# МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Нижегородский государственный технический университет

им. Р.Е. Алексеева» (НГТУ)

Кафедра: «Цифровая экономика»

Дисциплина: «Численные методы»

**Лабораторная работа №3**

**«Методы решения обыкновенных дифференциальных уравнений»**

Выполнил:

студент 3-го курса группы 21-САИ

Краличев Игорь Евгеньевич





Проверил:

д.ф.м.н., проф. Катаева Лилия Юрьевна

11.10.2023

Подпись преподавателя:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород, 2023

Оглавление

[Постановка задачи 4](#_Toc149764630)

[Задание №1 4](#_Toc149764631)

[Задание №2 4](#_Toc149764632)

[Задание №3 4](#_Toc149764633)

[Методы решений и инструменты 5](#_Toc149764634)

[Инструменты 5](#_Toc149764635)

[Методы решений 6](#_Toc149764636)

[Метод Рунге-Кутта 6](#_Toc149764637)

[Правило Рунге 6](#_Toc149764638)

[Метод Адамса 7](#_Toc149764639)

[Метод прогонки 8](#_Toc149764640)

[Проекционный метод 9](#_Toc149764641)

[Таблица идентификаторов 10](#_Toc149764642)

[Метод Рунге-Кутта 10](#_Toc149764643)

[Метод прогонки 10](#_Toc149764644)

[Проекционный метод 11](#_Toc149764645)

[Ручной счёт 13](#_Toc149764646)

[Метод Рунге-Кутта 13](#_Toc149764647)

[Метод прогонки 13](#_Toc149764648)

[Проекционный метод 13](#_Toc149764649)

[Реализация задачи в excel 14](#_Toc149764650)

[Метод Рунге-Кутта 14](#_Toc149764651)

[Метод прогонки 15](#_Toc149764652)

[Проекционный метод 16](#_Toc149764653)

[Реализация задачи в Mathcad 15 17](#_Toc149764654)

[Метод Рунге-Кутта 17](#_Toc149764655)

[Метод прогонки 19](#_Toc149764656)

[Проекционный метод 20](#_Toc149764657)

[Реализация задачи в C++ 21](#_Toc149764658)

[Основа и выбор пользователя 21](#_Toc149764659)

[Метод Рунге-Кутта 21](#_Toc149764660)

[Метод прогонки 23](#_Toc149764661)

[Проекционный метод 25](#_Toc149764662)

[Реализация задачи в Java 29](#_Toc149764663)

[Основа и выбор пользователя 29](#_Toc149764664)

[Метод Рунге-Кутта 29](#_Toc149764665)

[Метод прогонки 31](#_Toc149764666)

[Проекционный метод 33](#_Toc149764667)

[Результат и анализ всех реализаций 36](#_Toc149764668)

[Список литературы 37](#_Toc149764669)

# Постановка задачи

Вариант 10

## Задание №1

1. Найти решение задачи Коши на отрезке методом Рунге-Кутта с шагом Решение в точках 0.4 и 0.5 найти методом Адамса. Оценить погрешность полученного решения, применяя правило Рунге.

;

## Задание №2

2. Решить краевую задачу

Методом прогонки по равномерной сетке

Для аппроксимации первой производной воспользоваться формулой

## Задание №3

3. Решить краевую задачу

Одним из проекционных методов. Данные взять из задачи 2.

# Методы решений и инструменты

## Инструменты

При решении лабораторной работы были использованы следующие инструменты:

1. Mathcad 15 (версия: M050 [], разрядность: x64);
2. Eclipse IDE (версия: 2022-09 (4.25.0), разрядность: x64);
3. Visual Studio IDE (версия: 17.7.3, разрядность: x64).

Mathcad 15 — это программное обеспечение для математического моделирования и анализа технических и научных данных. Оно предоставляет удобную среду для создания и решения математических выражений и уравнений, а также выполнения численных и символьных вычислений.

Eclipse IDE — это интегрированная среда разработки (IDE), используемая в компьютерном программировании. Оно содержит базовое рабочее пространство и расширяемую систему подключаемых модулей для настройки среды. Это вторая по популярности среда IDE для разработки на Java, и до 2016 года она была самой популярной. Eclipse написан в основном на Java и его основное применение заключается в разработке приложений Java, но он также может быть использован для разработки приложений на других языках программирования с помощью плагинов.

Visual Studio IDE- это интегрированная среда разработки (IDE) от компании Microsoft, предназначенная для разработки программного обеспечения. В Visual Studio предоставляются различные инструменты и функциональные возможности, упрощающие процесс разработки, отладки и тестирования приложений.

Visual Studio поддерживает множество языков программирования, включая C++, C#, Visual Basic, F#, JavaScript и другие. Для разработки на C++ в Visual Studio используется компилятор Microsoft C++, который обеспечивает мощные возможности компиляции и оптимизации кода.

Visual Studio обеспечивает разработчиков на C++ всеми необходимыми инструментами для создания высококачественного программного обеспечения. Отличительной чертой Visual Studio является его обширная функциональность и поддержка различных платформ и технологий, что делает его одним из популярных выборов для разработки на C++.

## Методы решений

### Метод Рунге-Кутта

Пусть известно решение дифференциального уравнения в точке . Решение в точке будем искать в виде:

Параметры формулы Рунге-Кутта выбирают таким образом, чтобы погрешность приближённого решения, т.е. разность между точным значением и приближённым , была как можно меньшей. Обозначим погрешность на шаге через

Число шагов N, которое необходимо сделать до произвольной точки x, обратно пропорционально шагу h, т.е.

Наиболее распространена в практике следующая формула Рунге-Кутта, имеющая погрешность четвёртого порядка:

### Правило Рунге

Обозначим приближённое решение в точке , найденное с шагом h через , а через - приближённое решение, найденное с шагом 2h.

Справедливы приближённые равенства

Приравнивая правые части (1.5), найдём

Отсюда

### Метод Адамса

Особенности метода Адамса:

1. метод применяется для равномерной сетки т.е.

,

1. для нахождения приближённого решения в точке требуется знание приближённого решения в предыдущих точках, т.е. в

Точки образуют начальный отрезок. Решение на начальном отрезке может быть найдено каким-нибудь другим методом, например, методом Рунге-Кутта.

Пусть известно решение в точках

Проинтегрируем исходное уравнение на отрезке

или

Построим для подынтегральной функции интерполяционный многочлен Если интерполяционный многочлен построен по значениям функции в узлах то получим экстраполяционную формулу Адамса. Если интерполяционный многочлен построен по значениям в узлах , то получим интерполяционную формулу Адамса.

Так как значение интерполяционного полинома вычисляем на отрезке то в качестве интерполяционной формулы выберем формулу Ньютона для интерполирования назад.

Ниже записаны формулы Адамса, имеющие погрешность четвёртого порядка.

Экстраполяционная формула Адамса:

**

Интерполяционная формула Адамса:

, где

Следует заметить, что интерполяционная формула Адамса представляет собой уравнение относительно . Один из методов решения этого уравнения – метод последовательных приближений

, где .

Практическая оценка погрешности приближённого решения может быть получена с помощью правила Рунге.

### Метод прогонки

Пусть дано дифференциальное уравнение

требуется найти его решение, удовлетворяющее дифференциальному уравнению внутри отрезка и краевым условиям

Пусть задана краевая задача в виде

Построим равномерную сетку на отрезке

, , .

Заменив вторую производную в точке разностным отношением

а первую производную

Получим систему разностных уравнений

где – приближённые значения решения в точке .

В методе прогонки решение системы ищем в виде

где – прогоночные коэффициенты.

Сравнивая и краевое условие , получим

Если в

подставить вместо правую часть выражения и полученное уравнение записать, то получим рекуррентные формулы для :

Вычисления по методу прогонки проводят в два этапа. Первый этап называется прямой прогонкой и состоит из вычисления прогоночных коэффициентов . Второй этап называется обратной прогонкой и заключается в вычислении для значений

### Проекционный метод

В проекционных методах приближённое аналитическое решение ищут в виде:

где функции , удовлетворяют следующим условиям :

Эти ограничения необходимы для того, чтобы функция удовлетворяла граничным условиям

Система функций , должна быть линейно-независимой и полной в классе дважды непрерывно дифференцируемых функций. Этим условиям удовлетворяют, например, система степенных функций

Если приближённое решение подставить в правую часть дифференциального уравнения, то в общем случае

Разность называется невязкой.

