|  |  |  |
| --- | --- | --- |
| ЭМБЛЕМА 41 | **МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  **имени М.В.Ломоносова** | vmk |
| **Факультет вычислительной математики и кибернетики** | | |

**Компьютерный практикум по учебному курсу**

**«ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ»**

**ЗАДАНИЕ № 2.**

**Численные методы решения дифференциальных уравнений**

**ОТЧЕТ**

**о выполненном задании**

студента 204 учебной группы факультета ВМК МГУ

Столярова Романа Константиновича

(фамилия, имя, отчество)

гор. Москва

2020 г.

### Оглавление

Цели и задачи……………………………………………………………………………….III

Описание методов Рунге-Кутта 2-го и 4-го порядка…………………..VII

Описание метода прогонки и метода конечных разностей

для решения краевой задачи для дифференциального

уравнения второго порядка ………………………………………………………VIII

Тестирование………………………………………………………………………………..X

Исходные коды программ на языке С………………………………………..XX

Выводы………………………………………………………………………………………XXV

**Цели и задачи**

**Метод Рунге-Кутта решения задачи Коши для дифференциального уравнения 1-го порядка**

**Цель работы**

Освоить методы Рунге-Кутта второго и четвертого порядка точности, применяемые для численного решения задачи Коши для дифференциального уравнения (или системы дифференциальных уравнений) первого порядка.

**Постановка задачи**

Рассматривается обыкновенное дифференциальное уравнение первого порядка, разрешенное относительно производной и имеющее вид:

(1)

с дополнительным начальным условием, заданным в точке :

(2)

Предполагается, что правая часть уравнения (1) функция  такова, что гарантирует существование и единственность решения задачи Коши (1)-(2).

В том случае, если рассматривается не одно дифференциальное уравнение вида (1), а система обыкновенных дифференциальных уравнений первого порядка, разрешенных относительно производных неизвестных функций, то соответствующая задача Коши имеет вид (на примере двух дифференциальных уравнений):

(3)

Дополнительные (начальные) условия задаются в точке :

(4)

Также предполагается, что правые части уравнений из (3) заданы так, что это гарантирует существование и единственность решения задачи Коши (3)-(4), но уже для системы обыкновенных дифференциальных уравнений первого порядка в форме, разрешенной относительно производных неизвестных функций.

Заметим, что к подобным задачам сводятся многие важные задачи, возникающие в механике (уравнения движения материальной точки), небесной механике, химической кинетике, гидродинамике и т.п.

**Цели и задачи практической работы**

1. Решить задачу Коши (1)-(2) (или (3)-(4)) наиболее известными и широко используемыми на практике методами Рунге-Кутта второго и четвертого порядка точности, аппроксимировав дифференциальную задачу соответствующей разностной схемой (на равномерной сетке); полученное конечно-разностное уравнение (или уравнения в случае системы), представляющее фактически некоторую рекуррентную формулу, просчитать численно;
2. Найти численное решение задачи и построить его график;
3. Найденное численное решение сравнить с точным решением дифференциального уравнения, полученным при помощи онлайн системы Wolfram Alpha.

**Метод прогонки и метод конечных разностей для решения краевой задачи для дифференциального уравнения второго порядка**

**Цель работы**

Освоить метод прогонки решения краевой задачи для дифференциального уравнения второго порядка.

**Постановка задачи**

Рассматривается линейное дифференциальное уравнение второго порядка вида

(1)

с дополнительными условиями в граничных точках

(2)

**Цели и задачи практической работы**

1. Решить краевую задачу (1)-(2) методом конечных разностей, аппроксимировав ее разностной схемой второго порядка точности (на равномерной сетке); полученную систему конечно-разностных уравнений решить методом прогонки;
2. Найти разностное решение задачи и построить его график;
3. Найденное разностное решение сравнить с точным решением дифференциального уравнения, полученным при помощи онлайн системы Wolfram Alpha.

**Описание методов Рунге-Кутта 2-го и 4-го порядка**

Пусть дана задача Коши:

И на отрезке [x0, x0 + a] задана равномерная сетка:

Сопоставим этой равномерной сетке схему Рунге-Кутта II порядка точности:

;

;

.

и IV порядка точности:

;

;

;

;

.

Пусть дана система дифференциальных уравнений с начальными условиями:

Для ее решения воспользуемся следующими рекуррентными соотношениями схемы Рунге-Кутта:

II порядка:

;

;

;

;

.

IV порядка:

;

;

;

;

;

;

;

;

;

.

Результатом применения данных методов будут сеточные функции определенные на сетке:

**Описание метода прогонки и метода конечных разностей для решения краевой задачи для дифференциального уравнения второго порядка**

Рассмотрим дифференциальное уравнение вида:

(1)

C дополнительными условиями в граничных точках:

(2)

Разобьем отрезок [a, b] на сетку из n частей:

Приблизим производные в исходном уравнении конечно-разностными отношениями:

=

=

Приблизим производные в дополнительных условиях:

Соберем коэффициенты во всех полученных равенствах при соответствующих и получим систему линейных уравнений с постоянными коэффициентами имеющую трехдиагональную матрицу.

