Дисциплина: Численные методы Лабораторное задание №2

Отчет

Тема: Численные методы типа Рунге - Кутта

Выполнила:

студентка 3 курса 61 группы

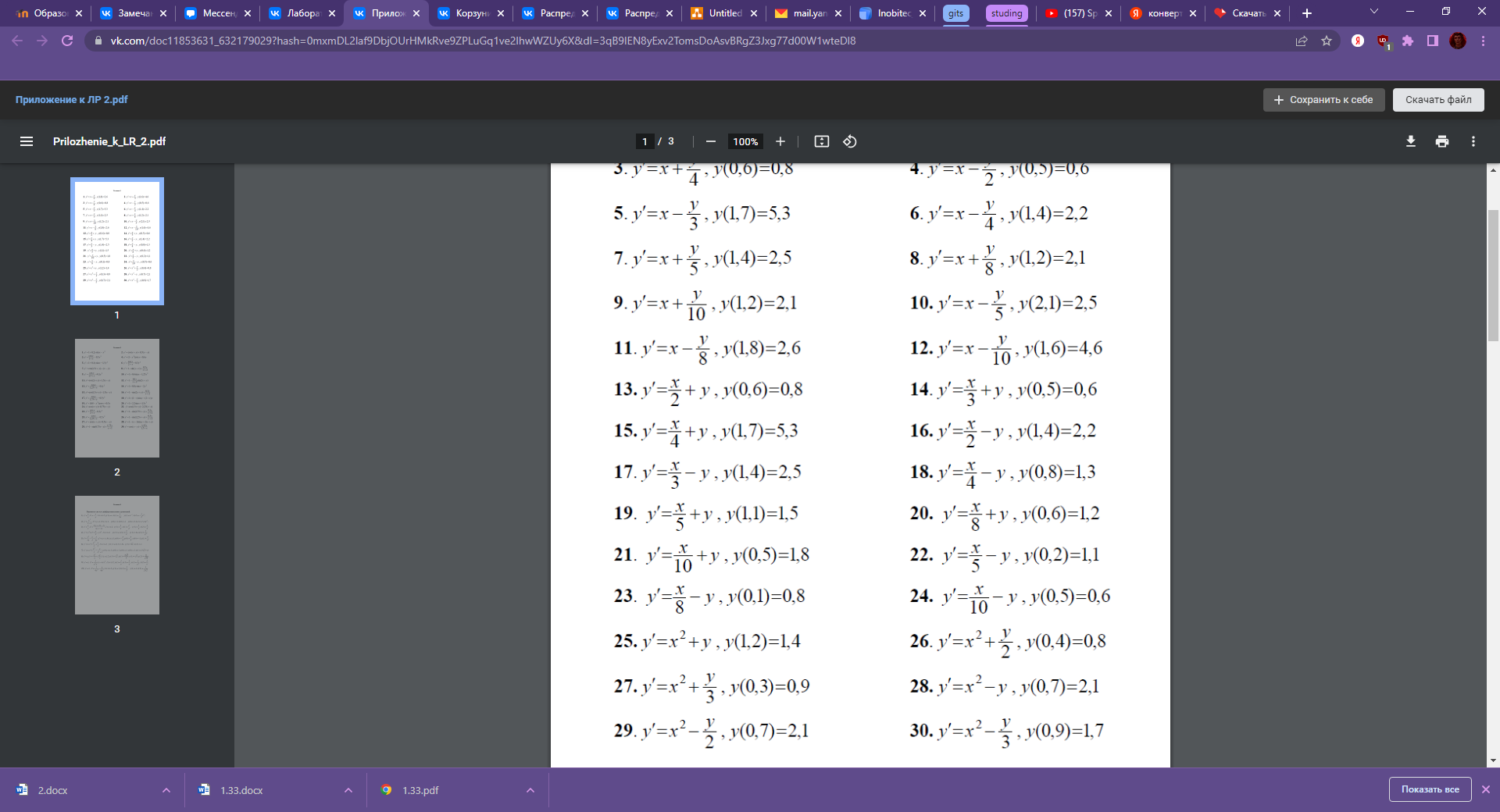
Шелудякова Е.М.