Метод Коллокаций

Выберем на систему точек

и потребуем, чтобы

Эти соотношения представляют собой систему n линейных алгебраических уравнений для определения неизвестных параметров

# Таблица идентификаторов

## Метод Рунге-Кутта

Таблица 1

Таблица идентификаторов для метода Рунге-Кутта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Java | Комментарий |
| i | i | i | i | i | Итерации |
| j |  | j | j | j | Переменные для прохода итераций |
| n | n | n | n | n | n из дано |
| h | h | h | h | h | Шаг |
| y | y | y | y[i] | y[i] | Точки функции |
| x | x | x | x[i] | x[i] | Точки функции |
| f | f | f(x) | f[x] | f[x] | Данная функция f от x |
| k1 | k1 | k1 | k1 | k1 | Формулы Рунге-Кутта |
| k2 | k2 | k2 | k2 | k2 | Формулы Рунге-Кутта |
| k3 | k3 | k3 | k3 | k3 | Формулы Рунге-Кутта |
| k4 | k4 | k4 | k4 | k4 | Формулы Рунге-Кутта |

## Метод прогонки

Таблица 2

Таблица идентификаторов для метода прогонки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Java | Комментарий |
| i | i | i | i | i | Итерации |
| j |  | j | j | j | Переменные для прохода итераций |
| a | a | a | a | a | Первый x из дано |
| b | b | b | b | b | Последний x из дано |
| n | n | n | n | n | n из дано |
| h | h | h | h | h | Шаг |
| y | y | y | y[i] | y[i] | Точки функции |
| x | x | x | x[i] | x[i] | Точки функции |
| A | A | A(x) | A[x] | A[x] | Данная функция а |
| B | B | B(x) | B[x] | B[x] | Данная функция b |
| C | C | C(x) | C[x] | C[x] | Данная функция c |
| f | f | f(x) | f[x] | f[x] | Данная функция f |
| ai | ai | ai(x) | ai[x] | ai[x] | Изменённая функция a |
| bi | bi | bi(x) | bi[x] | bi[x] | Изменённая функция b |
| ci | ci | ci(x) | ci[x] | ci[x] | Изменённая функция c |
| L | L | L | L[i] | L[i] | Коэффициент L |
| K | K | K | K[i] | K[i] | Коэффициент K |

## Проекционный метод

Таблица 3

Таблица идентификаторов для проекционного метода

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Java | Комментарий |
| i | i | i | i | i | Итерации |
| j |  | j | j | j | Переменные для прохода итераций |
| a | a | a | a | a | Первый x из дано |
| b | b | b | b | b | Последний x из дано |
| n | n | n | n | n | n из дано |
| h | h | h | h | h | Шаг |
| y | y | y | y[i] | y[i] | Точки функции |
| x | x | x | x[i] | x[i] | Точки функции |
| A | A | A(x) | A[x] | A[x] | Данная функция а |
| B | B | B(x) | B[x] | B[x] | Данная функция b |
| C | C | C(x) | C[x] | C[x] | Данная функция c |
| f | f | f(x) | f[x] | f[x] | Данная функция f |
| c | c | c | c | c | Коэффициенты для функции u(x) |
| d | d | d | d | d | Коэффициенты для функции u(x) |
| u | u(x) | u(x) | u[x] | u[x] | Функция от x для функции L. Проекция |
| a1 | a1 | a1 | a1 | a1 | Коэффициенты для решения L(u) |
| b1 | b1 | b1 | b1 | b1 | Коэффициенты для решения L(u) |
| c1 | c1 | c1 | c1 | c1 | Коэффициенты для решения L(u) |
| L | L(u(x)) | L(u(x)) | lu[x] | lu[x] | Функция от u(x) |

# Ручной счёт

## Метод Рунге-Кутта

h=0.1

Шаг 0:

x0=0, y0=0, y0’=0

Шаг 1:

x1=0.1

Ищем наклоны точек в интервале [0;0.1]:

Тогда приближённое значение на этом этапе будет равно

Решение задачи Коши на этом этапе будет равно

Шаг 2:

x2=0.2

Ищем наклоны точек в интервале [0.1;0.2]:

Тогда приближённое значение на этом этапе будет равно

Решение задачи Коши на этом этапе будет равно

Шаг 3:

x3=0.3

Ищем наклоны точек в интервале [0.2;0.3]:

Тогда приближённое значение на этом этапе будет равно

Решение задачи Коши на этом этапе будет равно

Метод Адамса

Экстраполяционная формула для x4=0.4

Решение задачи Коши на этом этапе будет равно

Экстраполяционная формула для x5=0.5

Решение задачи Коши на этом этапе будет равно

Интерполяционная формула для x4=0.4

Решение задачи Коши на этом этапе будет равно

Интерполяционная формула для x5=0.5

Решение задачи Коши на этом этапе будет равно

Правило Рунге для оценки погрешности

## Метод прогонки

,

,

,

,

,

,

Прямой ход

Шаг 0:

,

Вычисляем шаг:

,

Значения функций A(x), B(x), C(x) в точке :

,

,

,

Расчёт значений для нахождения прогоночных коэффициентов:

,

Подсчёт прогоночных коэффициентов:

Решение f(x) в точке :

Шаг 1:

,

Значения функций A(x), B(x), C(x) в точке :

,

,

,

Расчёт коэффициентов для нахождения прогоночных коэффициентов

,

Подсчёт прогоночных коэффициентов:

Решение f(x) в точке :

Шаги со 2 по 11 представлены в таблице 1

Таблица 1

Таблица с результатами расчётов прямого хода для метода прогонки

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | xi | A(xi) | B(xi) | C(xi) | ai | bi | ci | Li | Ki | f(xi) |
| 0 | 0.310000 | 1.208064 | 2.088012 | -0.096100 | 112.477479 | -245.198538 | 132.624959 | 0.000000 | 1.300000 | 1.426793 |
| 1 | 0.413636 | 1.274196 | 2.078689 | -0.171095 | 118.634789 | -257.498201 | 138.692317 | 0.538615 | 0.593396 | 1.896608 |
| 2 | 0.517273 | 1.338889 | 2.066740 | -0.267571 | 124.658020 | -269.525835 | 144.600244 | 0.714487 | 0.356131 | 2.308368 |
| 3 | 0.620909 | 1.402268 | 2.052195 | -0.385528 | 130.558974 | -281.305356 | 150.360854 | 0.799696 | 0.235013 | 2.685701 |
| 4 | 0.724545 | 1.464442 | 2.035094 | -0.524966 | 136.347732 | -292.857301 | 155.984602 | 0.848570 | 0.159709 | 3.041108 |
| 5 | 0.828182 | 1.525505 | 2.015482 | -0.685885 | 142.032993 | -304.199508 | 161.480630 | 0.879167 | 0.106944 | 3.381979 |
| 6 | 0.931818 | 1.585537 | 1.993414 | -0.868285 | 147.622325 | -315.347628 | 166.857017 | 0.899193 | 0.066852 | 3.713026 |
| 7 | 1.035455 | 1.644610 | 1.968946 | -1.072166 | 153.122374 | -326.315518 | 172.120978 | 0.912485 | 0.034584 | 4.037423 |
| 8 | 1.139091 | 1.702788 | 1.942146 | -1.297528 | 158.539008 | -337.115552 | 177.279015 | 0.921164 | 0.007511 | 4.357398 |
| 9 | 1.242727 | 1.760125 | 1.913085 | -1.544371 | 163.877448 | -347.758863 | 182.337044 | 0.926505 | -0.015887 | 4.674570 |
| 10 | 1.346364 | 1.816673 | 1.881842 | -1.812695 | 169.142362 | -358.255542 | 187.300485 | 0.929327 | -0.036526 | 4.990143 |
| 11 | 1.450000 | 1.872476 | 1.848499 | -2.102500 | 174.337947 | -368.614791 | 192.174345 |  |  | 5.305032 |

Обратный ход:

Шаг 0:

Шаг 1:

Шаг 2:

Шаг 3:

Шаг 4:

Шаг 5:

Шаг 6:

Шаг 7:

Шаг 8:

Шаг 9:

Шаг 10:

Шаг 10:

## Проекционный метод

,

,

,

,

,

,

Шаг 0:

,

Вычисляем шаг

,

Решаем систему уравнений:

Создаём обратную матрицу:

Для нахождения c и d перемножаем матрицы:

Отсюда ,

Шаг 0:

Значения функций A(x), B(x), C(x) в точке :

,

,

,

Расчёт коэффициентов для нахождения прогоночных коэффициентов

,

Подсчёт прогоночных коэффициентов:

Решение f(x) в точке :

Методом подбора находим коэффициенты a1=-4.41, b1=5.25, c1=0.32

Делаем проекцию

Шаг 1:

,

Значения функций A(x), B(x), C(x) в точке :

,

,

,

Расчёт коэффициентов для нахождения прогоночных коэффициентов

,

Подсчёт прогоночных коэффициентов:

Решение f(x) в точке :

