Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра оптимального управления

Программа для решения краевых задач методом продолжения по параметру на языке Python

Студент: Эмиров Самир Магомедович, группа 313

Преподаватели: Киселёв Юрий Николаевич

Аввакумов Сергей Николаевич

Дряженков Андрей Александрович

Москва, 2023 г.

Содержание

1 Метод продолжения по параметру

Рассмотрим краевую задачу

$$\dot{x} = f(t,x),\ R(x(a),x(b)) = 0,\ t \in [a,b],\ x \in E^n. \tag{1}$$

Здесь $f(t,x) : E^1 \times E^n \mapsto E^n,\ R(x,y) : E^n \times E^n \mapsto E^n$ являются гладкими векторными функциями. Предполагая существование решения краевой задачи (1), обсудим алгоритмические вопросы поиска её решения. Решение краевой задачи можно свести к некоторому нелинейному векторному уравнению в E^n. Выберем некоторую точку $t_* \in [a,b]$ и рассмотрим задачу Коши

$$x = f(t,x),\ x|_{t=t_*} = p \in E^n. \tag{2}$$

Свобода выбора точки t_* может быть полезна для вычислительной практики. Пусть

$$x(t,p),\ a \leq t \leq b. \tag{3}$$

— решение задачи Коши (2). Предполагается продолжимость решения (3) на весь отрезок $[a,b]$ для любого p. Начальное значение параметра $p \in E^n$ ищется из условий выполнения векторного граничного условия в задаче (1), т.е. искомое p является решением уравнения

$$\Phi(p) \equiv R(x(a,p),x(b,p)) = 0. \tag{4}$$

Итак, краевая задача (1) сведена к конечному векторному уравнению (4).

Далее к уравнению (4) применяется метод продолжения. Матрица $\Phi'(p)$ определяется равенством

$$\Phi'(p) = R'_x \frac{\partial x(a,p)}{\partial p} + R'_y \frac{\partial x(b,p)}{\partial p}$$

Здесь $(n \times n)$ — матрицы $R'_x(x,y),\ R'_y(x,y)$ вычисляются вдоль решения (3),
т.е. при $x = x(a,p),\ y = x(b,p)$. Введём обозначение

$$X(t,p) \equiv \frac{\partial x(t,p)}{\partial p}$$

для $(n \times n)$—матрицы производных решения (3) по начальному условию. Матрица $X(t,p)$ определяется дифференциальным уравнением в вариациях

$$\dot{X} = AX,\ X|_{t=t_*} = I,\ a \leq t \leq b,$$

где $A = A(t,p) \equiv f'_x(t,x)|_{x=x(t,p)}$ есть $(n \times n)$—матрица, I—единичная матрица. Основная задача Коши схемы продолжения по параметру имеет вид

$$\textbf{IVP:} \quad \frac{dp}{d\mu} = -[\Phi'(p)]^{-1}\Phi(p_0),\ p(0) = p_0,\ 0 \le \mu \le 1, \tag{5}$$

где

$$\Phi(p) = R(x(a,p), x(b,p)),$$
$$\Phi'(p) = R'_x(x(a,p), x(b,p))X(a,p) + R'_y(x(a,p), x(b,p))X(b,p).$$

Для одновременного вычисления векторной функции $x(t,p)$ и матричной функции $X(t,p)$ может быть записана следующая векторно-матричная задача Коши

$$\begin{cases} \dot{x} = f(t,x), & x|_{t=t_*} = p, \\ \dot{X} = f'_x(t,x)X, & X|_{t=t_*} = I,\ a \le t \le b. \end{cases} \tag{6}$$

Задачу Коши (5) будем называть внешней задачей, задачу Коши (6) — внутренней задачей. Таким образом, предлагается итерационный процесс для решения рассматриваемой краевой задачи (1) на основе внешней задачи (5) и внутренней задачи (6). На одном шаге итерационного процесса выполняется решение внешней задачи (5), в ходе решения которой происходит многократное обращение к решению внутренней задачи Коши (6) при различных значениях параметра p.

2 О проекте

2.1 Постановка задачи

В рамках курса «Практикум на ЭВМ» была поставлена задача изучить метод продолжения по параметру для решения краевых задач и написать программу на языке Python, реализующую данный метод.

2.2 Используемые библиотеки

Tkinter – библиотека (входит в стандартную библиотеку Python), предназначенная для работы с графическим интерфейсом.
NumPy – библиотека для высокоуровневой работы с многомерными массивами.
SciPy – библиотека для научных вычислений, в данной программе из нее используется функция, решающая задачу Коши.
odeintw – оболочка scipy.integrate.odeint, которая позволяет обрабатывать комплексные и матричные дифференциальные уравнения.
SymPy – библиотека символьных вычислений.
Pandas – библиотека для обработки и анализа табличных данных.
Matplotlib – библиотека для визуализации данных.
PIL – библиотека для работы с изображениями.

