

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 диференціальних рівнянь та систем»

Виконав:
студент гр. ПЗП-17-1
Кириченко О. В.
Перевірив:
Матвеев 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();
    }
}

```

```

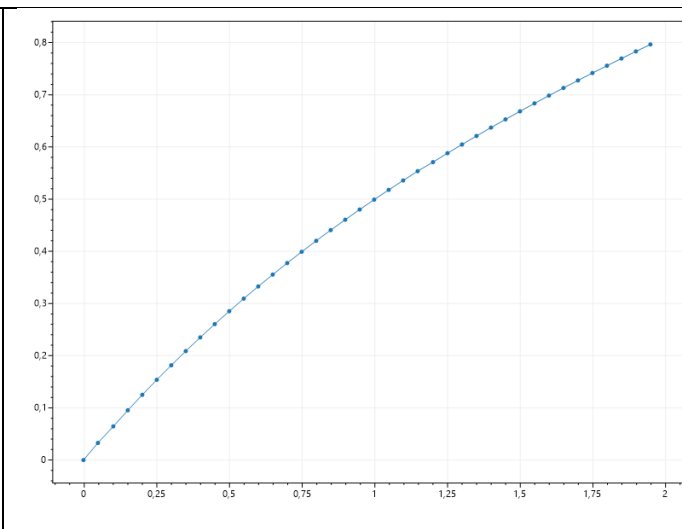
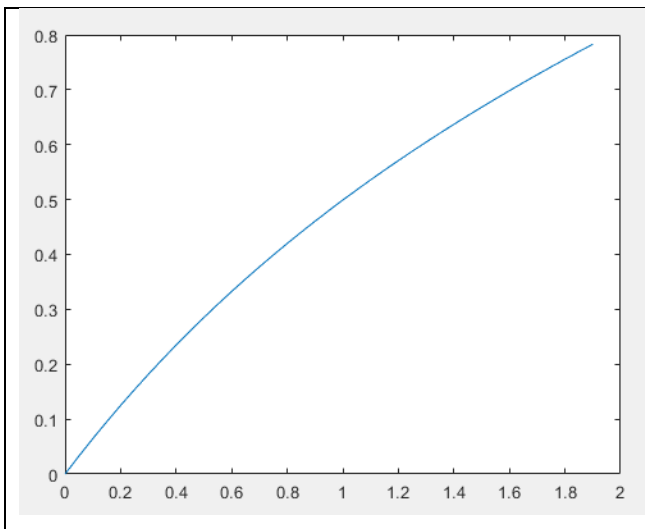
        ys[i + 1] = ys[i].Add(
            k1.Add(k2.Multiply(2))
                .Add(k3.Multiply(2))
                .Add(k4)
                .Multiply(parameters.Delta / 6)
        );
    }
    return new DkSolve() { X = x, Ys = ys };
}

```

1. Обчислити таблицю наближених значень диференційного рівняння

$y' = \frac{\cos(y)}{x+1.5} + 0.1y^2$, що задовольняє початковій умові $y(0) = 0$ на проміжку $[0, 1.9]$. Крок зміни значень аргументу $h = 0.05$.

MATLAB	C#
<pre> function [y] = RK(f, a, x0, y0, h) x = x0:h:a; y = zeros(size(x)); y(1) = y0; for i=1:(length(x)-1) k1 = f(x(i),y(i)); k2 = f(x(i)+0.5*h,y(i)+0.5*h*k1); k3 = f((x(i)+0.5*h), (y(i)+0.5*h*k2)); k4 = f((x(i)+h), (y(i)+k3*h)); y(i+1) = y(i) + (1/6)*(k1+2*k2+2*k3+k4)*h; end end >> syms f(x, y) >> f(x, y) = cos(y) / (x + 1.5) + 0.1 * y^2 f(x, y) = cos(y)/(x + 3/2) + y^2/10 >> RK(f, 1.9, 0, 0, 0.05) >> plot(0:0.05:1.9, ans) </pre>	<pre> 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); </pre>



