МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №4

З ДИСЦИПЛІНИ «Математичне моделювання та чисельні методи»

НА ТЕМУ «Розв'язання задач пошуку екстремуму»

Виконав:

студент гр. ПЗПІ-17-1

Кириченко О. В.

Перевірив:

Матвєєв Д. І.

Харків – 2020

**Мета роботи:** Навчитись використовувати математичне середовище MATLAB для пошуку екстремумів функцій. Відтворити досліджену поведінку математичного середовища за допомогою мови програмування.

Задля локанічності нижче наведено загальні методи на мові C#, які будуть використані у завданнях:

public (double[] mins, double[] maxs) FindExtremumsUniform(

double[] poly, double start, double end, double delta

){

List<double> mins = new List<double>();

List<double> maxs = new List<double>();

double lastX = start;

while(lastX <= end)

{

Extremum extremum = FindExtremumUniformForUnimodal(

poly, lastX, end, delta

);

if (extremum == null)

break;

if (extremum.IsMin)

mins.Add(extremum.X);

else

maxs.Add(extremum.X);

lastX = extremum.X;

}

return (mins.ToArray(), maxs.ToArray());

}

public Extremum FindExtremumUniformForUnimodal(

double[] poly, double start, double end, double delta

){

double[] xs = Range.Make(start, end, delta);

Extremum extremum = new Extremum() { IsMin = true, X = start };

Predicate<int> breakCondition = i =>

Poly.Val(poly, xs[i + 1])[0] >= Poly.Val(poly, xs[i])[0];

if (breakCondition(0))

{

breakCondition = i =>

Poly.Val(poly, xs[i + 1])[0] <= Poly.Val(poly, xs[i])[0];

extremum.IsMin = false;

}

for (int i = 0; i < xs.Length - 1; ++i)

if (breakCondition(i))

{

extremum.X = xs[i] + delta / 2;

return extremum;

}

return null;

}

public class Extremum

{

public double X { get; set; }

public bool IsMin { get; set; }

}

public static class Poly

{

public static double[] PolySolvePrimitive(double[] poly, double eq)

{

if (poly.Length == 2)

return new double[] { (eq - poly[0]) / poly[1] };

throw new ArgumentException();

}

public static double[] GetPolyDerivative(double[] poly)

{

double[] diffPoly = new double[poly.Length - 1];

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

diffPoly[i] = poly[i + 1] \* (i + 1);

return diffPoly;

}

public static Func<double, double> Get(double[] p)

{

return x => p.Select((a, i) => a \* Math.Pow(x, i)).Sum();

}

public static double[] Val(double[] p, params double[] x)

{

Func<double, double> poly = Get(p);

return x.Select(xi => poly(xi)).ToArray();

}

}

public static class Range

{

public static double[] Make(double start, double end, double delta)

{

int count = (int)Math.Ceiling((end - start) / delta);

return Enumerable.Range(0, count).Select(

i => i \* delta + start

).ToArray();