Делаем проекцию

Шаги со 2 по 11 представлены в таблице 2

Таблица 2

Таблица с данными проекционного метода

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | xi | f(xi) | A(xi) | B(xi) | C(xi) | fu(xi) | Lu(xi) | |Lu(xi)- f(xi)| |
| 0 | 0.310000 | 1.426793 | 1.79147 | 1.896084 | -1.69 | 4,849099 | 1,51326 | 0,08647 |
| 1 | 0.413636 | 1.896608 | 1.746618 | 1.920168 | -1.48397 | 4,599635 | 1,90343 | 0,00682 |
| 2 | 0.517273 | 2.308368 | 1.701268 | 1.942881 | -1.29132 | 4,34902 | 2,28431 | 0,02406 |
| 3 | 0.620909 | 2.685701 | 1.655393 | 1.964183 | -1.11207 | 4,096643 | 2,65582 | 0,02989 |
| 4 | 0.724545 | 3.041108 | 1.608966 | 1.984038 | -0.9462 | 3,841731 | 3,01788 | 0,02323 |
| 5 | 0.828182 | 3.381979 | 1.561958 | 2.002415 | -0.79372 | 3,583295 | 3,37045 | 0,01153 |
| 6 | 0.931818 | 3.713026 | 1.514336 | 2.019281 | -0.65463 | 3,32004 | 3,71352 | 0,00050 |
| 7 | 1.035455 | 4.037423 | 1.466063 | 2.03461 | -0.52893 | 3,05024 | 4,04711 | 0,00968 |
| 8 | 1.139091 | 4.357398 | 1.417099 | 2.048374 | -0.41661 | 2,771523 | 4,37124 | 0,01384 |
| 9 | 1.242727 | 4.674570 | 1.367398 | 2.060551 | -0.31769 | 2,480522 | 4,68601 | 0,01144 |
| 10 | 1.346364 | 4.990143 | 1.316911 | 2.071121 | -0.23215 | 2,17224 | 4,99152 | 0,00138 |
| 11 | 1.450000 | 5.305032 | 1.26558 | 2.080067 | -0.16 | 1,838824 | 5,28794 | 0,01709 |

