МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №2

З ДИСЦИПЛІНИ «Математичне моделювання та чисельні методи»

НА ТЕМУ «Розв’язання диференційних рівнянь та систем»

Виконав:

студент гр. ПЗПІ-17-1

Кириченко О. В.

Перевірив:

Матвєєв Д. І.

Харків – 2020

**Мета роботи:** Навчитись використовувати математичне середовище MATLAB розв’язання диференційних рівнянь та систем. Відтворити досліджену поведінку математичного середовища за допомогою мови програмування.

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

public static class DkExtensioins

{

public static Plot GetPlot(this DkSolve solve)

{

Plot plot = new Plot();

int plotCount = solve.Ys.Min(ys => ys.Count());

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

plot.PlotSignalXY(

solve.X, solve.Ys.Select(

ys => ys.ElementAt(i)

).ToArray()

);

return plot;

}

public static Point AddToForm(

this Plot plot, Form form, Point location  
 ) {

Bitmap chart = plot.Render();

PictureBox picture = new PictureBox();

picture.Size = chart.Size;

picture.Location = location;

picture.BackgroundImage = chart;

form.Controls.Add(picture);

return new Point(location.X, location.Y + chart.Height);

}

}

public static class EnumerableExtensions

{

public static double[] Multiply(

this IEnumerable<double> collection, double multiplier

) {

return collection.Apply(item => item \* multiplier).ToArray();

}

public static double[] Add(

this IEnumerable<double> collection,

IEnumerable<double> other

) {

return collection.Apply(

other, (item, otherItem) => item + otherItem

).ToArray();

}

public static double[] Add(

this IEnumerable<double> collection, double added

) {

return collection.Apply(item => item + added).ToArray();

}

public static IEnumerable<T> Apply<T>(

this IEnumerable<T> collection,

IEnumerable<T> other, Func<T, T, T> operation

) {

if (collection.Count() != other.Count())

throw new ArgumentException();

T[] result = new T[collection.Count()];

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

result[i] = operation(

collection.ElementAt(i), other.ElementAt(i)

);

return result;

}

public static IEnumerable<T> Apply<T>(

this IEnumerable<T> collection, Func<T, T> operation

) {

T[] result = new T[collection.Count()];

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

result[i] = operation(collection.ElementAt(i));

return result;

}

}

public class DkParameters

{

public List<Func<double[], double>> Equations { get; set; } = new

List<Func<double[], double>>();

public double[] Init { get; set; }

public double Start { get; set; }

public double End { get; set; }

public double Delta { get; set; }

public bool PassX { get; set; }

}

public class DkSolve

{

public double[] X { get; set; }

public double[][] Ys { get; set; }

}

public DkSolve SolveInPointsRK(DkParameters parameters)

{

double[] x = new double[(int)Math.Ceiling(

(parameters.End - parameters.Start) / parameters.Delta

)];

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

x[i] = parameters.Delta \* i + parameters.Start;

double[][] ys = new double[x.Length][];

ys[0] = parameters.Init;

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

double[] prefix = parameters.PassX ? new double[] { x[i] } : new double[] { };

double[] prefixHalf = prefix.Add(0.5 \* parameters.Delta);

double[] prefixFull = prefix.Add(parameters.Delta);

double[] k1 = parameters.Equations.Select(eq =>

eq(prefix.Concat(ys[i]).ToArray())

).ToArray();

double[] k2 = parameters.Equations.Select(

eq => eq(prefixHalf.Concat(

k1.Multiply(0.5 \* parameters.Delta).Add(ys[i])

).ToArray())

).ToArray();

double[] k3 = parameters.Equations.Select(

eq => eq(prefixHalf.Concat(

k2.Multiply(0.5 \* parameters.Delta).Add(ys[i])

).ToArray())

).ToArray();

double[] k4 = parameters.Equations.Select(

eq => eq(prefixFull.Concat(

k3.Multiply(parameters.Delta).Add(ys[i])

).ToArray())

).ToArray();

ys[i + 1] = ys[i].Add(

k1.Add(k2.Multiply(2))

.Add(k3.Multiply(2))

.Add(k4)

.Multiply(parameters.Delta / 6)

);

}

return new DkSolve() { X = x, Ys = ys };

}

1. Обчислити таблицю наближених значень диференційного рівняння

, що задовольняє початковій умові на проміжку [0,1.9]. Крок зміни значень аргументу h = 0.05.

|  |  |
| --- | --- |
| MATLAB | C# |
|  | public double Func1(params double[] data)=>  Math.Cos(data[1]) / (data[0] + 1.5) +  0.1 \* Math.Pow(data[1], 2.0);  DkParameters parameters = new DkParameters()  {  Equations = new List<Func<double[], double>>() { Func1 },  Init = new double[] { 0 },  Start = 0,  End = 2,  Delta = 0.05,  PassX = true  };  SolveInPointsRK(parameters)  .GetPlot().AddToForm(form, location); |

1. Розв’язати систему диференційних рівнянь

на проміжку 0 <= t <= 2 з кроком 0.05 та початковими умовами та :

|  |  |
| --- | --- |
| MATLAB | C# |
|  | public double Func2(params double[] data)  {  return 0.9 \* data[0] - 4 \* data[1];  }  public double Func3(params double[] data)  {  return data[0] + 0.9 \* data[1];  }  DkParameters parameters = new DkParameters()  {  Equations = new List<Func<double[], double>>() { Func2, Func3 },  Init = new double[] { 4, 6 },  Start = 0,  End = 2,  Delta = 0.05,  PassX = false  };  SolveInPointsRK(parameters)  .GetPlot()  .AddToForm(form, location); |

**Висновки**: в ході роботи було здобуто навички використання математичного середовища MATLAB для розв’язання диференційних рівнянь та їх систем. Досліджену поведінку математичного середовища було відтворено за допомогою мови програмування C#.