Решим данную систему методом прогонки. Решение будем искать в виде:

С прогоночными коэффициентами:

;

**Тестирование**

Для тестирования методов Рунге-Кутта будем использовать тесты, предложенные в варианте (1-3 и 2-17), а также наборы функций для дополнительного тестирования (2 для одиночного уравнение и 1 для системы). Проверку результатов и погрешности будем проводить через Wolfram Alpha и встроенные функции из библиотеки языка С <math.h>.

**Основной тест:**

в точке

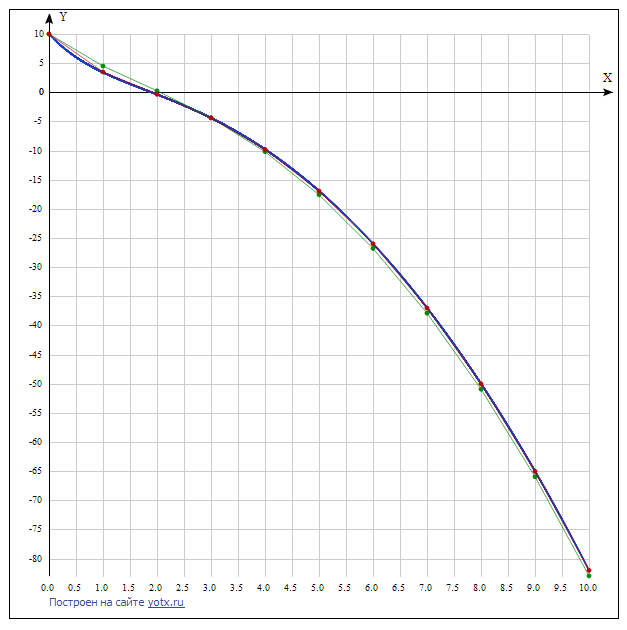
Аналитическое решение

Решение методом Рунге-Кутта на отрезке [0,1] c n = 10:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 0.100 correct = 9.048 y = 9.050  x = 0.200 correct = 8.185 y = 8.187  x = 0.300 correct = 7.400 y = 7.403  x = 0.400 correct = 6.684 y = 6.688  x = 0.500 correct = 6.028 y = 6.033  x = 0.600 correct = 5.426 y = 5.430  x = 0.700 correct = 4.869 y = 4.874  x = 0.800 correct = 4.352 y = 4.357  x = 0.900 correct = 3.869 y = 3.874  x = 1.000 correct = 3.415 y = 3.419 | **4-го порядка:**  x = 0.100 correct = 9.048 y = 9.048  x = 0.200 correct = 8.185 y = 8.185  x = 0.300 correct = 7.400 y = 7.400  x = 0.400 correct = 6.684 y = 6.684  x = 0.500 correct = 6.028 y = 6.028  x = 0.600 correct = 5.426 y = 5.426  x = 0.700 correct = 4.869 y = 4.869  x = 0.800 correct = 4.352 y = 4.352  x = 0.900 correct = 3.869 y = 3.869  x = 1.000 correct = 3.415 y = 3.415 |
| **Погрешность:**  **0.0039819441** | **Погрешность:**  **0.0000021898** |

Решение методом Рунге-Кутта на отрезке [0,10] c n = 10:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 1.000 correct = 3.415 y = 4.500  x = 2.000 correct = -0.376 y = 0.250  x = 3.000 correct = -4.403 y = -4.375  x = 4.000 correct = -9.780 y = -10.188  x = 5.000 correct = -16.919 y = -17.594  x = 6.000 correct = -25.970 y = -26.797  x = 7.000 correct = -36.989 y = -37.898  x = 8.000 correct = -49.996 y = -50.949  x = 9.000 correct = -64.999 y = -65.975  x = 10.000 correct = -81.999 y = -82.987 | **4-го порядка:**  x = 1.000 correct = 3.415 y = 3.479  x = 2.000 correct = -0.376 y = -0.341  x = 3.000 correct = -4.403 y = -4.399  x = 4.000 correct = -9.780 y = -9.795  x = 5.000 correct = -16.919 y = -16.944  x = 6.000 correct = -25.970 y = -26.000  x = 7.000 correct = -36.989 y = -37.021  x = 8.000 correct = -49.996 y = -50.029  x = 9.000 correct = -64.999 y = -65.032  x = 10.000 correct = -81.999 y = -82.033 |
| **Погрешность:**  **0.7474055708** | **Погрешность:**  **0.0303606408** |