Проверил: преподаватель Махинова О.А.

# Задача 1. Численное решение задачи Коши для обыкновенного дифференциального уравнения 1-ого порядка методами типа Рунге-Кутта

**Постановка задачи**

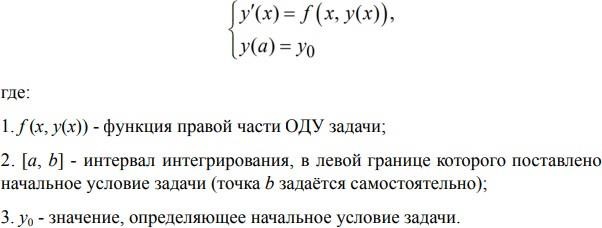
Найти численное решение задачи Коши для ОДУ 1-ого порядка трехчленным двухсторонним методом Рунге-Кутта с требуемой точностью ε, увеличивая количество подотрезков разбиения до 2n. Исследовать две задачи Коши, вычисляя в одном из них аналитическое (точное) решение и численное решение в другом. Проанализировать результаты для условий:



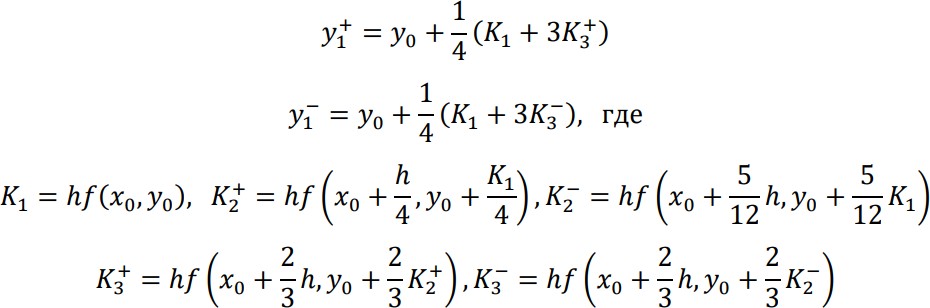
# 

# Теоретические сведения

Задача Коши для ОДУ 1-го порядка имеет вид:



Численное решение задачи Коши при помощи трехчленного двухстороннего метода Рунге – Кутта осуществляется по следующим формулам:



Процесс удвоения отрезков и вычисления ε происходит до тех пор, пока не будет выполнено условие ε(n, 2n) ≤ ε0 , где ε0 – заданная точность метода, а ε(n, 2n) имеет вид:

где:

𝜀(𝑛,2𝑛) =

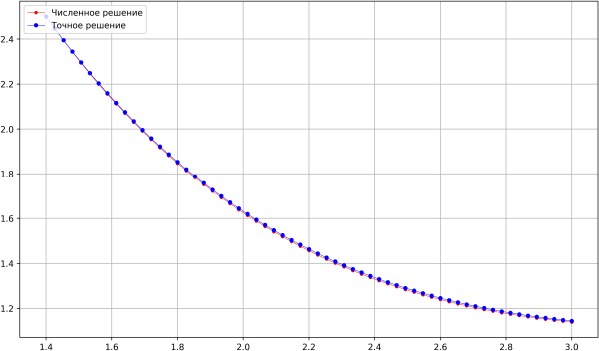
|𝑦𝑛 − 𝑦2𝑛| 2𝑃 − 1 ,

* yn – численное решение в крайнем правом узле, найденное на сетке из n узлов;
* y2n – численное решение в крайнем правом узле, найденное на сетке из 2n узлов.