# Реализация задачи в excel

## Метод Рунге-Кутта

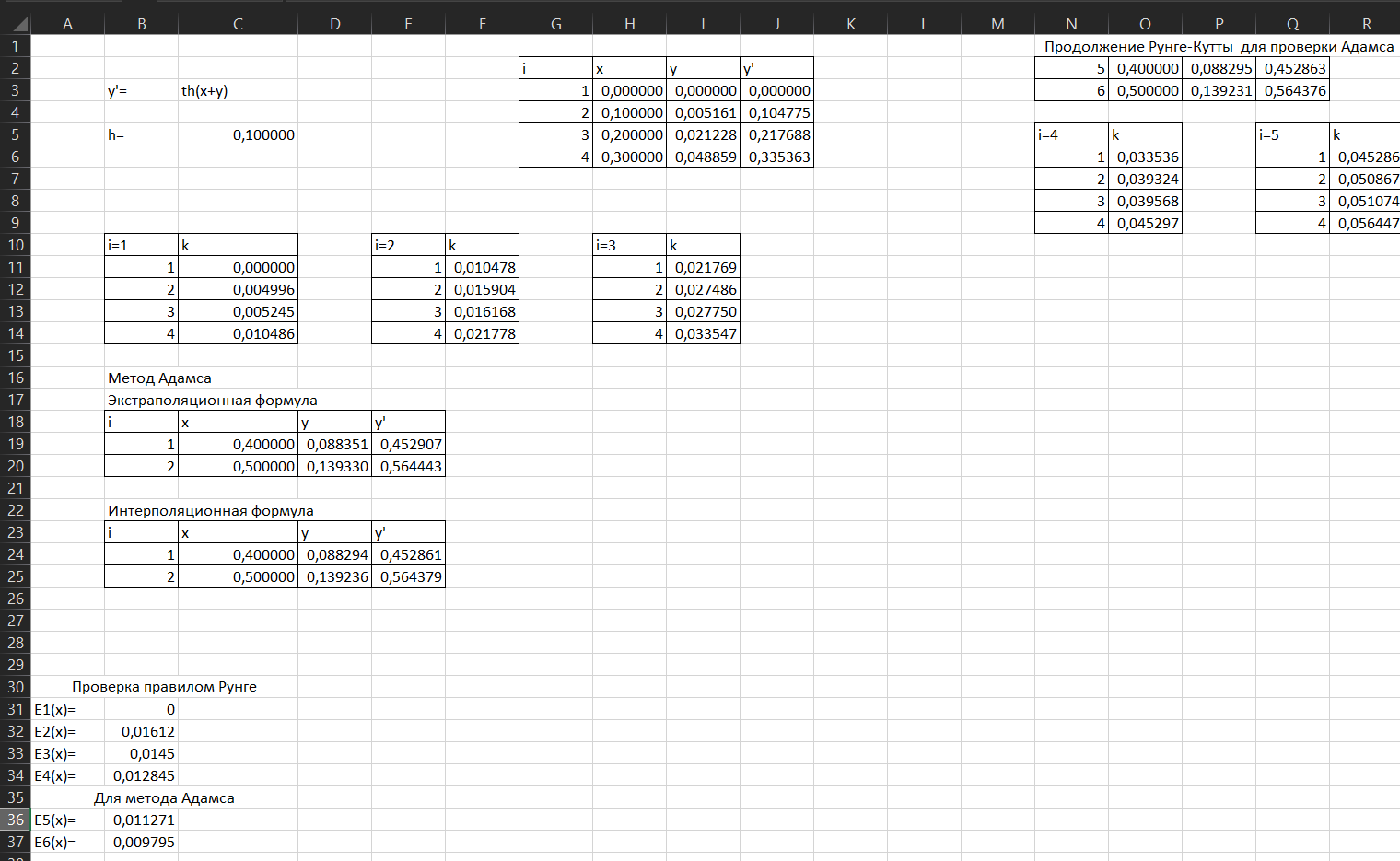


Рисунок 1. - Метод Рунге-Кутта

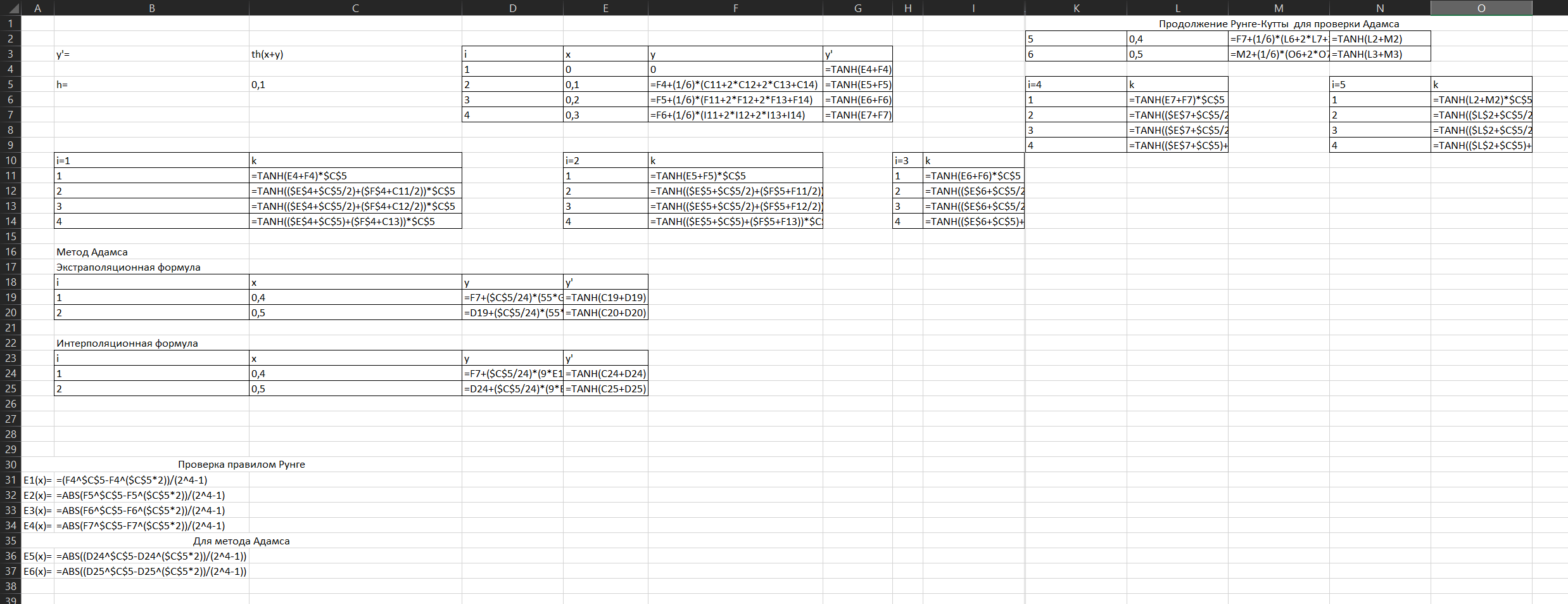


Рисунок 2. - Метод Рунге-Кутта

## Метод прогонки

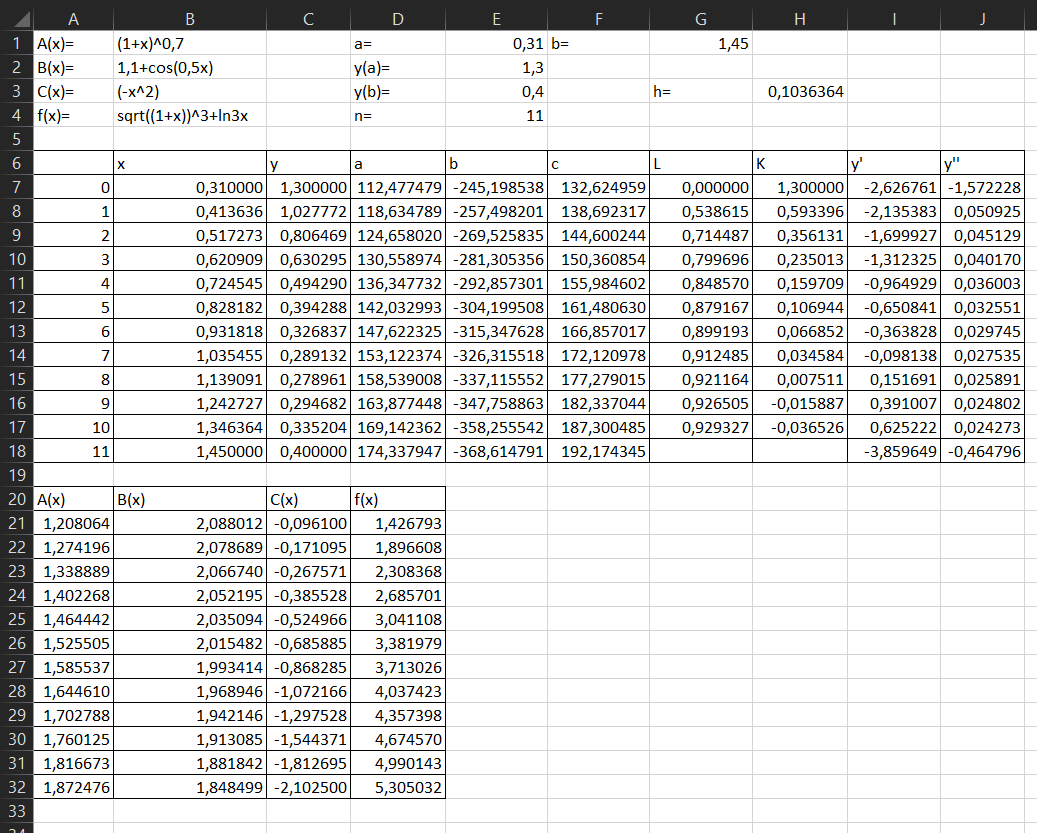


Рисунок 3. - Метод прогонки

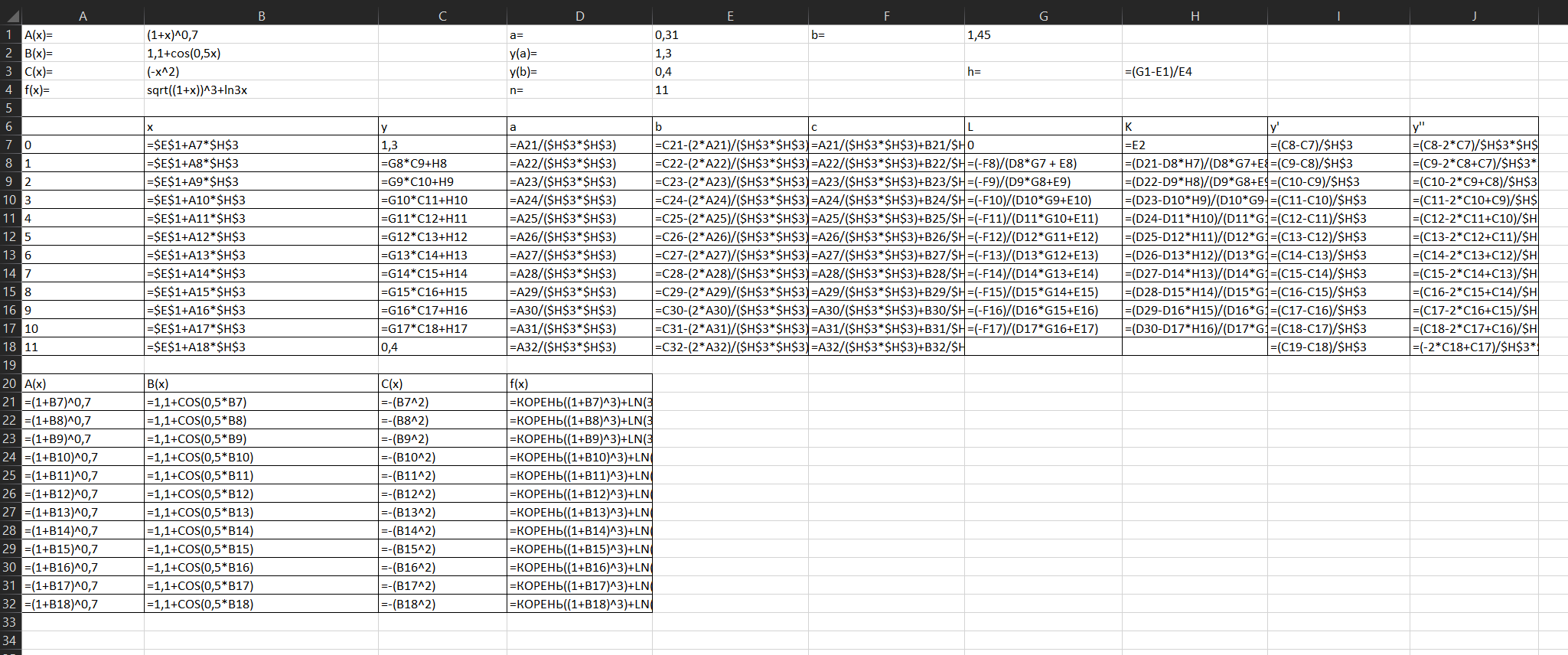


Рисунок 4. - Метод прогонки

## Проекционный метод



Рисунок 5. - Проекционный метод



Рисунок 6. - Проекционный метод

# Реализация задачи в Mathcad 15

## Метод Рунге-Кутта

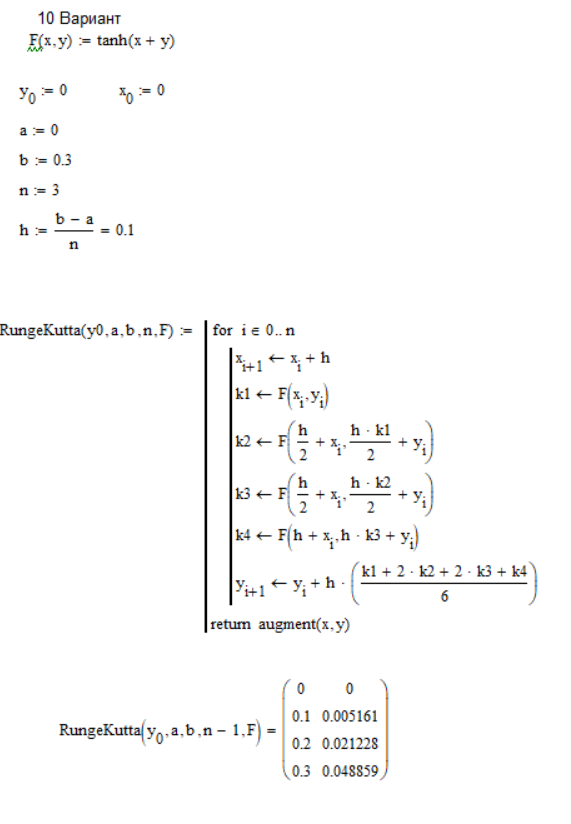


Рисунок 7. - Метод Рунге-Кутта

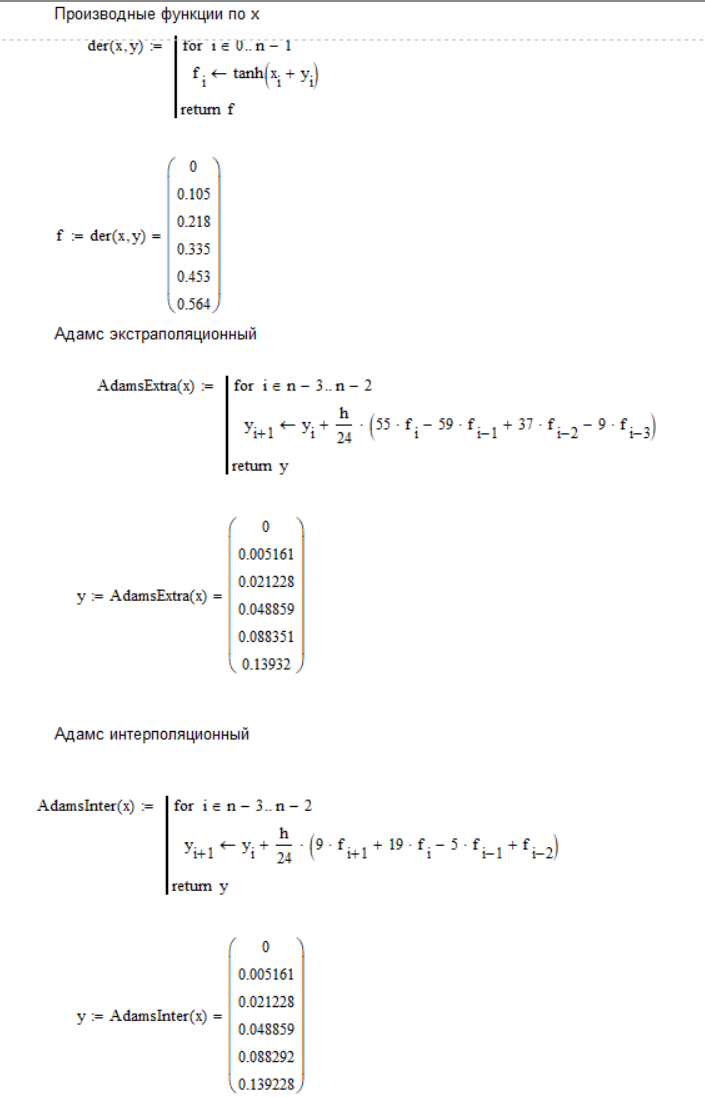


Рисунок 8. – Метод Адамса экстраполяционный и интерполяционный

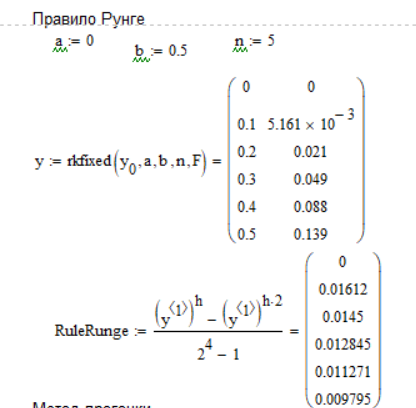


Рисунок 9. – Правило Рунге

## Метод прогонки

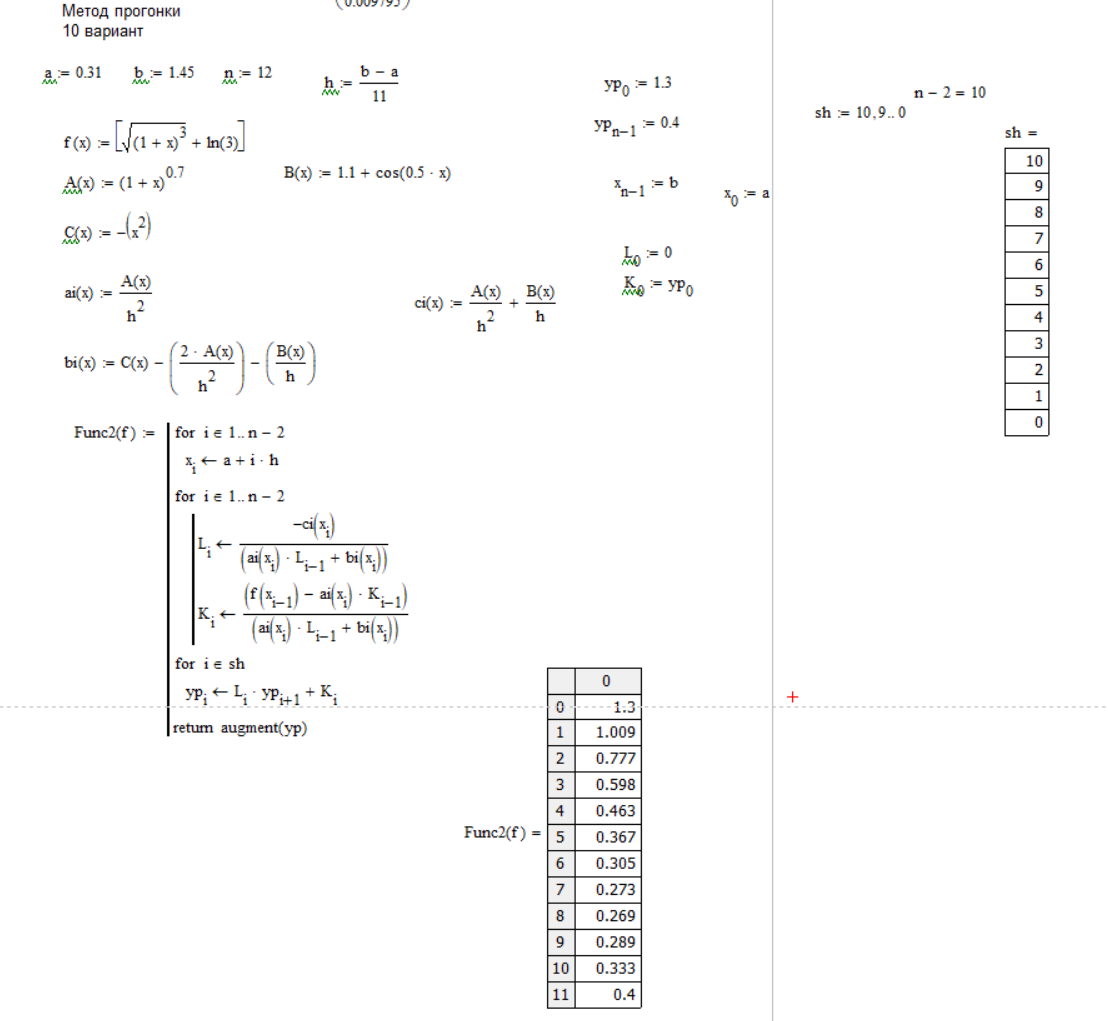


Рисунок 10. - Метод прогонки

# Реализация задачи в C++

## Основа и выбор пользователя

#include <iostream>

#include <math.h>

#include <cmath>

#include <complex>

#include "Runge.h"

#include "RunTh.h"

#include "Proj.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