Синяя – аналитическое решение

Зеленая – 2-й порядок

Красная – 4-й порядок

**Дополнительные тесты:**

в точке

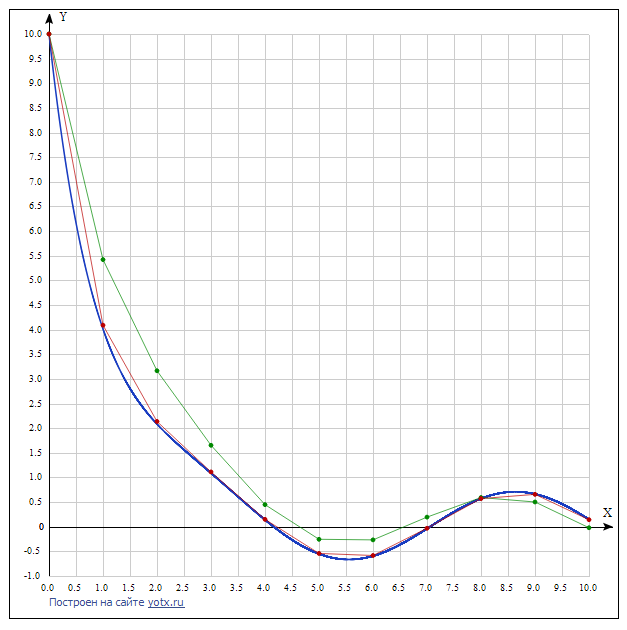
Аналитическое решение

Решение методом Рунге-Кутта на отрезке [0,1] c n = 10:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 0.100 correct = 8.601 y = 9.055  x = 0.200 correct = 7.797 y = 8.209  x = 0.300 correct = 7.078 y = 7.453  x = 0.400 correct = 6.437 y = 6.778  x = 0.500 correct = 5.866 y = 6.175  x = 0.600 correct = 5.358 y = 5.639  x = 0.700 correct = 4.906 y = 5.160  x = 0.800 correct = 4.504 y = 4.735  x = 0.900 correct = 4.147 y = 4.357  x = 1.000 correct = 3.829 y = 4.020 | **4-го порядка:**  x = 0.100 correct = 8.601 y = 9.053  x = 0.200 correct = 7.797 y = 8.206  x = 0.300 correct = 7.078 y = 7.449  x = 0.400 correct = 6.437 y = 6.773  x = 0.500 correct = 5.866 y = 6.169  x = 0.600 correct = 5.358 y = 5.632  x = 0.700 correct = 4.906 y = 5.154  x = 0.800 correct = 4.504 y = 4.728  x = 0.900 correct = 4.147 y = 4.350  x = 1.000 correct = 3.829 y = 4.013 |
| **Погрешность:**  **0.3059165879** | **Погрешность:**  **0.3005233118** |

Решение методом Рунге-Кутта на отрезке [0,10] c n = 10:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 1.000 correct = 3.829 y = 5.421  x = 2.000 correct = 2.016 y = 3.165  x = 3.000 correct = 1.063 y = 1.653  x = 4.000 correct = 0.132 y = 0.448  x = 5.000 correct = -0.554 y = -0.255  x = 6.000 correct = -0.595 y = -0.267  x = 7.000 correct = -0.039 y = 0.195  x = 8.000 correct = 0.571 y = 0.592  x = 9.000 correct = 0.663 y = 0.502  x = 10.000 correct = 0.148 y = -0.021 | **4-го порядка:**  x = 1.000 correct = 3.829 y = 4.090  x = 2.000 correct = 2.016 y = 2.136  x = 3.000 correct = 1.063 y = 1.112  x = 4.000 correct = 0.132 y = 0.150  x = 5.000 correct = -0.554 y = -0.542  x = 6.000 correct = -0.595 y = -0.584  x = 7.000 correct = -0.039 y = -0.031  x = 8.000 correct = 0.571 y = 0.571  x = 9.000 correct = 0.663 y = 0.657  x = 10.000 correct = 0.148 y = 0.142 |
| **Погрешность:**  **0.4857736985** | **Погрешность:**  **0.0491554289** |



Синяя – аналитическое решение

Зеленая – 2-й порядок

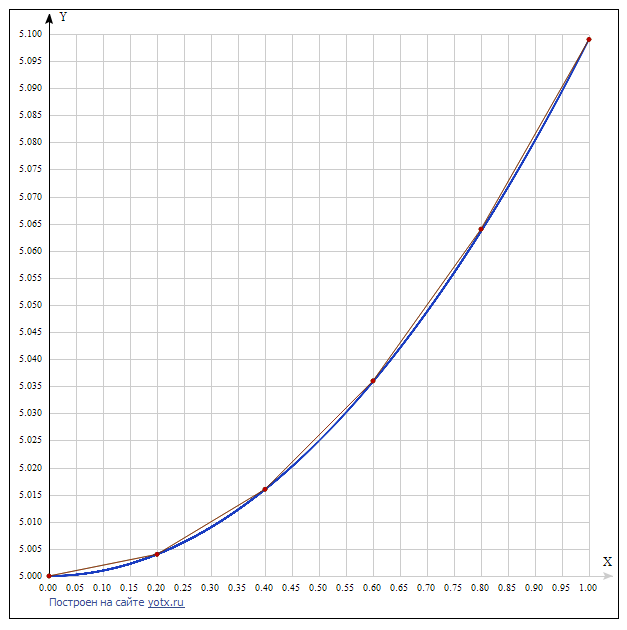
Красная – 4-й порядок

в точке

Аналитическое решение

Решение методом Рунге-Кутта на отрезке [0,1] c n = 5:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 0.200 correct = 5.004 y = 5.004  x = 0.400 correct = 5.016 y = 5.016  x = 0.600 correct = 5.036 y = 5.036  x = 0.800 correct = 5.064 y = 5.064  x = 1.000 correct = 5.099 y = 5.099 | **4-го порядка:**  x = 0.200 correct = 5.004 y = 5.004  x = 0.400 correct = 5.016 y = 5.016  x = 0.600 correct = 5.036 y = 5.036  x = 0.800 correct = 5.064 y = 5.064  x = 1.000 correct = 5.099 y = 5.099 |
| **Погрешность:**  **0.0000046512** | **Погрешность:**  **0.0000000022** |