# Вычислительный эксперимент

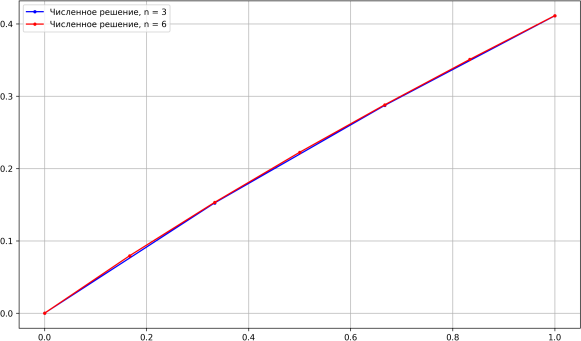
# 1-е условие

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | 𝒂(𝒙𝟎) | 𝒃 | 𝒏𝟎 | 𝒚𝟎 | **Требуемая точность** | 𝒏 **на**  **полученном решении** | 𝒚(𝒃) |
| 1 | 1,6 | 3 | 4 | 4,6 | 0,005 | 64 | 7,025093 |
| 2 | 1,6 | 3 | 4 | 4,6 | 0,00005 | 65536 | 7,024747 |



2-е условие

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | 𝒂(𝒙𝟎) | 𝒃 | 𝒏𝟎 | 𝒚𝟎 | **Требуемая точность** | 𝒏 **на**  **полученном решении** | 𝒚(𝒃) |
| 1 | 0 | 3 | 4 | 0 | 0,005 | 16 | 1,00568 |
| 2 | 0 | 3 | 4 | 0 | 0,00005 | 1024 | 1,04684 |



# Листинг

public double CountYMinus(double xPrev, double yPrev, double h)

{

return yPrev + (2 \* K1(xPrev, yPrev, h) + K3Minus(xPrev, yPrev, h) \* 3)/3;

}

public double CountYPlus(double xPrev, double yPrev, double h)

{

return yPrev + K3Plus(xPrev, yPrev, h);

}

public double K3Plus(double xPrev, double yPrev, double h)

{

return h \* F(xPrev + h / 2, yPrev + K2(xPrev, yPrev, h) / 2);

}

public double K3Minus(double xPrev, double yPrev, double h)

{

return h \* F(xPrev + h \* 5/6, yPrev + K2(xPrev, yPrev, h) \* 5/6);

}

public double K2(double xPrev, double yPrev, double h)

{

return h \* F(xPrev + h / 3, yPrev + K1(xPrev, yPrev, h) / 3);

}

public double K1(double xPrev, double yPrev, double h)

{

return h \* F(xPrev, yPrev);

}

public void CountFunc()

{

double yPlus = CountYPlus(Function[0, 0], Function[1, 0], H);

double yMinus = CountYMinus(Function[0, 0], Function[1, 0], H);

Function[1, 1] = (yMinus + yPlus) / 2.0;

for (int i = 2; i < Function.GetLength(1); i++)

{

double \_yPlus = CountYPlus(Function[0, i - 1], yPlus, H);

double \_yMinus = CountYMinus(Function[0, i - 1], yMinus, H);

Function[1, i] = (\_yMinus + \_yPlus) / 2.0;

yPlus = \_yPlus;

yMinus = \_yMinus;

}

}

public void CountAnaliticFunc()

{

double C = (Y0 / Math.Pow(Math.E, -A)) - (A - 1) \* Math.Pow(Math.E, A) / 8;

for (int i = 0; i < Function.GetLength(1); i++)

{

AnaliticFunction[0, i] = Function[0, i];

AnaliticFunction[1, i] = Math.Pow(Math.E, -AnaliticFunction[0, i]) \* ((AnaliticFunction[0, i] - 1) \* Math.Pow(Math.E, AnaliticFunction[0, i]) / 8 + C);

}

}

public RungeKutta(double a, double b, int n, double eps, double y0)

{

N = n; A = a; B = b; Eps = eps; Y0 = y0;

Function = new double[2, n + 1];

AnaliticFunction = new double[2, n + 1];

H = (b - a) / n;

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

{

Function[0, i] = a + H \* i;

}

Function[1, 0] = y0;

CountAnaliticFunc();

CountFunc();