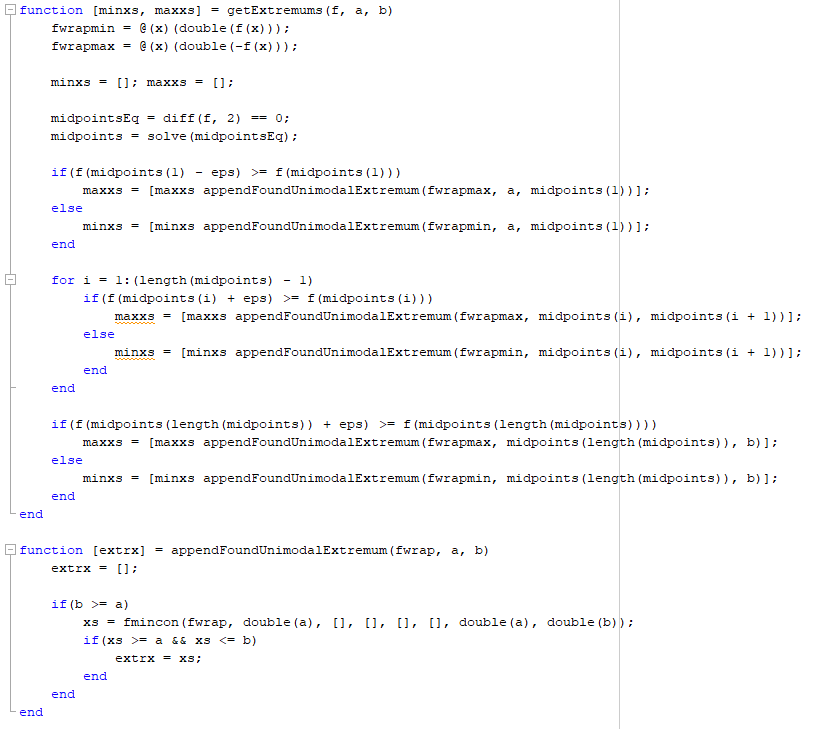
}

}

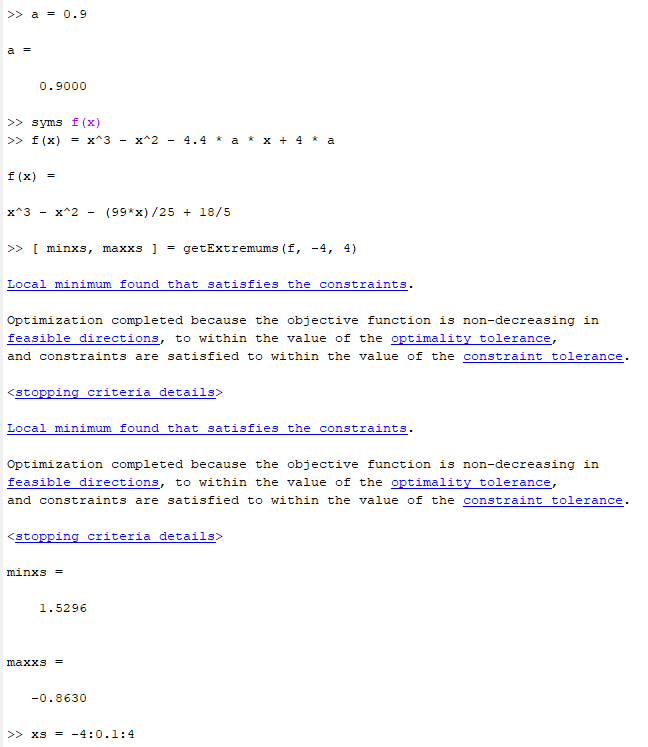
1. Визначити проміжки унімодальності функції . Обчислити координати усіх точок екстремумів за допомогою метода рівномірного пошуку.

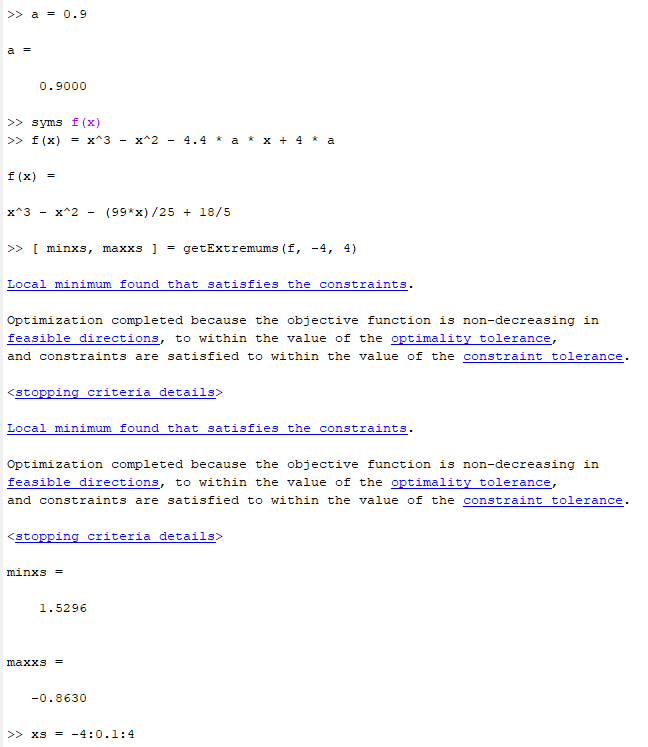
|  |  |
| --- | --- |
| MATLAB | C# |
|  | public Point Task1(  Form form, Point location  ) {  double a = 0.9;  double[] f = new double[] {  4 \* a, -4.4 \* a, -1, 1  };  double[] firstDerivative = Poly.GetPolyDerivative(f);  double[] secondDerivative = Poly.GetPolyDerivative(firstDerivative);  double[] xs = Range.Make(-5, 5, 0.1);  double[] breakXs = Poly.PolySolvePrimitive(  secondDerivative, 0  );  (double[] mins, double[] maxs) = FindExtremumsUniform(f, -4, 4, 0.01);  Console.WriteLine(  "MINS: " + string.Join(" ", mins)  );  Console.WriteLine(  "MAXS: " + string.Join(" ", maxs)  );  Plot plot = new Plot();  plot.PlotSignalXY(  xs, Poly.Val(f, xs), Color.Red, 7,  label: "Original signal"  );  plot.PlotSignalXY(  xs, Poly.Val(firstDerivative, xs),  Color.Green, 5,  label: "First Derivative"  );  plot.PlotSignalXY(  xs, Poly.Val(secondDerivative, xs),  Color.Blue, 3,  label: "Second Derivative"  );  plot.PlotScatter(  breakXs.Concat(breakXs).Concat(  breakXs  ).ToArray(),  Poly.Val(f, breakXs).Concat(  Poly.Val(firstDerivative, breakXs)  ).Concat(  Poly.Val(secondDerivative, breakXs)  ).ToArray(),  Color.Yellow,  label: "Unimodal break points",  markerSize: 3,  lineStyle: LineStyle.Dot  );  plot.PlotScatter(  mins, Poly.Val(f, mins),  Color.Cyan, markerSize: 7,  label: "Min points",  lineStyle: LineStyle.Dot  );  plot.PlotScatter(  maxs, Poly.Val(f, maxs),  Color.LightGreen, markerSize: 7,  label: "Max points",  lineStyle: LineStyle.Dot  );  plot.Legend();  return plot.AddToForm(form, location);  } |