Runge Runge;

RunTh RunTh;

Proj Proj;

int choice;

int exit = 0;

do {

cout << "Программа решает ОДУ. Выберите метод, которым хотите решить (введите номер метода):\n";

cout << "1.Метод Рунге-Кутта.\n";

cout << "2.Метод прогонки.\n";

cout << "3.Проекционный метод.\n";

cout << "4.Завершение работы.\n";

cout << "Введите число:";

cin >> choice;

// Правильность выбора метода

while (!(choice == 1 || choice == 2 || choice == 3 || choice == 4 )) {

cout << "Выберите числа от 1 до 4!\n";

cin >> choice;

//Проверка на число

while (!cin)

{

cout << "Введите число!\n";

cin.clear();

while (cin.get() != '\n') continue;

cin >> choice;

}

}

switch (choice) {

case 1:

cout << "Вы выбрали метод Рунге-Кутта.\n";

Runge.main();

break;

case 2:

cout << "Вы выбрали метод прогонки.\n";

RunTh.main();

break;

case 3:

cout << "Вы выбрали проекционный метод.\n";

Proj.main();

break;

case 4:

cout << "Завершение работы.\n\n";

cout << "Обратная связь:\n";

cout << "Студент гр. 21-САИ, Краличев Игорь Евгеньевич, ikralichev@list.ru\n";

exit = 1;

break;

}

} while (exit == 0);