Синяя – аналитическое решение

Зеленая – 2-й порядок

Красная – 4-й порядок

Решение методом Рунге-Кутта на отрезке [0,10] c n = 5:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 2.000 correct = 5.385 y = 5.400  x = 4.000 correct = 6.403 y = 6.422  x = 6.000 correct = 7.810 y = 7.827  x = 8.000 correct = 9.434 y = 9.448  x = 10.000 correct = 11.180 y = 11.193 | **4-го порядка:**  x = 2.000 correct = 5.385 y = 5.385  x = 4.000 correct = 6.403 y = 6.403  x = 6.000 correct = 7.810 y = 7.811  x = 8.000 correct = 9.434 y = 9.434  x = 10.000 correct = 11.180 y = 11.181 |
| **Погрешность:**  **0.0154159070** | **Погрешность:**  **0.0002819929** |

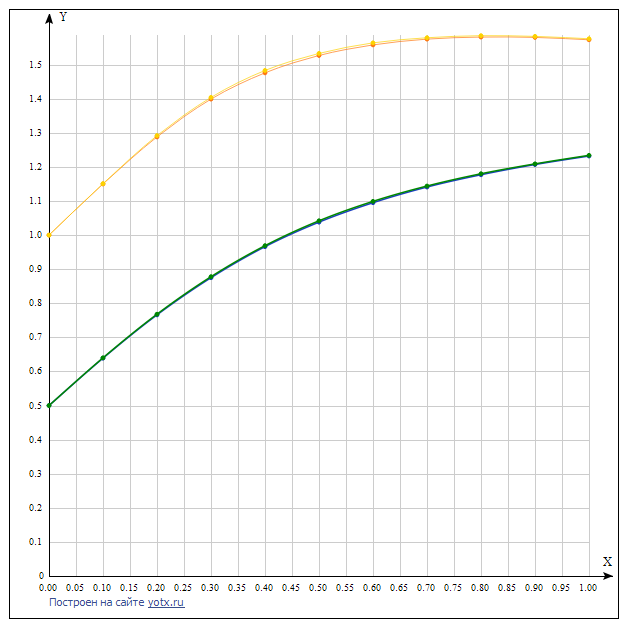
**Основной тест для системы уравнений:**

В точке

Аналитическое решение не было найдено системой Wolfram Alpha

Решение методом Рунге-Кутта на отрезке [0,1] c n = 10:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 0.100 u = 1.150 v = 0.639  x = 0.200 u = 1.288 v = 0.766  x = 0.300 u = 1.399 v = 0.875  x = 0.400 u = 1.476 v = 0.966  x = 0.500 u = 1.527 v = 1.038  x = 0.600 u = 1.558 v = 1.095  x = 0.700 u = 1.575 v = 1.141  x = 0.800 u = 1.581 v = 1.177  x = 0.900 u = 1.580 v = 1.207  x = 1.000 u = 1.573 v = 1.232 | **4-го порядка:**  x = 0.100 u = 1.151 v = 0.640  x = 0.200 u = 1.292 v = 0.768  x = 0.300 u = 1.404 v = 0.878  x = 0.400 u = 1.483 v = 0.969  x = 0.500 u = 1.533 v = 1.042  x = 0.600 u = 1.564 v = 1.099  x = 0.700 u = 1.579 v = 1.144  x = 0.800 u = 1.585 v = 1.180  x = 0.900 u = 1.583 v = 1.209  x = 1.000 u = 1.576 v = 1.234 |

График с аппроксимацией полученных кривых:

Синяя – 2-й порядок v(x)

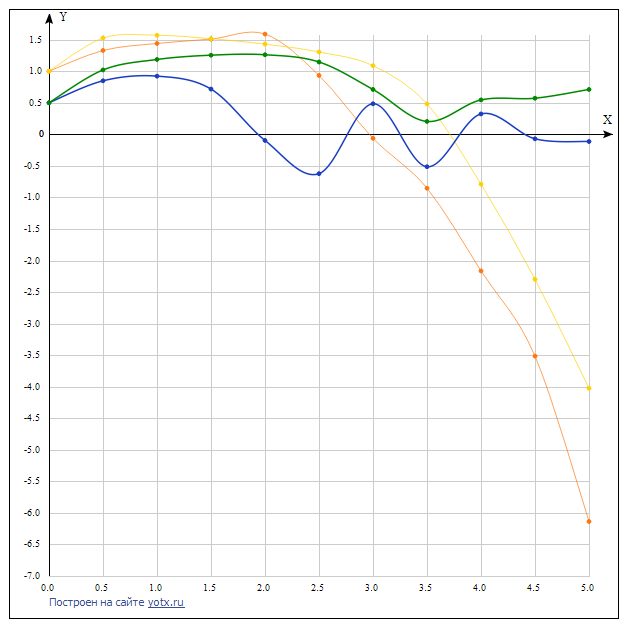
Зеленая – 4-й порядок v(x)

Оранжевая – 2-й порядок u(x)

Желтая – 4-й порядок u(x)

