно. Памятайте, що
воно має бути додатнім" + Environment.NewLine;
    return;
}
string function = textBox2.Text;
try
{
    pars.AddVariable("e", Math.E);
    pars.AddVariable("pi", Math.PI);
    for (i = 1; i <= NumberOfDim; i++)
    {
        pars.AddVariable("x" + i, 0);
    }
    pars.SimplifyDouble(function);
    for (i = 1; i <= NumberOfDim; i++)
    {
        pars.RemoveVariable("x" + i);
    }
}
catch
{
    textBox4.Text += "Функція введена невірно. Можливо, ви ввели розривну
функцію" + Environment.NewLine;
    return;
}
double epsilon;
try
{
    epsilon = Convert.ToDouble(textBox5.Text);
}
catch
{
    textBox4.Text += "Точність введена невірно" + Environment.NewLine;
    return;
}
double CurFunc;
double NextFunc;
double[] Grad;
double TempVal;
```

```

bool Changed = false;
double[] CurPoint = new double[NumberOfDim];
double[] TempPoint = new double[NumberOfDim];
try
{
    for (i = 0; i < NumberOfDim; i++)
    {
        CurPoint[i] = Convert.ToDouble(dataGridView1.Rows[i].Cells[1].Value);
        pars.AddVariable("x" + (i + 1), CurPoint[i]);
    }
    CurFunc = pars.SimplifyDouble(function);
    for (i = 1; i <= NumberOfDim; i++)
    {
        pars.RemoveVariable("x" + i);
    }
}
catch
{
    textBox4.Text += "Початкова точка введена невірно" + Environment.NewLine;
    return;
}
int Num = 0;
Grad = FindGrad(pars, function, CurPoint);
/////
Random a = new Random();
double TempMaxLam = (LastLambda = Lambda);
double TempLambda;
while (FindLeng(Grad) > epsilon)
{
    // Lambda *= 0.99;
    LastLambda = Lambda;
    Num++;
    if (Num > 10000)
    {
        textBox4.Text += "Трапилося зациклення. Спробуйте підкоректувати  
початкове лямбда" + Environment.NewLine;
        return;
    }
    /////
    for (i = 0; i < NumberOfDim; i++)
    {
        TempPoint[i] = CurPoint[i] - LastLambda * Math.Sign(Grad[i]);
    }
    /////
    for (i = 0; i < NumberOfDim; i++)
    {
        pars.AddVariable("x" + (i + 1), TempPoint[i]);
    }
    NextFunc = pars.SimplifyDouble(function);
    for (i = 1; i <= NumberOfDim; i++)
    {
        pars.RemoveVariable("x" + i);
    }
    while (NextFunc >= CurFunc)
    {
        LastLambda /= 2;
        for (i = 0; i < NumberOfDim; i++)
        {
            TempPoint[i] = CurPoint[i] - LastLambda * Math.Sign(Grad[i]);
        }
        /////
        for (i = 0; i < NumberOfDim; i++)
        {
            pars.AddVariable("x" + (i + 1), TempPoint[i]);
        }
    }
}

```

```

        NextFunc = pars.SimplifyDouble(function);
        for (i = 1; i <= NumberOfDim; i++)
        {
            pars.RemoveVariable("x" + i);
        }
    }
    for (i = 0; i < NumberOfDim; i++)
    {
        CurPoint[i] -= LastLambda * Math.Sign(Grad[i]);
    }
    /////
    for (i = 0; i < NumberOfDim; i++)
    {
        pars.AddVariable("x" + (i + 1), CurPoint[i]);
    }
    CurFunc = pars.SimplifyDouble(function);
    for (i = 1; i <= NumberOfDim; i++)
    {
        pars.RemoveVariable("x" + i);
    }
    /////
    //Lambda = 3*TempMaxLam;
    Grad = FindGrad(pars, function, CurPoint);
    Lambda -= (Lambda-LastLambda)/10;
    /////
}
/////
for (i = 0; i < NumberOfDim; i++)
{
    pars.AddVariable("x" + (i + 1), CurPoint[i]);
}
CurFunc = pars.SimplifyDouble(function);
for (i = 1; i <= NumberOfDim; i++)
{
    pars.RemoveVariable("x" + i);
}
for (i = 0; i < NumberOfDim; i++)
{
    dataGridView2.Rows[i].Cells[1].Value = "" + Math.Round(CurPoint[i], 6);
}
dataGridView2.Rows[NumberOfDim].Cells[1].Value = "" + Math.Round(CurFunc, 6);
/////
textBox4.Text += "Кількість кроків - " + Num + Environment.NewLine;

```