2.3 Пользовательский интерфейс

При запуске программы пользователя встречает основное окно:

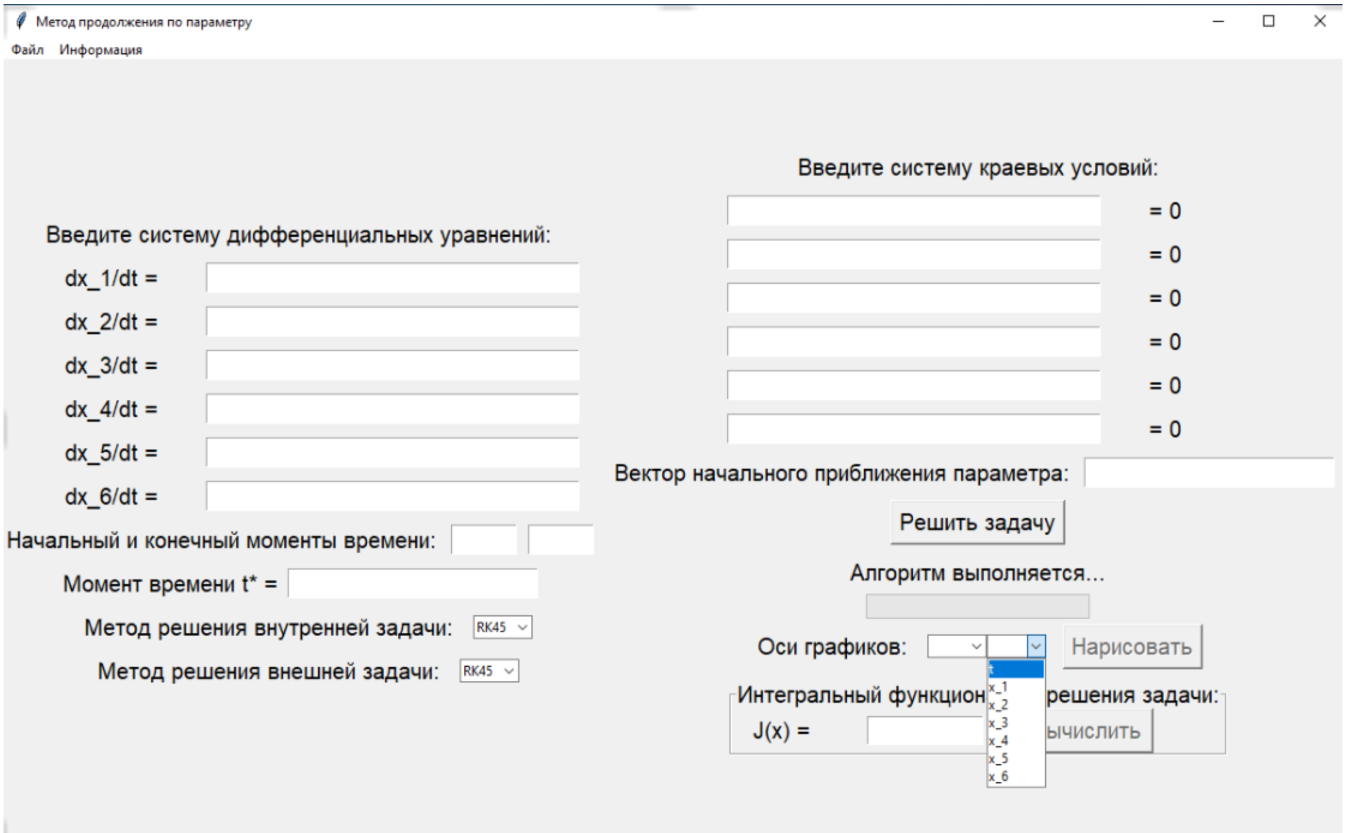

Рис. 1: Основное окно программы

2.3.1 Ввод данных

Пользователь вводит правую часть системы дифференциальных уравнений $f_i(x_i, t)$ с краевыми условиями $R(x_i(a), x_j(b)) = 0$, начальный вектор p_0, концы отрезка решения задачи a и b, точку t^*.
Если возникают трудности с вводом, имеется возможность перейти в раздел «информация -> помощь». Также можно перейти в раздел «Файл» -> «Открыть задачу...» и выбрать один из вариантов, после чего в поля ввода подставятся соответствующие уравнения.

2.3.2 Получение результатов

После ввода всех данных краевой задачи можно вычислить ее решение по нажатию кнопки «Решить задачу». Перед вычислением есть возможность выбрать 1 из 6 численных методов для решения внутренней и внешней задачи. После вычисления решения имеется возможность графически отобразить его. Также имеется возможность вычислить значение интегрального функционала вдоль полученной траектории. Примеры будут рассмотрены ниже.