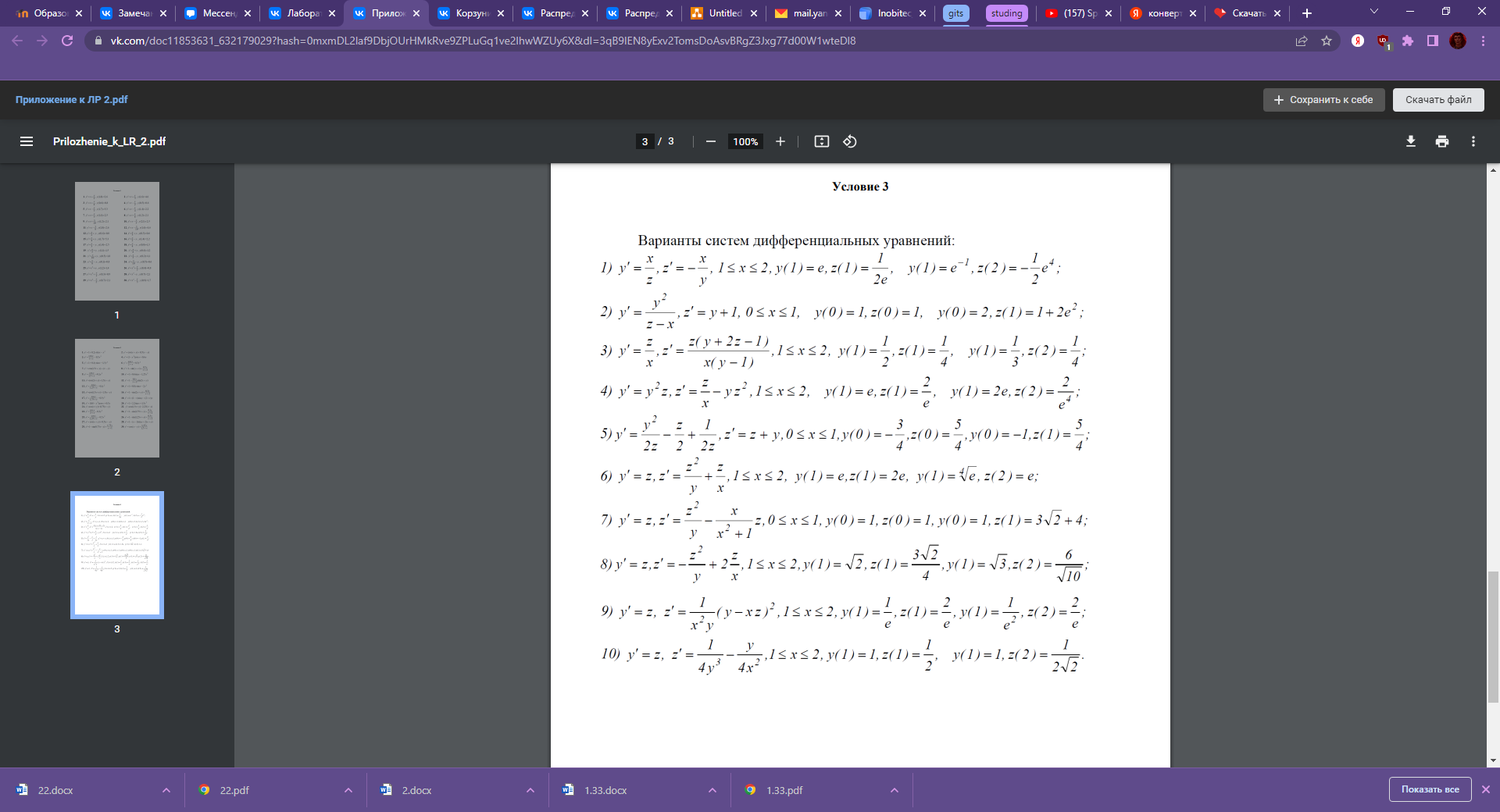
CountError();

# }

# Задача 2. Численное решение задачи Коши для системы обыкновенных дифференциальных уравнений 1-ого порядка методами типа Рунге-Кутта

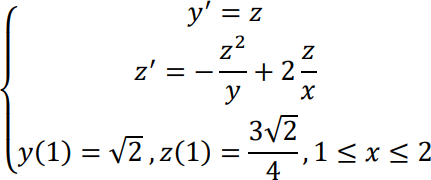
**Постановка задачи**

Найти численное решение задачи Коши для системы ОДУ 1-ого порядка методом Рунге-Кутта второго порядка с требуемой точностью для условия