Решение методом Рунге-Кутта на отрезке [0,5] c n = 10:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 0.500 u = 1.329 v = 0.848  x = 1.000 u = 1.440 v = 0.922  x = 1.500 u = 1.509 v = 0.717  x = 2.000 u = 1.589 v = -0.097  x = 2.500 u = 0.933 v = -0.624  x = 3.000 u = -0.063 v = 0.484  x = 3.500 u = -0.854 v = -0.514  x = 4.000 u = -2.165 v = 0.323  x = 4.500 u = -3.515 v = -0.071  x = 5.000 u = -6.136 v = -0.114 | **4-го порядка:**  x = 0.500 u = 1.526 v = 1.020  x = 1.000 u = 1.570 v = 1.186  x = 1.500 u = 1.516 v = 1.254  x = 2.000 u = 1.430 v = 1.261  x = 2.500 u = 1.305 v = 1.146  x = 3.000 u = 1.088 v = 0.711  x = 3.500 u = 0.482 v = 0.204  x = 4.000 u = -0.790 v = 0.545  x = 4.500 u = -2.297 v = 0.572  x = 5.000 u = -4.024 v = 0.711 |

График с аппроксимацией полученных кривых:

Синяя – 2-й порядок v(x)

Зеленая – 4-й порядок v(x)

Оранжевая – 2-й порядок u(x)

Желтая – 4-й порядок u(x)

**Дополнительный тест для системы уравнений:**

В точке

Аналитическое решение:

Решение методом Рунге-Кутта на отрезке [0,1] c n = 10:

|  |  |
| --- | --- |
| **2-го порядка:**  x = 0.100 cor1 = 0.277 u = 0.277 cor2 = 1.105 v = 1.105  x = 0.200 cor1 = 0.307 u = 0.307 cor2 = 1.221 v = 1.221  x = 0.300 cor1 = 0.341 u = 0.340 cor2 = 1.350 v = 1.349  x = 0.400 cor1 = 0.378 u = 0.378 cor2 = 1.492 v = 1.491  x = 0.500 cor1 = 0.420 u = 0.420 cor2 = 1.649 v = 1.647  x = 0.600 cor1 = 0.468 u = 0.467 cor2 = 1.822 v = 1.820  x = 0.700 cor1 = 0.522 u = 0.521 cor2 = 2.014 v = 2.012  x = 0.800 cor1 = 0.583 u = 0.582 cor2 = 2.226 v = 2.223  x = 0.900 cor1 = 0.654 u = 0.653 cor2 = 2.460 v = 2.456  x = 1.000 cor1 = 0.738 u = 0.736 cor2 = 2.718 v = 2.714 | **4-го порядка:**  x = 0.100 cor1 = 0.277 u = 0.277 cor2 = 1.105 v = 1.105  x = 0.200 cor1 = 0.307 u = 0.307 cor2 = 1.221 v = 1.221  x = 0.300 cor1 = 0.341 u = 0.341 cor2 = 1.350 v = 1.350  x = 0.400 cor1 = 0.378 u = 0.378 cor2 = 1.492 v = 1.492  x = 0.500 cor1 = 0.420 u = 0.420 cor2 = 1.649 v = 1.649  x = 0.600 cor1 = 0.468 u = 0.468 cor2 = 1.822 v = 1.822  x = 0.700 cor1 = 0.522 u = 0.522 cor2 = 2.014 v = 2.014  x = 0.800 cor1 = 0.583 u = 0.583 cor2 = 2.226 v = 2.226  x = 0.900 cor1 = 0.654 u = 0.654 cor2 = 2.460 v = 2.460  x = 1.000 cor1 = 0.738 u = 0.738 cor2 = 2.718 v = 2.718 |
| **Погрешность u(x):**  **0.0006314561** | **Погрешность:**  **0.0000002099** |
| **Погрешность v(x):**  **0.0017615490** | **Погрешность:**  **0.0000008738** |

Для тестирования метода конечных разностей и метода прогонки будем использовать тесты, предложенные в варианте (1-8), а также 2 краевые задачи для дополнительного тестирования. Проверку результатов и погрешности будем проводить через Wolfram Alpha и встроенные функции из библиотеки языка С <math.h>.

**Основной тест:**

с условиями

Аналитическое решение не было найдено системой Wolfram Alpha

Решение методом прогонки на отрезке [1.2,1.5] c n = 10:

x = 1.500 y = 0.732

x = 1.470 y = 0.703

x = 1.440 y = 0.674

x = 1.410 y = 0.646

x = 1.380 y = 0.617

x = 1.350 y = 0.588

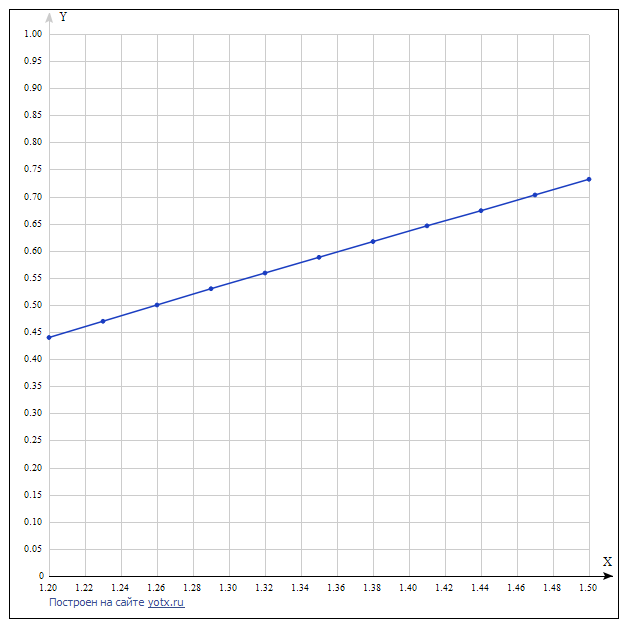
x = 1.320 y = 0.559

x = 1.290 y = 0.530

x = 1.260 y = 0.500

x = 1.230 y = 0.470

x = 1.200 y = 0.440

График с аппроксимацией полученной кривой:

**Дополнительные тесты:**

с условиями

Аналитическое решение

Решение методом прогонки на отрезке [0,1] c n = 10:

x = 1.000 correct = 1.000 y = 1.000

x = 0.900 correct = 0.704 y = 0.717

x = 0.800 correct = 0.444 y = 0.469

x = 0.700 correct = 0.215 y = 0.250

x = 0.600 correct = 0.013 y = 0.057

x = 0.500 correct = -0.167 y = -0.115

x = 0.400 correct = -0.327 y = -0.267

x = 0.300 correct = -0.471 y = -0.404

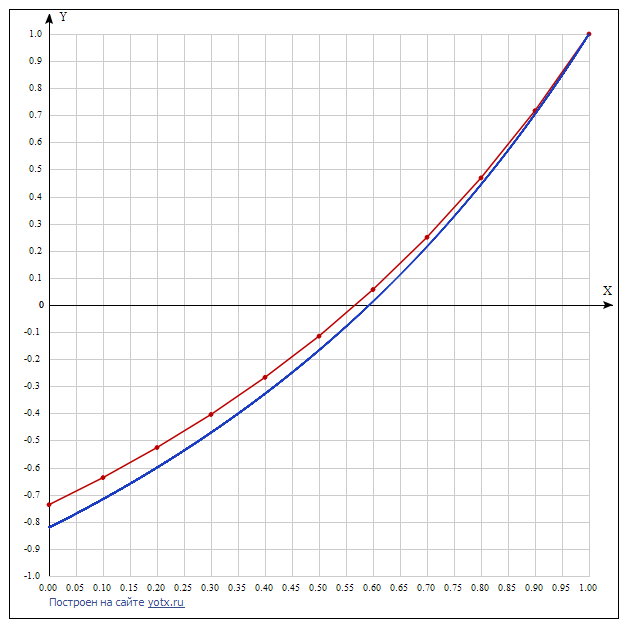
x = 0.200 correct = -0.600 y = -0.526

x = 0.100 correct = -0.716 y = -0.637

x = 0.000 correct = -0.822 y = -0.737

**Погрешность:**

**0.0532627564**



Синяя – аналитическое решение

Красная – решение методом прогонки

с условиями

Аналитическое решение

Решение методом прогонки на отрезке [0,1] c n = 10:

x = 1.000 correct = 2.000 y = 2.214

x = 0.900 correct = 1.458 y = 1.614

x = 0.800 correct = 1.024 y = 1.134

x = 0.700 correct = 0.686 y = 0.759

x = 0.600 correct = 0.432 y = 0.478

x = 0.500 correct = 0.250 y = 0.277

x = 0.400 correct = 0.128 y = 0.142

x = 0.300 correct = 0.054 y = 0.060

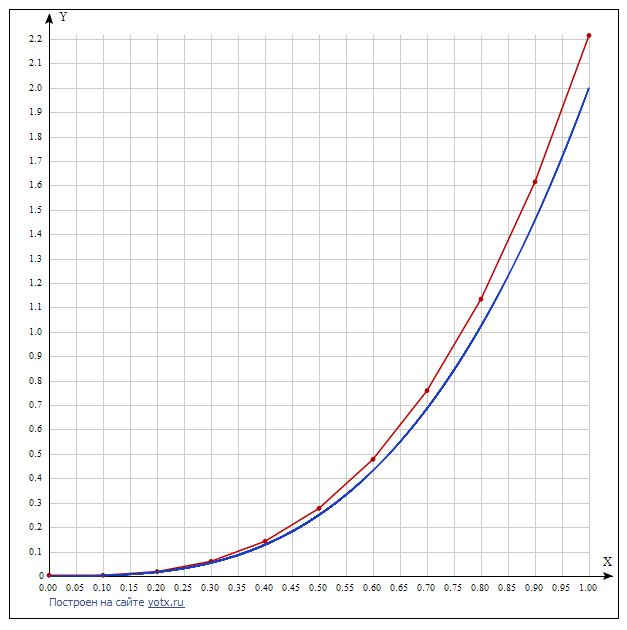
x = 0.200 correct = 0.016 y = 0.018

x = 0.100 correct = 0.002 y = 0.003

x = 0.000 correct = 0.000 y = 0.003

**Погрешность:**

**0.0650973213**



Синяя – аналитическое решение

Красная – решение методом прогонки

**Исходные коды программ**

**Метод Рунге-Кутта**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

double

func\_sing(double x, double y)

{

return - y - x \* x;

}

double

func\_sys\_1(double x, double u, double v)

{

return sin(1.4 \* u \* u) - x + v;

}

double

func\_sys\_2(double x, double u, double v)

{

return x + u - 2.2 \* v \* v + 1;

}

// функции для проверки решения диф. уравнения

double

correct(double x)

{

return 0;

}

double

correct\_1(double x)

{

return 0;

}

double

correct\_2(double x)

{

return 0;

