|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.03 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 2 |

**Дисциплина:** Моделирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-62Б |  |  | Н.А. Гарасев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | В.М. Градов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

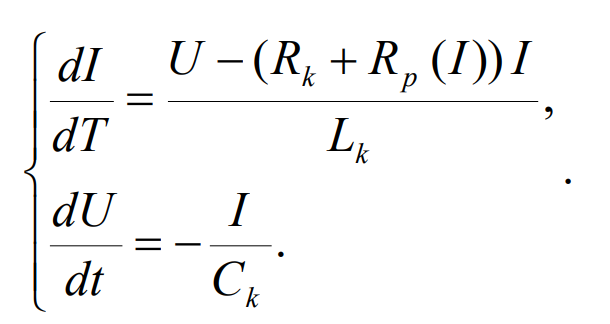
Москва, 2021

**Тема:** Программно-алгоритмическая реализация метода Рунге-Кутта 4-го порядка точности при решении системы ОДУ в задаче Коши.

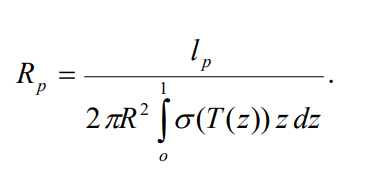
**Цель работы:** Получение навыков разработки алгоритмов решения задачи Коши при реализации моделей, построенных на системе ОДУ, с использованием метода Рунге-Кутта 4-го порядка точности.

**Исходные данные:**

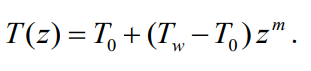
Задана система электротехнических уравнений, описывающих разрядный контур, включающий постоянное активное сопротивление Rk , нелинейное сопротивление R (I) p , зависящее от тока I , индуктивность Lk и емкость Ck .



Начальные условия: t =0, I = I0 , U = U0. Здесь I, U - ток и напряжение на конденсаторе. Сопротивление Rp рассчитать по формуле

****

**, где**

****

**Параметры разрядного контура:**

R=0.35 см

lэ=12 см

Lk=187 10-6 Гн

Ck=268 10-6 Ф

Rk=0.25 Ом

Uco=1400 В

Io=0..3 A

Tw=2000 K

**Ход работы:**

Для вычисления I и U, заданных системой электротехнических уравнений, используется метод Рунге-Кутта 4-го порядка точности, описанный в листинге 1.

Листинг 1. Метод Рунге-Кутта 4-го порядка точности.

**def** RungeKutta4**(**I**,** U**,** h**):**

k1 **=** f1**(**I**,** U**)**

q1 **=** f2**(**I**)**

k2 **=** f1**(**I **+** h **\*** k1 **/** 2**,** U **+** h **\*** q1 **/** 2**)**

q2 **=** f2**(**I **+** h **\*** k1 **/** 2**)**

k3 **=** f1**(**I **+** h **\*** k2 **/** 2**,** U **+** h **\*** q2 **/** 2**)**

q3 **=** f2**(**I **+** h **\*** k2 **/** 2**)**

k4 **=** f1**(**I **+** h **\*** k3**,** U **+** h **\*** q3**)**

q4 **=** f2**(**I **+** h **\*** k3**)**

I\_res **=** I **+** h **\*** **(**k1 **+** 2 **\*** k2 **+** 2 **\*** k3 **+** k4**)** **/** 6

U\_res **=** U **+** h **\*** **(**q1 **+** 2 **\*** q2 **+** 2 **\*** q3 **+** q4**)** **/** 6

**return** I\_res**,** U\_res

Где функции f1 и f2 – функции системы электротехнических уравнений из условия, описанные в листинге 2.

Листинг 2. Система электротехнических уравнений.

**def** f1**(**I**,** U**):**

**return** **(**U **-** **(**Rk **+** calcR**(**I**))** **\*** I**)** **/** Lk

**def** f2**(**I**):**

**return** **-**1 **\*** I **/** Ck

Rp – рассчитывается по формуле из условия, реализация представлена в листинге 3.

Листинг 3. Вычисление Rp.

**def** calcR**(**I**):**

r **=** l **/** **(**2 **\*** math**.**pi **\*** R **\*** R **\*** integral**(**I**))**

**return** r

Вычисление численного значения интеграла, зависящего от силы тока, реализован в листинге 4.

Листинг 4. Вычисление интеграла.

**def** integral**(**I**):**

**return** trapezoid**(**T**,** 0**,** 1**,** 10 **\*\*** **-**2**,** I**)**

**def** trapezoid**(**f**,** a**,** b**,** h**,** i**):**

p **=** **round((**b **-** a**)** **//** h**)**

s **=** 0

**for** \_ **in** **range(**p**):**

s **+=** **(**f**(**a**,** i**)** **+** f**(**a **+** h**,** i**))** **/** 2 **\*** h

a **+=** h

**return** s

Интеграл вычисляется по методу трапеции с шагом 10^-2 (такой шаг выбран как, что-то среднее между временем вычислением и точностью результата). Значение в точках вычисляются с помощью функции T, представленную в листинге 5.