}

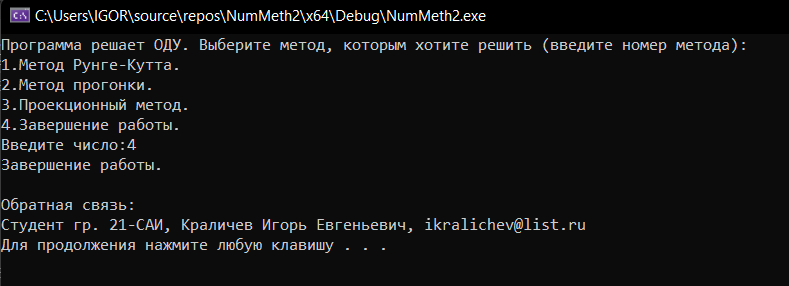


Рисунок 11.-Меню выбора пользователя

## Метод Рунге-Кутта

#include <iostream>

#include <cmath>

using namespace std;

class Runge {

public:

void main() {

int n = 6;

double h = 0.1;

double x[6];

x[0] = 0;

double y[6];

double f[6];

y[0] = 0; //y' = th(x+y)

double k1, k2, k3, k4;

cout << "y[" << 0 << "] = " << y[0] << ", x[" << 0 << "] = " << x[0] << endl;

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

x[i] = 0 + i \* h;

}

cout << "Метод Рунге-Кутта:" << endl;

// метод Рунге-Кутта

for (int i = 0; i < n - 2; i++) {

k1 = tanh(x[i] + y[i]) \* h;

k2 = tanh((x[i] + h / 2) + ((k1) / 2 + y[i])) \* h;

k3 = tanh((x[i] + h / 2) + ((k2) / 2 + y[i])) \* h;

k4 = tanh((x[i] + h) + (y[i] + k3)) \* h;

f[i] = tanh(x[i] + y[i]);

y[i + 1] = y[i] + ((k1 + 2 \* k2 + 2 \* k3 + k4) / 6);

}

for (int i = 0; i < n - 3; i++) {

cout << "y[" << (i + 1) << "] = " << y[i + 1] << ", x[" << (i + 1) << "] = " << x[i + 1] << endl;

}

cout << "Метод Аддамса экстраполяционный:" << endl;

// метод Аддамса экстраполяционный

for (int i = n - 3; i < n - 1; i++) {

f[i] = tanh(x[i] + y[i]);

y[i + 1] = y[i] + (h / 24) \* (55 \* f[i] - 59 \* f[i - 1] + 37 \* f[i - 2] - 9 \* f[i - 3]);

cout << "y[" << (i + 1) << "] = " << y[i + 1] << ", x[" << (i + 1) << "] = " << x[i + 1] << endl;

}

f[5] = tanh(x[5] + y[5]);

cout << "Метод Аддамса интерполяционный:" << endl;

// метод Аддамса интерполяционный

for (int i = n - 3; i < n - 1; i++) {

y[i + 1] = y[i] + (h / 24) \* (9 \* f[i + 1] + 19 \* f[i] - 5 \* f[i - 1] + f[i - 2]);

cout << "y[" << (i + 1) << "] = " << y[i + 1] << ", x[" << (i + 1) << "] = " << x[i + 1] << endl;

}

cout << "Правило Рунге:" << endl;

// правило Рунге

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

cout << "Погрешность y[" << i << "] = " << abs((pow(y[i], h) - pow(y[i], h \* 2)) / (15)) << endl;

}

}

};

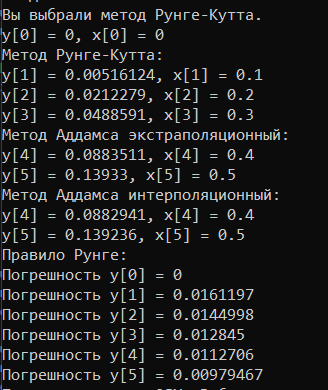


Рисунок 12. - Метод Рунге-Кутта и Адамса

## Метод прогонки

#include <iostream>

#include <cmath>

using namespace std;

class RunTh {

public:

double a = 0.31;

double b = 1.45;

static const int n = 12;

static const int hn = 11;

double h = (b - a) / hn;

double y[n];

double x[n];

double L[n];

double K[n];

double A(double x) {

return pow((1 + x), 0.7);

}

double B(double x) {

return 1.1 + cos(0.5 \* x);

}

double C(double x) {

return -pow(x, 2.0);

}

double f(double x) {

return pow(pow((1.0 + x), 3.0), 0.5) + log(3.0 \* x);

}

double ai(double x) {

return (A(x)) / (h \* h);

}

double bi(double x) {

return C(x) - ((2 \* A(x)) / (h \* h)) - (B(x) / h);

}

double ci(double x) {

return (A(x) / (h \* h)) + (B(x) / h);

}

void main() {

y[0] = 1.3;

y[n - 1] = 0.4;

L[0] = 0;

K[0] = y[0];

x[0] = 0.31;

x[n - 1] = 1.45;

for (int i = 1; i < n - 1; i++) {

x[i] = a + i \* h;

}

// Прогоночные коэффициенты

for (int i = 1; i < n - 1; i++) {

L[i] = (-ci(x[i])) / (ai(x[i]) \* L[i - 1] + bi(x[i]));

K[i] = ((f(x[i - 1]) - ai(x[i]) \* K[i - 1]) / (ai(x[i]) \* L[i - 1] + bi(x[i])));

}

// Вычисление решения

for (int i = n - 2; i > -1; i--) {

y[i] = L[i] \* y[i + 1] + K[i];

}

cout << "Решение методом прогонки:" << endl;

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

cout << "y[" << i << "] = " << y[i] << endl;

}

}

};

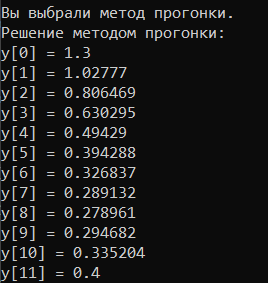


Рисунок 13. - Метод прогонки

## Проекционный метод

#include <iostream>

#include <cmath>

using namespace std;

class Proj {

public:

double A(double x) {

return pow((1 + x), 0.7);

}

double B(double x) {

return 1.1 + cos(0.5 \* x);

}

double C(double x) {

return -pow(x, 2);

}

double f(double x) {

return pow(pow((1 + x), 3), 0.5) + log(3 \* x);

}

double L(double u, double a1, double b1, double c1) {

return a1 \* A(u) + b1 \* B(u) + c1 \* C(u);

}

void main() {

double a = 0.31;

double b = 1.45;

double c, d, a2, b2, c2, sum;

const int n = 11;

double h = (b - a) / n;

double y[12];

double x[12];

double u[12];

double matrix[2][2];

x[0] = a;

x[n] = b;

y[0] = 1.3;

y[n] = 0.4;

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

x[i] = a + i \* h;

}

double det = a - b; // a \* 1 - b \* 1

matrix[0][0] = 1 / det;

matrix[0][1] = -1 / det;

matrix[1][0] = -b / det;

matrix[1][1] = a / det;

for (int i = 0; i < 2; i++) {

cout << matrix[i][0] << endl;

cout << matrix[i][1] << endl;

if (i == 0) c = matrix[i][0] \* y[0] + matrix[i][1] \* y[n];

d = matrix[i][0] \* y[0] + matrix[i][1] \* y[n];

}

cout << "c = " << c << ", d = " << d << endl;

for (int i = 0; i < n + 1; i++) {

u[i] = c \* x[i] + d;

cout << "u[" << i << "] = " << u[i] << endl;

}

double e = 0.24;

double starta = -5, enda = 0;

double startb = 5, endb = 7;

double startc = 0, endc = 2;

double step = 0.01;

bool gotit = false;

for (double a1 = starta; a1 <= enda && !gotit; a1 += step) {

for (double b1 = startb; b1 <= endb && !gotit; b1 += step) {

for (double c1 = startc; c1 <= endc && !gotit; c1 += step) {

sum = 0;

for (int i = 0; i < n + 1; i++) {

sum += abs(L(u[i], a1, b1, c1) - f(x[i]));

}

if (sum < e) {

cout << "Найдено решение для коэффициентов: a1 = " << a1 << ", b1 = " << b1 << ", c1 = " << c1 << endl;

for (int j = 0; j < 3; j++) {

a2 = a1;

b2 = b1;

c2 = c1;

}

gotit = true;

}

}

}

}

for (int i = 0; i < n + 1; i++) {

cout << "L(u[" << i << "]) = " << L(u[i], a2, b2, c2) << ", f(x[" << i << "]) = " << f(x[i]) << ", lu - fx = " << abs(L(u[i], a2, b2, c2) - f(x[i])) << endl;

}

cout << "Сумма погрешностей = " << sum << endl;

}

};