}

// метод Рунге-Кутта 2-го порядка для одного уравнения

void

RungeKutt\_single\_2(double x\_0, double y\_0, double h, int n, double (\*func)())

{

double pr = 0;

double y\_cur;

double y\_prev = y\_0;

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

double x\_prev = x\_0 + i \* h;

double x\_cur = x\_0 + (i + 1) \* h;

double k1 = h \* func(x\_prev, y\_prev);

double k2 = h \* func(x\_cur, y\_prev + k1);

y\_cur = y\_prev + (k1 + k2)/ 2;

// вывод решения диф. уравнения в точке

printf("x = %.3lf ", x\_cur);

printf("correct = %.3lf ", correct(x\_cur));

printf("y = %.3lf\n", y\_cur);

pr += fabs(correct(x\_cur) - y\_cur);

y\_prev = y\_cur;

}

// вывод погрешности метода

printf("%.10lf\n", pr / n);

}

// метод Рунге-Кутта 4-го порядка для одного уравнения

void

RungeKutt\_single\_4(double x\_0, double y\_0, double h, int n, double (\*func)())

{

double pr = 0;

double y\_cur;

double y\_prev = y\_0;

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

double x\_prev = x\_0 + i \* h;

double x\_cur = x\_0 + (i + 1) \* h;

double k1 = h \* func(x\_prev, y\_prev);

double k2 = h \* func(x\_prev + h / 2, y\_prev + k1 / 2);

double k3 = h \* func(x\_prev + h / 2, y\_prev + k2 / 2);

double k4 = h \* func(x\_prev + h, y\_prev + k3);

y\_cur = y\_prev + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;

printf("x = %.3lf ", x\_cur);

printf("correct = %.3lf ", correct(x\_cur));

printf("y = %.3lf\n", y\_cur);

pr += fabs(correct(x\_cur) - y\_cur);

y\_prev = y\_cur;

}

printf("%.10lf\n", pr / n);

}

// метод Рунге-Кутта 2-го порядка для системы из 2-х уравнений

void

RungeKutt\_system\_2(double x\_0, double u\_0, double v\_0, double h, int n, double (\*f1)(), double (\*f2)())

{

double pru;

double prv;

double u\_cur;

double v\_cur;

double u\_prev = u\_0;

double v\_prev = v\_0;

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

double x\_prev = x\_0 + i \* h;

double x\_cur = x\_0 + (i + 1) \* h;

double k1 = h \* f1(x\_prev, u\_prev, v\_prev);

double m1 = h \* f2(x\_prev, u\_prev, v\_prev);

double k2 = h \* f1(x\_cur, u\_prev + k1, v\_prev + m1);

double m2 = h \* f2(x\_cur, u\_prev + k1, v\_prev + m1);

u\_cur = u\_prev + (k1 + k2)/ 2;

v\_cur = v\_prev + (m1 + m2)/ 2;

printf("x = %.3lf ", x\_cur);

printf("cor1 = %.3lf ", correct\_1(x\_cur));

printf("u = %.3lf ", u\_cur);

printf("cor2 = %.3lf ", correct\_2(x\_cur));

printf("v = %.3lf\n", v\_cur);

pru += fabs(correct\_1(x\_cur) - u\_cur);

prv += fabs(correct\_2(x\_cur) - v\_cur);

u\_prev = u\_cur;

v\_prev = v\_cur;

}

printf("%.10lf\n", pru / n);

printf("%.10lf\n", prv / n);

}

// метод Рунге-Кутта 4-го порядка для системы из 2-х уравнений

void

RungeKutt\_system\_4(double x0, double u\_0, double v\_0, double h, int n, double (\*f1)(), double (\*f2)())

{

double pru;

double prv;

double u\_cur;

double v\_cur;

double u\_prev = u\_0;

double v\_prev = v\_0;

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

double x\_prev = x0 + i \* h, x\_cur = x0 + (i + 1) \* h;

double k1 = h \* f1(x\_prev, u\_prev, v\_prev);

double m1 = h \* f2(x\_prev, u\_prev, v\_prev);

double k2 = h \* f1(x\_prev + h / 2, u\_prev + k1 / 2, v\_prev + m1 / 2);

double m2 = h \* f2(x\_prev + h / 2, u\_prev + k1 / 2, v\_prev + m1 / 2);

double k3 = h \* f1(x\_prev + h / 2, u\_prev + k2 / 2, v\_prev + m2 / 2);

double m3 = h \* f2(x\_prev + h / 2, u\_prev + k2 / 2, v\_prev + m2 / 2);

double k4 = h \* f1(x\_prev + h, u\_prev + k3, v\_prev + m3);

double m4 = h \* f2(x\_prev + h, u\_prev + k3, v\_prev + m3);

u\_cur = u\_prev + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;

v\_cur = v\_prev + (m1 + 2 \* m2 + 2 \* m3 + m4) / 6;

printf("x = %.3lf ", x\_cur);

printf("cor1 = %.3lf ", correct\_1(x\_cur));

printf("u = %.3lf ", u\_cur);

printf("cor2 = %.3lf ", correct\_2(x\_cur));

printf("v = %.3lf\n", v\_cur);

pru += fabs(correct\_1(x\_cur) - u\_cur);

prv += fabs(correct\_2(x\_cur) - v\_cur);

u\_prev = u\_cur;

v\_prev = v\_cur;