Листинг 5. Функция Т().

**def** T**(**z**,** i**):**

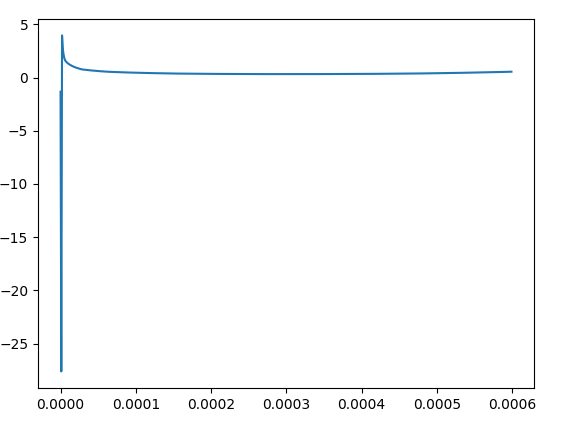
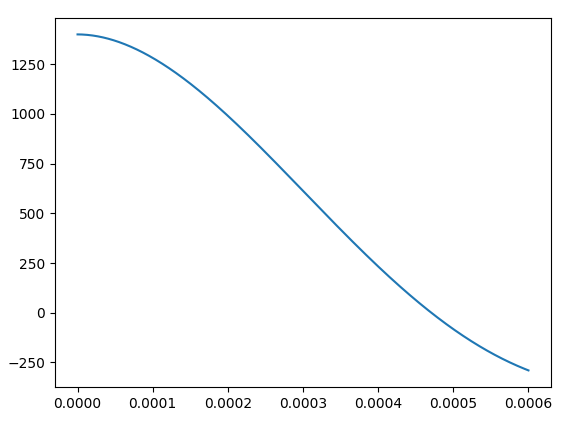
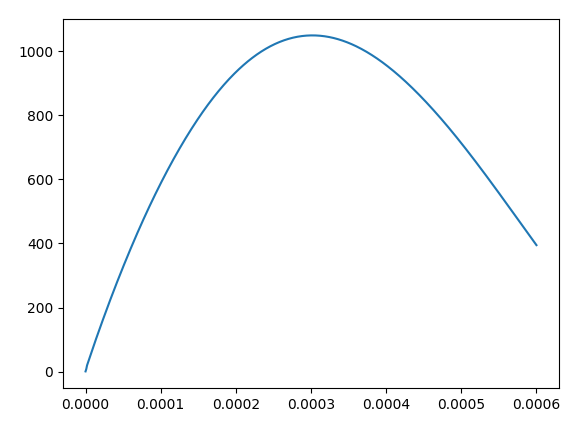
temp **=** newton**(**tab2**[**0**],** tab2**[**1**])**

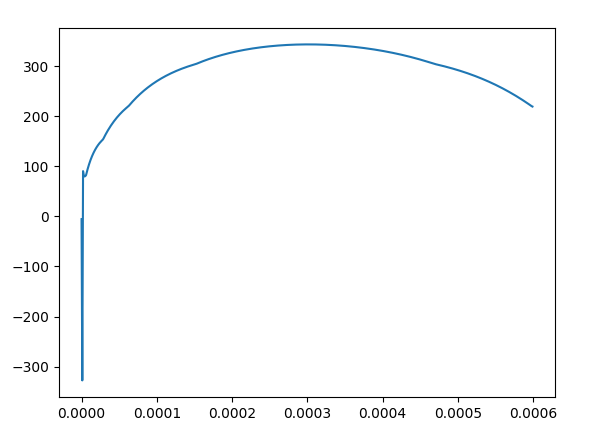
f\_t **=** linear**(**tab1**[**0**],** tab1**[**1**])**

f\_m **=** linear**(**tab1**[**1**],** tab1**[**2**])**

**return** temp**(**f\_t**(**i**)** **+** **(**Tw **-** f\_t**(**i**))** **\*** z**\*\***f\_m**(**f\_t**(**i**)))**

Первым делам в функции Т() определяются функции temp(), f\_t(), f\_m(). С помощью разных методов интерполяции, заданные функции вычисляют значения при заданном i. Функция temp() - используя интерполяцию по Ньютону, вычисляет значение сигмы по заданной температуре. Для функций f\_t() и f\_m() был выбран метод линейной интерполяции, так как у узлов в таблице переменный шаг. Функция f\_t() – вычисляет значение T при заданном I. Функция f\_m() – вычисляет значение m при заданном T.