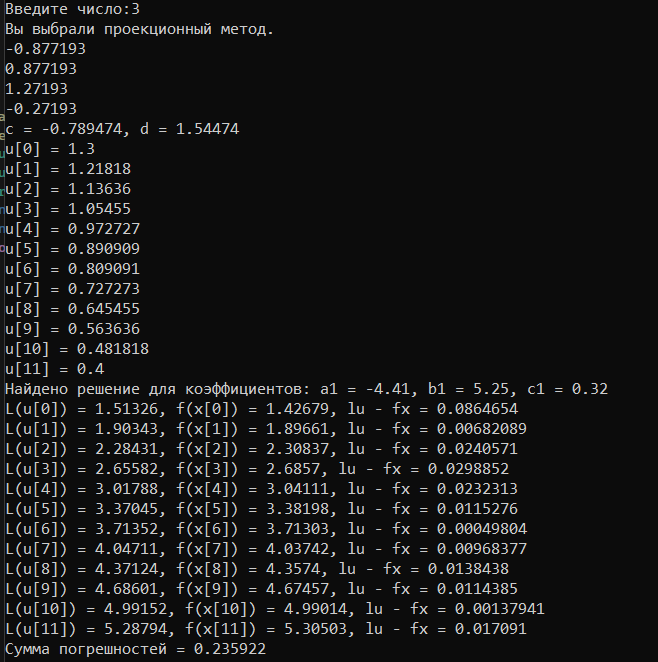


Рисунок 14. - Проекционный метод

# Реализация задачи в Java

## Основа и выбор пользователя

package Lab3;

import java.util.Scanner;

public class Menu {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

int choice;

int exit = 0;

do {

System.***out***.println("Программа решает ОДУ. Выберите метод, которым хотите решить (введите номер метода):");

System.***out***.println("1. Метод Рунге-Кутта.");

System.***out***.println("2. Метод прогонки.");

System.***out***.println("3. Проекционный метод.");

System.***out***.println("4. Завершение работы.");

System.***out***.println("Введите число: ");

choice = scanner.nextInt();

while(choice !=1 && choice !=2 && choice !=3 && choice !=4) {

System.***out***.println("Введите число от 1 до 4!");

while (true) {

if (scanner.hasNextInt()) {

choice = scanner.nextInt();

break;

} else {

System.***out***.println("Введите число!");

scanner.nextLine();

}

}

}

switch (choice) {

case 1:

System.***out***.println("Вы выбрали метод Рунге-Кутта.");

Runge.*main*(args);

break;

case 2:

System.***out***.println("Вы выбрали метод прогонки.");

//RunTh.main(args);

break;

case 3:

System.***out***.println("Вы выбрали проекционный метод.");

// Proj.main(args);

break;

case 4:

System.***out***.println("Завершение работы.\n");

System.***out***.println("Обратная связь:");

System.***out***.println("Студент гр. 21-САИ, Краличев Игорь Евгеньевич, ikralichev@list.ru");

exit = 1;

break;

}

} while (exit == 0);

// scanner.close();

} }

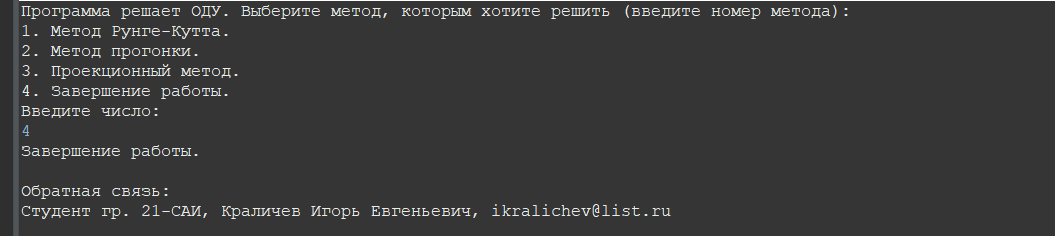


Рисунок 15.-Меню выбора пользователя

## Метод Рунге-Кутта

package Lab3;

public class Runge {

public static void main(String[] args) {

//Вариант 10

int n = 6;

double h = 0.1;

double[] x = new double [n];

x[0] = 0;

double[] y = new double [n];

double[] f = new double [n];

y[0] = 0; //y' = th(x+y)

double k1, k2, k3, k4;

System.***out***.println("y[" + 0 + "] = " + y[0] + ", x[" + 0 + "] = " + x[0]);

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

x[i] = 0 + i \* h;

}

System.***out***.println("Метод Рунге-Кутта:");

// метод Рунге-Кутта

for (int i = 0; i < n - 2; i++) {

k1 = Math.*tanh*(x[i] + y[i])\* h;

k2 = Math.*tanh*((x[i] + h / 2) + ((k1) / 2 + y[i]))\*h;

k3 = Math.*tanh*((x[i] + h / 2) + ((k2) / 2 + y[i]))\*h;

k4 = Math.*tanh*((x[i] + h) + (y[i] + k3))\*h;

f[i] =Math.*tanh*(x[i] + y[i]);

y[i + 1] = y[i] + ((k1 + 2 \* k2 + 2 \* k3 + k4) / 6);

}

for (int i = 0; i < n - 3; i++) {

System.***out***.println("y[" + (i + 1) + "] = " + y[i + 1] + ", x[" + (i + 1) + "] = " + x[i + 1]);

}

System.***out***.println("Метод Аддамса экстраполяционный:");

// метод Аддамса экстраполяционный

for (int i = n - 3; i < n - 1; i++) {

f[i] =Math.*tanh*(x[i] + y[i]);

y[i + 1] = y[i] + (h / 24) \* (55 \* f[i] - 59 \* f[i - 1] + 37 \* f[i - 2] - 9 \* f[i - 3]);

System.***out***.println("y[" + (i + 1) + "] = " + y[i + 1] + ", x[" + (i + 1) + "] = " + x[i + 1]);

}

f[5] = Math.*tanh*(x[5] + y[5]);

System.***out***.println("Метод Аддамса интерполяционный:");

// метод Аддамса интерполяционный

for (int i = n - 3; i < n - 1; i++) {

y[i + 1] = y[i] + (h / 24) \* (9 \* f[i + 1] + 19 \* f[i] - 5 \* f[i - 1] + f[i - 2]);

System.***out***.println("y[" + (i + 1) + "] = " + y[i + 1] + ", x[" + (i + 1) + "] = " + x[i + 1]);

}

System.***out***.println("Правило Рунге:");

// правило Рунге

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

System.***out***.println("Погрешность y[" + i + "] = " + Math.*abs*((Math.*pow*(y[i], h) - Math.*pow*(y[i], h \* 2)) / (15)));

}

}

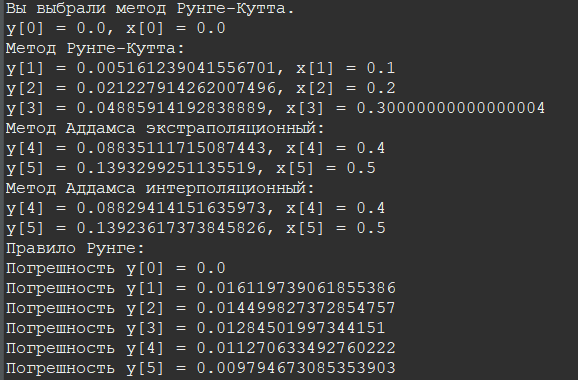
}

Рисунок 16. - Метод Рунге-Кутта и Адамса

## Метод прогонки

package Lab3;

public class RunTh {

static double *a* = 0.31;

static double *b* = 1.45;

static int *n* = 12;

static double *h* = (*b* - *a*) / 11;

static double[] *y* = new double[*n*];

static double[] *x* = new double[*n*];

static double[] *L* = new double[*n*];

static double[] *K* = new double[*n*];

public static double A(double x) {

return Math.*pow*((1+x), 0.7);

}

public static double B(double x) {

return 1.1 + Math.*cos*(0.5 \* x);

}

public static double C(double x) {

return -Math.*pow*(x, 2.0);

}

public static double f(double x) {

return Math.*pow*(Math.*pow*((1.0 + x),3.0),0.5) + Math.*log*(3.0\*x);

}

public static double ai(double x) {

return (*A*(x)) / (*h* \* *h*);

}

public static double bi(double x) {

return *C*(x) - ((2 \* *A*(x)) / (*h* \* *h*)) - (*B*(x) / *h*) ;

}

public static double ci(double x) {

return (*A*(x) / (*h* \* *h*)) + (*B*(x) / *h*);

}

public static void main(String[] args) {

*y*[0] = 1.3;

*y*[*n*-1] = 0.4;

*L*[0] = 0;

*K*[0] = *y*[0];

*x*[0] = 0.31;

*x*[*n*-1] = 1.45;

for (int i = 1; i < *n* - 1; i++) {

*x*[i] = *a* + i \* *h*;

}

for (int i = 1; i < *n*-1; i++) {

*L*[i] = (- *ci*(*x*[i])) / (*ai*(*x*[i]) \* *L*[i - 1] + *bi*(*x*[i]));