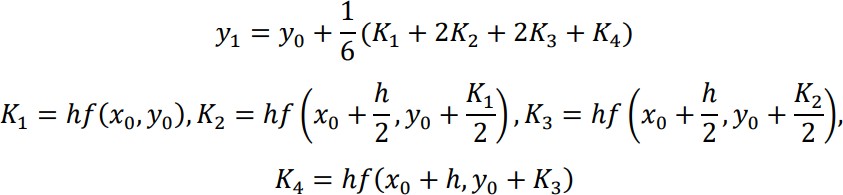


# Теоретические сведения:

Задачи Коши для системы ОДУ 1-ого порядка имеет вид:



Стандартная формула Рунге – Кутта четвертого порядка:



Процесс удвоения отрезков и вычисления ε происходит до тех пор, пока не будет выполнено условие ε(n, 2n) ≤ ε0 для обеих функций системы , где ε0

– заданная точность метода, а ε(n, 2n) имеет вид:

|𝑦𝑛 − 𝑦2𝑛|

𝜀(𝑛,2𝑛) = 2𝑃 − 1

где:

* yn – численное решение в крайнем правом узле, найденное на сетке из n узлов;
* y2n – численное решение в крайнем правом узле, найденное на сетке из 2n узлов.

# Вычислительный эксперимент:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | 𝒂(𝒙𝟎) | 𝒃 | 𝒏𝟎 | 𝒚𝟎 | 𝒛𝟎 | **Требуемая точность** | 𝒏 **на полученном решении** | 𝒚(𝒃) | 𝒛(𝒃) |
| 1 | 1 | 2 | 4 | 𝑒 | 2𝑒 | 0,5 | 512 | 54,14294 | 216,27654 |
| 2 | 1 | 2 | 4 | 𝑒 | 2𝑒 | 0,05 | 4096 | 54,5412 | 218,12786 |
| 2 | 1 | 2 | 4 | 𝑒 | 2𝑒 | 0,005 | 32768 | 54,591 | 218,3595 |

График эксперимента №1 (желтый график y на последней итерации, красный график – график z на последней итерации, голубой график – график z на первой итерации, пурпурный график - график y на первой итерации)