Також задля перевірки у MATLAB було створено скрипт, що знаходить екстремуми за допомогою вбудованої функції пошуку мінімуму fmincon:

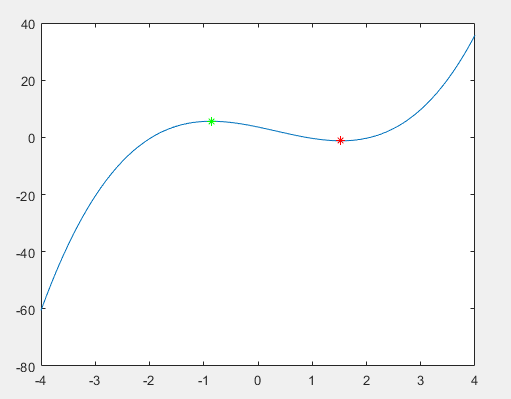


Результат виконання функції, що використовує метод рівномірного пошуку, наближається до результату, отриманого з використанням вбудованої функції fmincon.





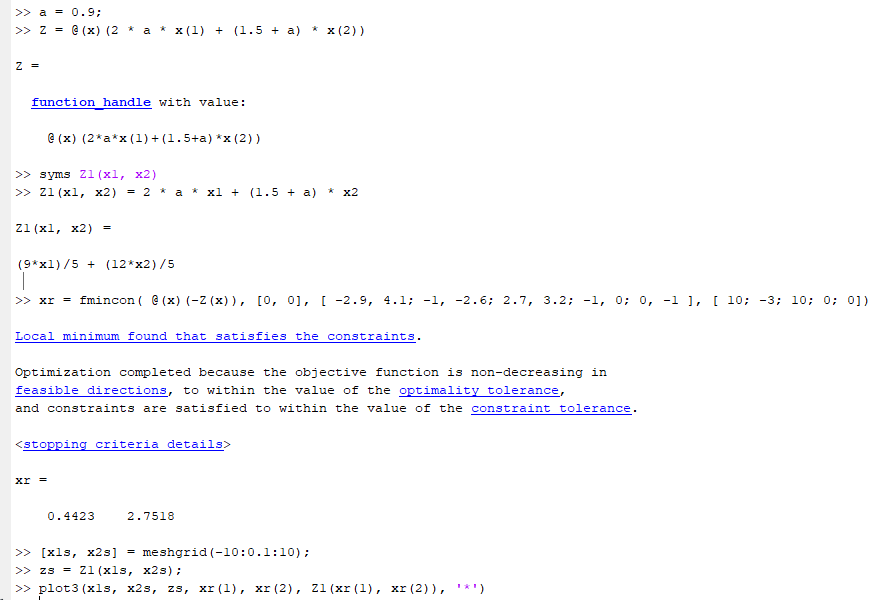


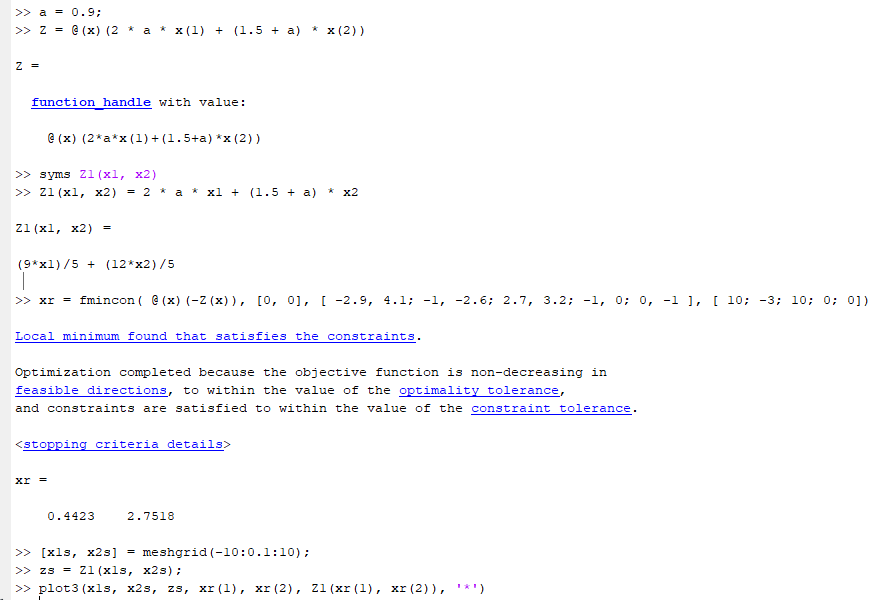


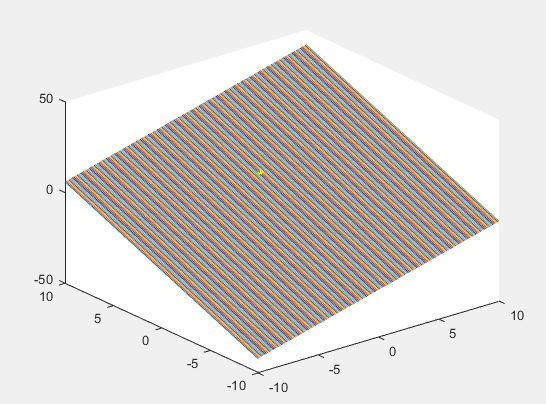
1. Розв'язати задачу лінійного програмування:

за обмежень:

Було записано наступні MATLAB-вирази:



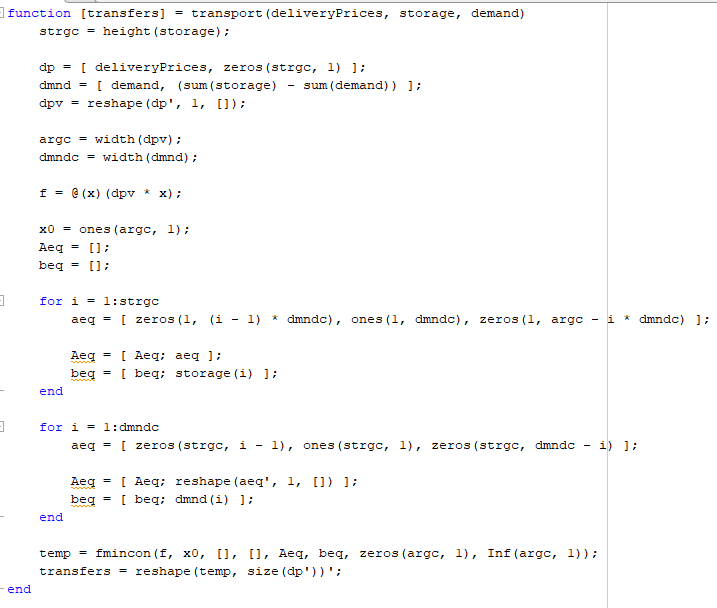




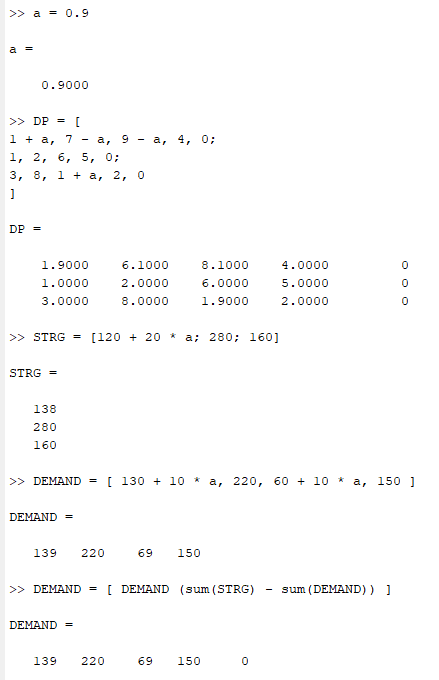
1. Розв'язати наступну транспортну задачу:



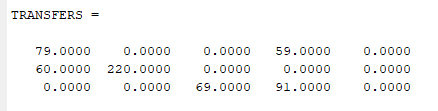
Для розв’язання типової транспортної задачі було створено наступна MATLAB-функція:



Вхідні дані задачі було передано у створену функцію та отримано наступний результат – таблицю перевезень товарів від постачальників до споживачів (останній стовпець результату позначає кількість продукту, що має лишитись на складі):



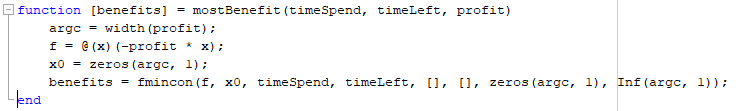




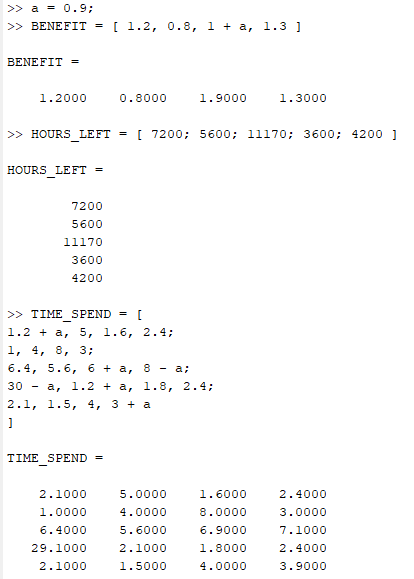
1. Розв’язати наступну задачу оптимізації:

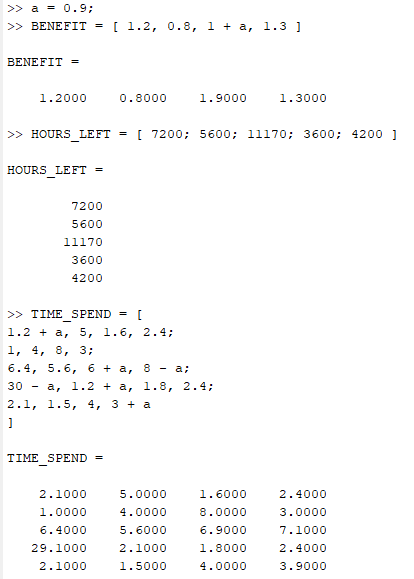


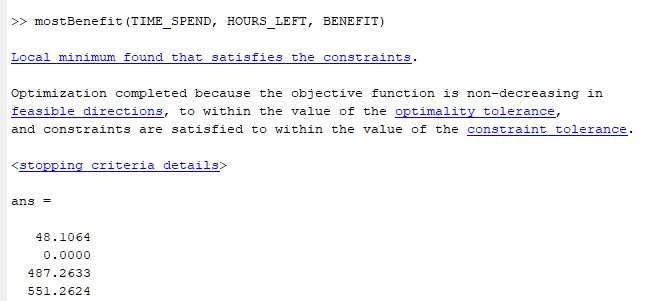
Для розв’язання типової задачі оптимізації було створено наступну MATLAB-функцію:

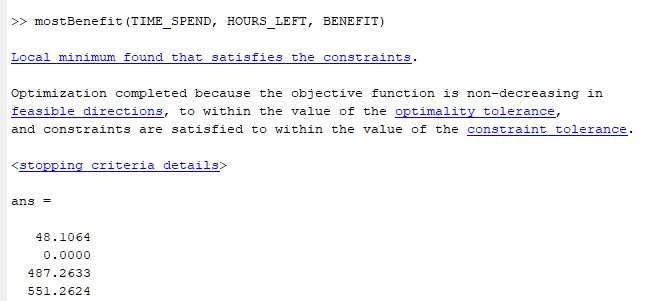


Вхідні дані задачі було передано до створеної функції та отримано наступний результат – кількість кілометрів кожної марки кабелю, яку треба виробити, аби мати найбільший прибуток:









**Висновки**: в ході роботи було здобуто навички використання математичного середовища MATLAB для пошуку екстремумів функцій, розв'язання транспортних задач та задач. Досліджену поведінку математичного середовища для задач пошуку екстремумів функцій було відтворено за допомогою мови програмування C#.