System.***out***.println("L " + *L*[i]);

*K*[i] = ((*f*(*x*[i-1]) - *ai*(*x*[i]) \* *K*[i - 1]) / (*ai*(*x*[i]) \* *L*[i - 1] + *bi*(*x*[i])));

System.***out***.println("f " + *f*(*x*[i-1]));

System.***out***.println("K " + *K*[i]);

}

for (int i = *n* - 2; i > - 1; i--) {

*y*[i] = *L*[i] \* *y*[i + 1] + *K*[i];

}

System.***out***.println("Решение методом прогонки:");

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

System.***out***.println("y[" + i + "] = " + *y*[i]);

}

}

}

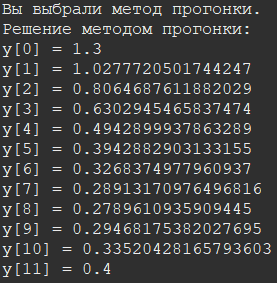


Рисунок 17. - Метод прогонки

## Проекционный метод

package Lab3;

public class Proj {

public static double A(double x) {

return Math.*pow*((1 + x), 0.7);

}

public static double B(double x) {

return 1.1 + Math.*cos*(0.5 \* x);

}

public static double C(double x) {

return -Math.*pow*(x, 2);

}

public static double f(double x) {

return Math.*pow*(Math.*pow*((1 + x), 3), 0.5) + Math.*log*(3 \* x);

}

public static double L(double u, double a1, double b1, double c1) {

return a1 \* *A*(u) + b1 \* *B*(u) + c1 \* *C*(u);

}

double a = 0.31;

double b = 1.45;

double c, d, a2, b2, c2, sum;

final int n = 11;

double h = (b - a) / n;

double[] y = new double[12];

double[] x = new double[12];

double[] u = new double[12];

double[][] matrix = new double[2][2];

public void main(String[] args) {

x[0] = a;

x[n] = b;

y[0] = 1.3;

y[n] = 0.4;

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

x[i] = a + i \* h;

}

double det = a - b; // a \* 1 - b \* 1

matrix[0][0] = 1 / det;

matrix[0][1] = -1 / det;

matrix[1][0] = -b / det;

matrix[1][1] = a / det;

for (int i = 0; i < 2; i++) {

System.***out***.println(matrix[i][0]);

System.***out***.println(matrix[i][1]);

if (i == 0) c = matrix[i][0] \* y[0] + matrix[i][1] \* y[n];

d = matrix[i][0] \* y[0] + matrix[i][1] \* y[n];

}

System.***out***.println("c = " + c + ", d = " + d);

for (int i = 0; i < n + 1; i++) {

u[i] = c \* x[i] + d;

System.***out***.println("u[" + i + "] = " + u[i]);

}

double e = 0.24;

double starta = -5, enda = 0; //-14 -14 -12

double startb = 5, endb = 7; //-10.5 -11 -8

double startc = 0, endc = 2; //-9 -9 -8

double step = 0.01;

boolean gotit = false;

for (double a1 = starta; a1 <= enda && !gotit; a1 += step) {

for (double b1 = startb; b1 <= endb && !gotit; b1 += step) {

for (double c1 = startc; c1 <= endc && !gotit; c1 += step) {

sum = 0;

for (int i = 0; i < n + 1; i++) {

sum += Math.*abs*(*L*(u[i], a1, b1, c1) - *f*(x[i]));

}

if (sum < e) {

System.***out***.println("Найдено решение для коэффициентов: a1 = " + a1 + ", b1 = " + b1 + ", c1 = " + c1);

for (int j = 0; j < 3; j++) {

a2 = a1;

b2 = b1;

c2 = c1;

}

gotit = true;

}

}

}

}

for (int i = 0; i < n + 1; i++) {

System.***out***.println("L(u[" + i + "]) = " + *L*(u[i], a2, b2, c2) + ", f(x[" + i + "]) = " + *f*(x[i]) + ", lu - fx = " + Math.*abs*(*L*(u[i], a2, b2, c2) - *f*(x[i])));

}

System.***out***.println("Сумма погрешностей = " + sum);

}

}

}

}

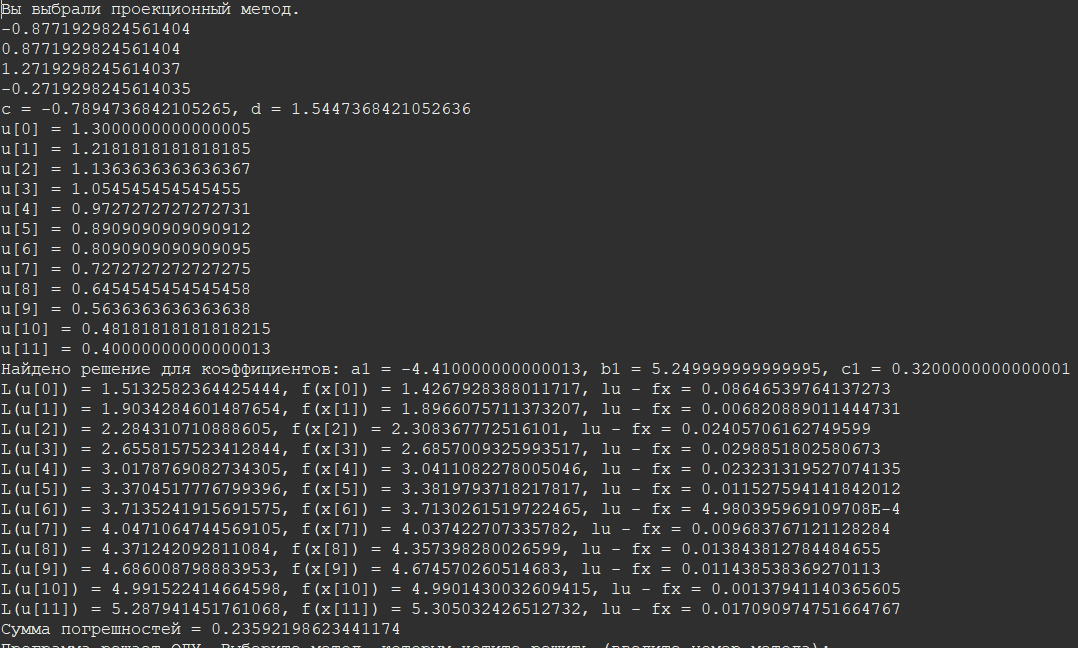


Рисунок 18. - Проекционный метод

# Результат и анализ всех реализаций

Таблица 5

Таблица результатов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод | Ручной счёт | Excel | Mathcad | C++ | Java |
| Рунге-Кутта, Адамса |  |  |  |  |  |
| Прогонки |  |  |  |  |  |
| Проекционный |  |  |  |  |  |

Всеми инструментами можно достичь высокой точности, но среди прочих можно выделить Excel и Mathcad, которые подходят больше остальных для математических вычислений.

# Список литературы

**1.** **Численное решение задач экономики с использованием EXCEL, C++ и MATLAB [Электронные текстовые данные] : Учеб.пособие / Л.Ю. Катаева [и др.]; НГТУ им.Р.Е.Алексеева. - Н.Новгород : [Изд-во НГТУ], 2020. - 230 с. : ил. - Прил.:c.188-230. - Библиогр.:с.187.:**

[**https://fdp.nntu.ru/books/Chisl\_reshenie\_zadach\_economiki/Chisl\_reshenie\_zadach\_economiki/assets/basic-html/index.html#189в**](https://fdp.nntu.ru/books/Chisl_reshenie_zadach_economiki/Chisl_reshenie_zadach_economiki/assets/basic-html/index.html#189%D0%B2)**.**

**2. Численные методы : Курс лекций / В.А. Срочко. - СПб.; М.; Краснодар : Лань, 2010. - 202 с. - (Учебники для вузов. Специальная литература). - Библиогр.:с.200. - ISBN 978-5-8114-1014-9 : 180-00.**

[**https://studfile.net/preview/5793014/page:4/**](https://studfile.net/preview/5793014/page:4/)

**3. Численные методы линейной алгебры : Учеб.пособие / Г.С. Шевцов, О.Г. Крюкова, Б.И. Мызникова. - 2-е изд.,испр.и доп. - СПб.; М.; Краснодар : Лань, 2011. - 495 с. - (Учебники для вузов. Специальная литература). - Предм.указ.:с.491-495. - Библиогр.:с.489-490. - ISBN 978-5-8114-1246-4 : 465-00.:**

[**https://dpm.pstu.ru/images/R/Z/shevcov\_lineynaya\_algebra.pdf**](https://dpm.pstu.ru/images/R/Z/shevcov_lineynaya_algebra.pdf)