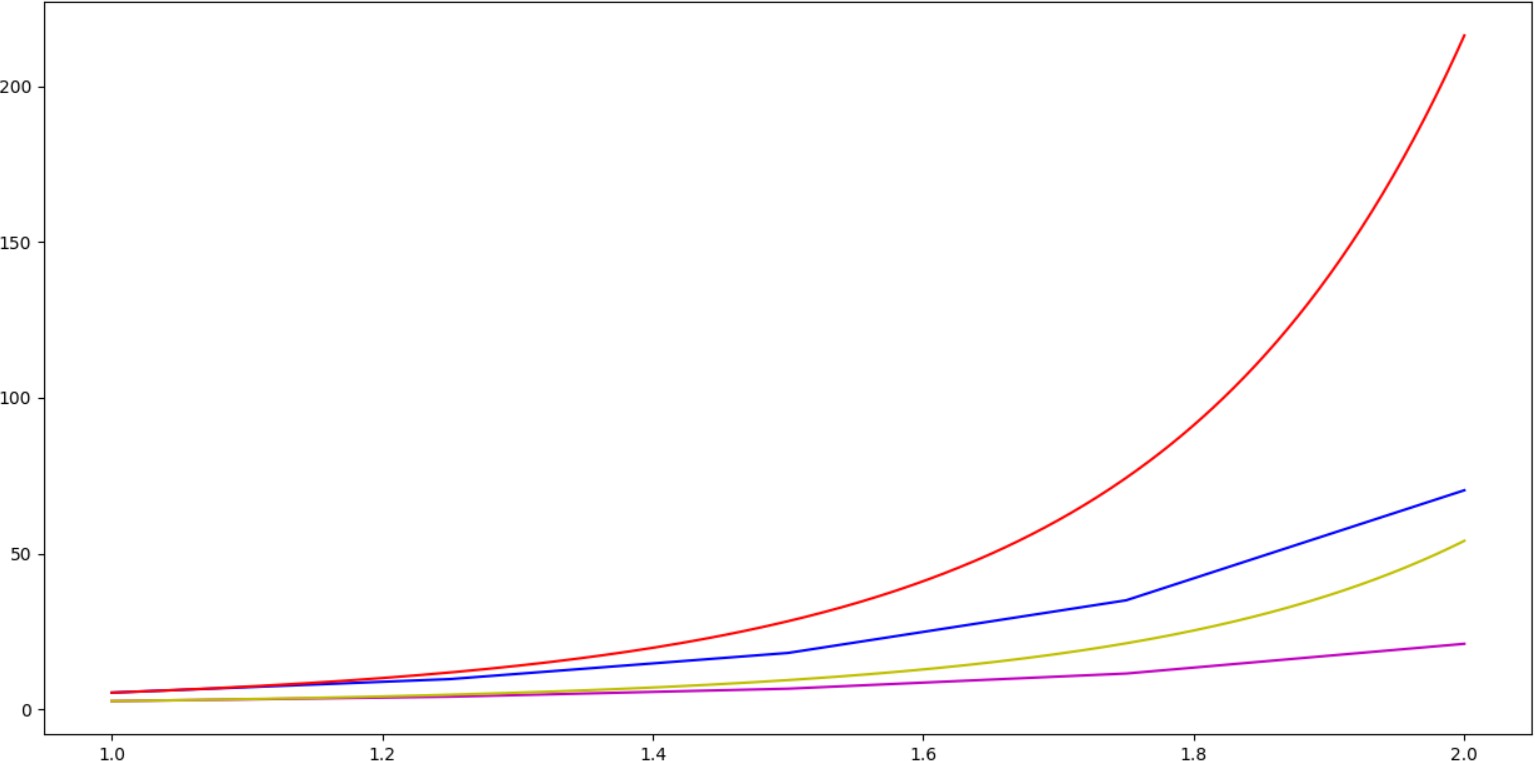
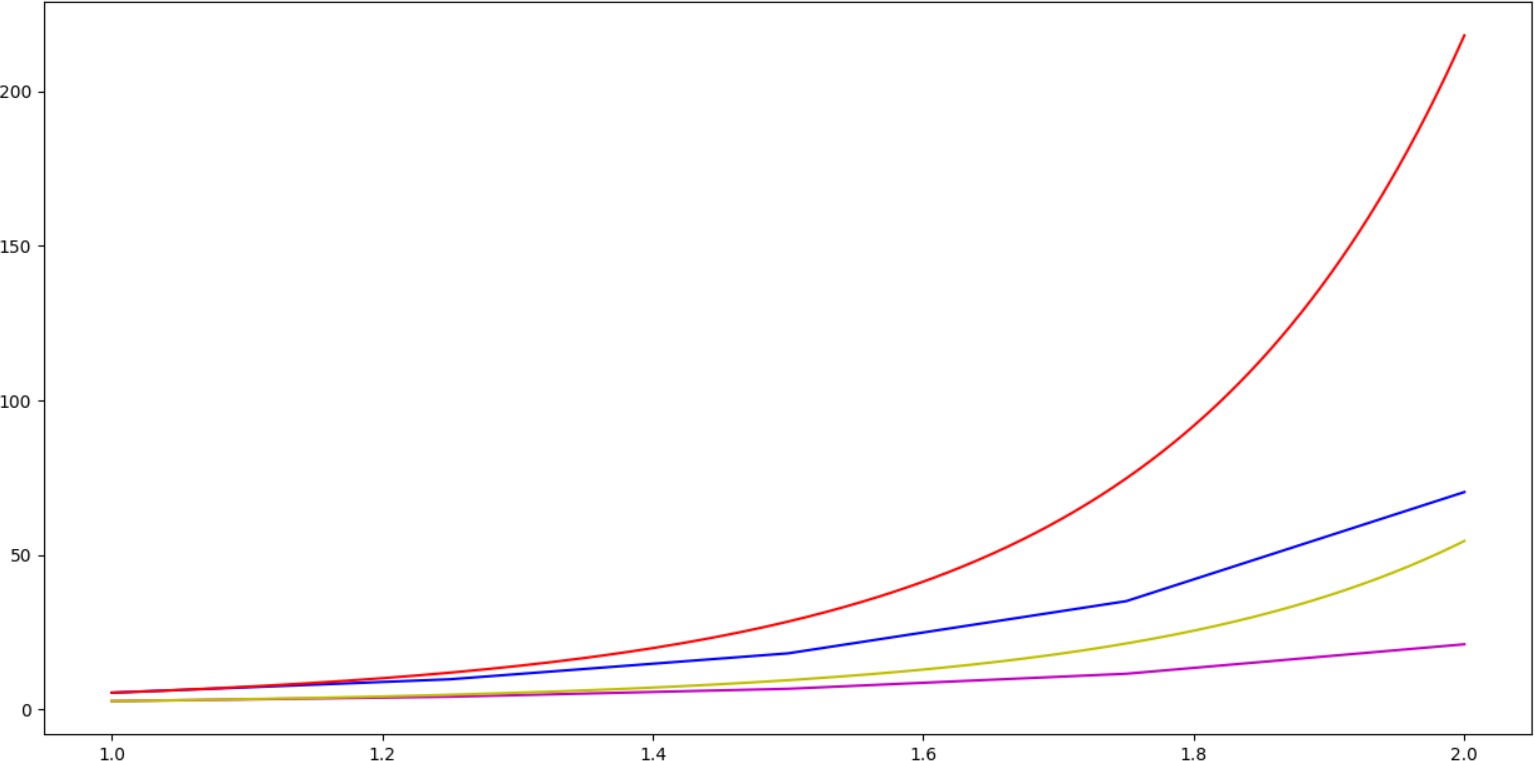