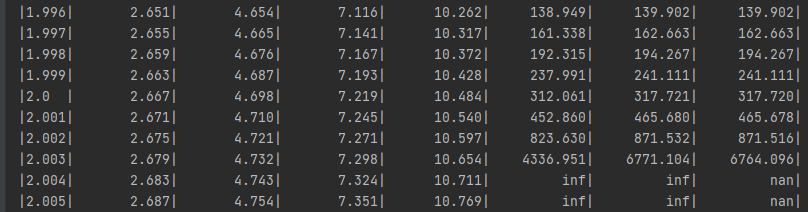
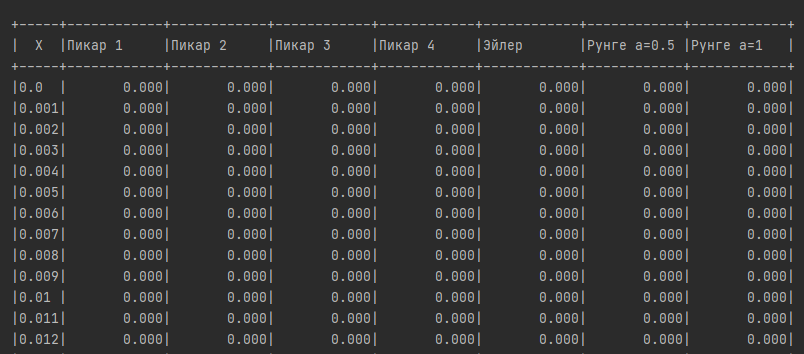




Результат программы:

Программа выводит таблицу, содержащую значения аргумента с заданным шагом в интервале [0, xmax] и результаты расчета функции u(x) в приближениях Пикара (от 1-го до 4-го), а также численными методами. Границу интервала xmax выбирать максимально возможной из условия, чтобы численные методы обеспечивали точность вычисления решения уравнения u(x) до второго знака после запятой.

Пример вывода.

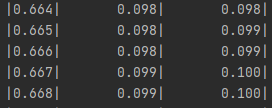


**Вопросы.**

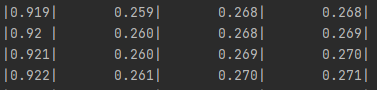
1. **Укажите интервалы значений аргумента, в которых можно считать решением заданного уравнения каждое из первых 4-х приближений Пикара. Точность результата оценивать до второй цифры после запятой. Объяснить свой ответ.**

Каждое приближение Пикара вычисляет значение все точнее и точнее. А значит, что n-ое приближение вычисляет значения ближе к истинному, чем (n-1)-ое приближение.

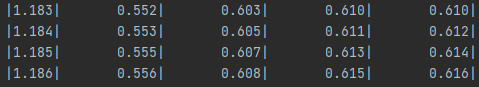
Первое приближение: **от 0 до 0,665**



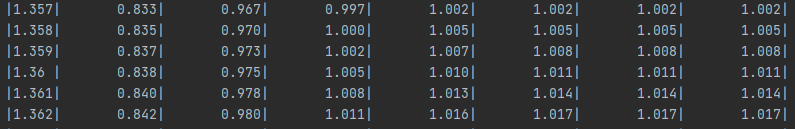
Второе приближение: **от 0 до 0,92**



Третье приближение**: от 0 до 1,184**



Для четвертого приближения Пикара нам необходимо знать пятое приближение Пикара, либо можно воспользоваться численными методами.



Четвертое приближение: **от 0 до 1,359**

Все вычисления производились для шага 10^-6

1. **Пояснить, каким образом можно доказать правильность полученного результата при фиксированном значении аргумента в численных методах.**

Численные методы зависят от шага. Возьмем точку x=2. Посмотрим, как меняются значения в этой точке в зависимости от численных методов.

Шаг = 10^-3

Эйлер: 126.597

Рунге-Кутта (a=0.5): 305.208

Рунге-Кутта (a=1): 300.663

Шаг = 10^-4

Эйлер: 270.068

Рунге-Кутта (a=0.5): 317.566

Рунге-Кутта (a=1): 317.490

Шаг = 10^-5

Эйлер: 312.061

Рунге-Кутта (a=0.5): 317.721

Рунге-Кутта (a=0.5): 317.720

Шаг = 10^-6

Эйлер: 317.145

Рунге-Кутта (a=0.5): 317.722

Рунге-Кутта (a=1): 317.722

Шаг = 10^-7

Эйлер: 317.665

Рунге-Кутта (a=0.5): 317.722

Рунге-Кутта (a=1): 317.722

Получается, что для метода Рунге-Кутта для значения x=2 достаточно шага 10^-5. Для метода Эйлера для значения x=2 при шаге = 10^-7 ответ с точностью до 2х знаков после запятой отличается от метода Рунге-Кутта. При попытке вычисления с шагом = 10^-8 программа выдает ошибку памяти.

**Вывод:**

При уменьшении шага точность увеличивается, однако компьютер работает с ограниченной разрядной сеткой, а значит нельзя знак бесконечно приближать к нулю.

1. **Каково значение функции при х = 2, т.е. привести значение u(2).**

При шаге = 10^-7

Эйлер: u(2) = 317.665

Рунге-Кутта (a=0.5): u(2) = 317.722

Рунге-Кутта (a=1): u(2) = 317.722