}

printf("%.10lf\n", pru / n);

printf("%.10lf\n", prv / n);

}

int

main(void)

{

double a, x\_0, y\_0, u\_0, v\_0;

int n;

char s[10];

printf("enter n:\n");

scanf("%d", &n);

printf("enter a:\n");

scanf("%lf", &a);

printf("enter method name:\n");

scanf("%s", s);

double h = a / n;

if (strcmp(s, "sing\_2") == 0) {

printf("enter x\_0:\n");

scanf("%lf", &x\_0);

printf("enter y\_0:\n");

scanf("%lf", &y\_0);

printf("results:\n");

RungeKutt\_single\_2(x\_0, y\_0, h, n, func\_sing);

}

if (strcmp(s, "sing\_4") == 0) {

printf("enter x\_0:\n");

scanf("%lf", &x\_0);

printf("enter y\_0:\n");

scanf("%lf", &y\_0);

printf("results:\n");

RungeKutt\_single\_4(x\_0, y\_0, h, n, func\_sing);

}

if (strcmp(s, "sys\_2") == 0) {

printf("enter x\_0:\n");

scanf("%lf", &x\_0);

printf("enter u\_0:\n");

scanf("%lf", &u\_0);

printf("enter v\_0:\n");

scanf("%lf", &v\_0);

printf("results:\n");

RungeKutt\_system\_2(x\_0, u\_0, v\_0, h, n, func\_sys\_1, func\_sys\_2);

}

if (strcmp(s, "sys\_4") == 0) {

printf("enter x\_0:\n");

scanf("%lf", &x\_0);

printf("enter u\_0:\n");

scanf("%lf", &u\_0);

printf("enter v\_0:\n");

scanf("%lf", &v\_0);

printf("results:\n");

RungeKutt\_system\_4(x\_0, u\_0, v\_0, h, n, func\_sys\_1, func\_sys\_2);

}

return 0;

}

**Метод конечных разностей для решения краевой задачи с помощью метода прогонки**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

//коэффициент при y"

double p(double x)

{

return 3;

}

//коэффициент при y'

double q(double x)

{

return - 1 / x;

}

//правая часть уравнения f(x)

double f(double x)

{

return x + 1;

}

//аналитическое решение для проверки

double correct(double x)

{

return 0;

}

int main(void)

{

// погрешность

double pr = 0;

int n;

printf("n = ");

scanf("%d", &n);

double a, b;

printf("a = ");

scanf("%le", &a);

printf("b = ");

scanf("%le", &b);

// ввод коэффициентов

double sigma\_1, sigma\_2, gamma\_1, gamma\_2, delta\_1, delta\_2;

printf("sigma\_1 = ");

scanf("%le", &sigma\_1);

printf("gamma\_1 = ");

scanf("%le", &gamma\_1);

printf("delta\_1 = ");

scanf("%le", &delta\_1);

printf("sigma\_2 = ");

scanf("%le", &sigma\_2);

printf("gamma\_2 = ");

scanf("%le", &gamma\_2);

printf("delta\_2 = ");

scanf("%le", &delta\_2);

double alpha[n], beta[n];

double h = (b - a) / n;

double x = a + h;

alpha[0] = - gamma\_1 / (sigma\_1 \* h - gamma\_1);

beta[0] = delta\_1 / (sigma\_1 - gamma\_1 / h);

// прогонка

for (int i = 1; i < n; i++, x += h) {

// вычисдение прогоночных коэфициентов

double k\_1 = 1 / (h \* h) + p(x) / (2 \* h);

double k\_2 = 1 / (h \* h) - p(x) / (2 \* h);

double k\_3 = -2 / (h \* h) + q(x);

alpha[i] = - k\_1 / (k\_2 \* alpha[i - 1] + k\_3);

beta[i] = (f(x) - beta[i - 1] \* k\_2) / (k\_2 \* alpha[i - 1] + k\_3);

}

printf("solution:\n");

double y = (delta\_2 \* h + gamma\_2 \* beta[n - 1]) / (sigma\_2 \* h + gamma\_2 - gamma\_2 \* alpha[n - 1]);

printf("x = %.3lf correct = %.3lf y = %.3lf\n", x, correct(x), y);

// подсчет погрешности

pr += fabs(y - correct(x));

// вывод решения системы конечно-разностных уравнений в виде y\_next = a\_i\*y\_prev + b\_i

for (int i = n - 1; i >= 0; i--) {

x -= h;

y = alpha[i] \* y + beta[i];

pr += fabs(y - correct(x));

printf("x = %.3lf correct = %.3lf y = %.3lf\n", x, correct(x), y);

}

// вывод погрешности

printf("%.10lf\n", pr / n);

return 0;

}

**Выводы**

1. В ходе данной практической работы были реализованы методы Рунге-Кутта 2-го и 4-го порядков для решения задачи Коши с дифференциальным уравнением первого порядка. Также была подсчитана и наглядно продемонстрирована на графиках погрешность этих методов.

В ходе тестирования было выявлено, что для более точного нахождения решения предпочтительнее использовать метод 4-го порядка.

1. В ходе 2-й части практической работы был реализован метод конечных разностей решения краевой задачи, в котором система конечно-разностных уравнений была решена методом прогонки.

В ходе тестирования также была выявлена и наглядно продемонстрирована на графиках погрешность данного метода.