График эксперимента №2 (желтый график y на последней итерации, красный график – график z на последней итерации, голубой график – график z на первой итерации, пурпурный график - график y на первой итерации)



# Листинг

private static Solution GetSolution(double h, double y0, double z0, double[] x, int pointsQuantity, ref double[] yi, ref double[] zi)

{

var yprev = y0;

var zprev = z0;

zi = new double[pointsQuantity];

yi = new double[pointsQuantity];

yi[0] = y0;

zi[0] = z0;

for (var i = 1; i < pointsQuantity; i++)

{

var k1 = h \* Y(x[i - 1], yprev, zprev);

var l1 = h \* Z(x[i - 1], yprev, zprev);

var k2 = h \* Y(x[i - 1] + h / 2, yprev + k1 / 2, zprev + l1 / 2);

var l2 = h \* Z(x[i - 1] + h / 2, yprev + k1 / 2, zprev + l1 / 2);

var k3 = h \* Y(x[i - 1] + h / 2, yprev + k2 / 2, zprev + l2 / 2);

var l3 = h \* Z(x[i - 1] + h / 2, yprev + k2 / 2, zprev + l2 / 2);

var k4 = h \* Y(x[i - 1] + h, yprev + k3, zprev + l3);

var l4 = h \* Z(x[i - 1] + h, yprev + k3, zprev + l3);

yprev = yprev + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;

zprev = zprev + (l1 + 2 \* l2 + 2 \* l3 + l4) / 6;

yi[i] = yprev;

zi[i] = zprev;

}

return new Solution(yprev, zprev);

**Вывод:**

При помощи метода Рунге-Кутта четвертого порядка и трехчленного двухстороннего метода Рунге-Кутта, мы нашли численные решения задачи Коши 1-го порядка, достигнув необходимой точности. Можем сделать вывод, что метод Рунге-Кутта является оптимально подходящим для нахождения решения задачи Коши для системы ОДУ 1-го порядка. Метод сходится, продемонстрировано на примерах, что при возрастании n, погрешность уменьшается. Так же при слишком высокой требуемой точности вычисление может прерваться из-за слишком маленького шага или отсутствия уменьшения погрешности с каждой новой итерацией.