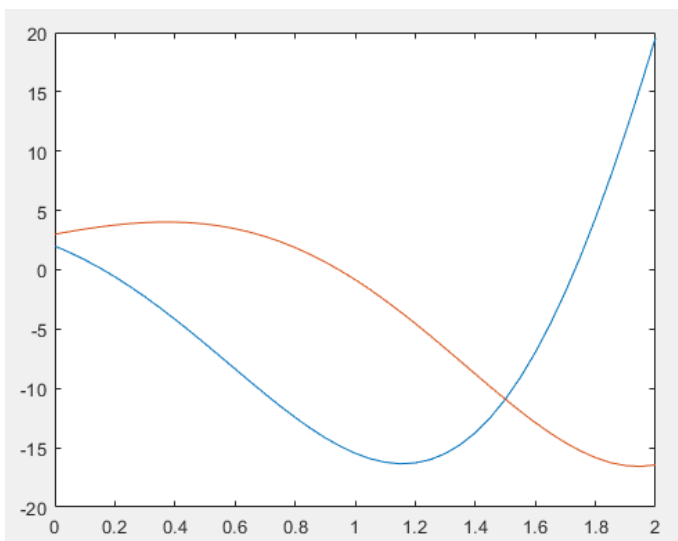
2. Розв'язати систему диференціальних рівнянь

$$\begin{cases} x' = 0.9x - 4y \\ y' = x + 0.9y \end{cases}$$

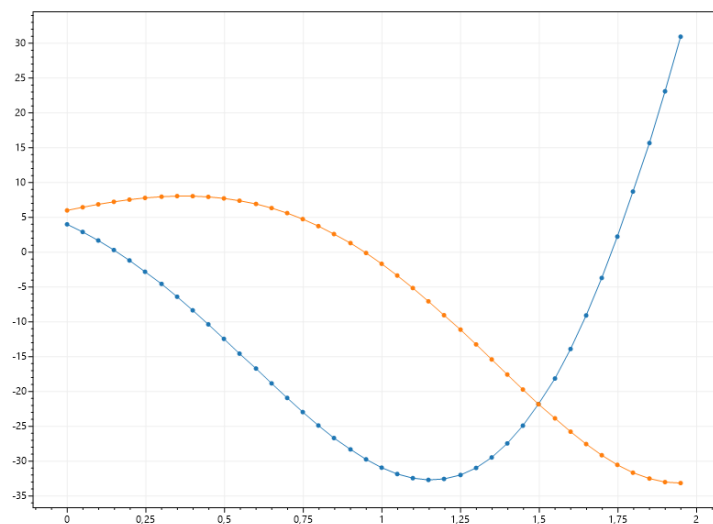
на проміжку $0 \leq t \leq 2$ з кроком 0.05 та початковими умовами $x(0) = 2$ та $y(0) = 3$:

MATLAB	C#
<pre>function [res] = RKs(fs, a, x0, y0, h) xv = (x0:h:a)'; yv = zeros(length(xv), numel(y0)); yv(1,:) = y0(:,1); for i=1:(length(xv)-1) ys1 = num2cell(yv(i,:),:); k1 = fs(ys1{:})'; ys2 = num2cell(yv(i,:) + k1 * 0.5 * h); k2 = fs(ys2{:})'; ys3 = num2cell(yv(i,:) + k2 * 0.5 * h); k3 = fs(ys3{:})'; ys4 = num2cell(yv(i,:) + k3 * h); k4 = fs(ys4{:})'; yv(i+1,:) = yv(i,:) + (k1 + 2 * k2 + 2 * k3 + k4) * (h / 6); end res = [xv, yv]; end</pre>	<pre>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,</pre>

```
>> f = @(y1, y2) [ 0.9 * y1 - 4 * y2; y1 + 0.9 * y2 ]
f =
function_handle with value:
@(y1,y2) [0.9*y1-4*y2;y1+0.9*y2]
>> res = RKs(f, 2, 0, [2; 3], 0.05)
>> plot(res(:,1), res(:,2), res(:,1), res(:,3))
```



```
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 диференціальних рівнянь та їх систем. Досліджену поведінку математичного середовища було відтворено за допомогою мови програмування C#.