2.3.3 Раздел «Меню»

В левом верхнем углу интерфейса находится раздел «меню», в котором можно открыть окна «Файл» и «Информация».

- Файл - содержит в себе функции «Сохранить задачу», и «Открыть прошлый ввод» для сохранения введённого примера задачи и загрузки уже имеющейся. Также имеется функция «Сохранить решение», сохраняющая численное решение в виде csv файла.

- Информация - содержит в себе функции «О программе» (общая информация о проекте), «Об авторе» и «Помощь» (подробное руководство пользователя),

3 Примеры

3.1 Краевая задача двух тел

$$
\begin{cases}
\dot{x}_1 = x_3, & x_1(0) = a_1,\ x_1(T) = b_1, \\
\dot{x}_2 = x_4, & x_2(0) = a_2,\ x_2(T) = b_2, \\
\dot{x}_3 = -x_1(x_1^2 + x_2^2)^{-3/2}, & \\
\dot{x}_4 = -x_2(x_1^2 + x_2^2)^{-3/2}. &
\end{cases}
$$

Для данных

$$T = 7,\ a_1 = 2,\ a_2 = 0,\ b_1 = 1.0738644361,\ b_2 = -1.0995343576,$$

при выборе параметра $t_* = 0$, для начального приближения

$$p_0 = [2, 0, 0.5, -0.5].$$

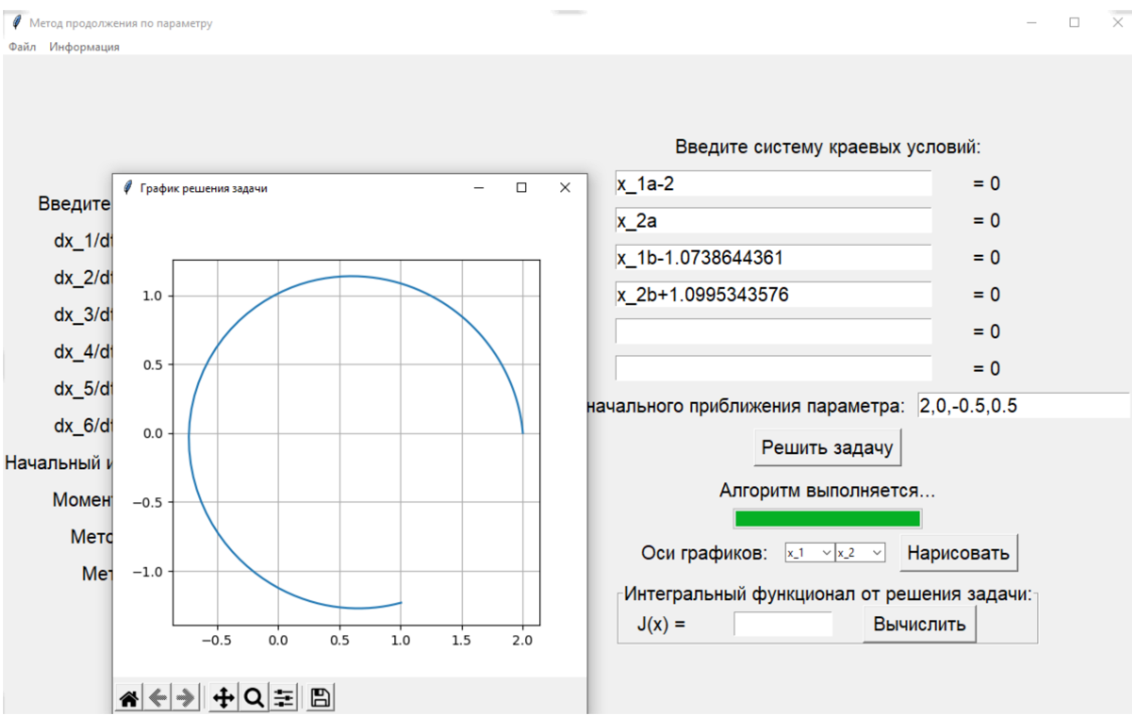

Рис. 2: График в осях x_1, x_2

3.2 Предельные циклы в системе Эквейлера

$$\begin{cases} \dot{x}_1 = x_3 x_2, & x_1(0) = x_4(0),\ x_1(1) = x_4(1), \\ \dot{x}_2 = x_3(-x_1 + sin(x_2)), & x_2(0) = 0,\ x_2(1) = 0, \\ \dot{x}_3 = 0, \\ \dot{x}_4 = 0. \end{cases}$$

При выборе параметра $t_* = 0$, для начального приближения

$$p_0 = [2, 0, 2\pi, 2].$$

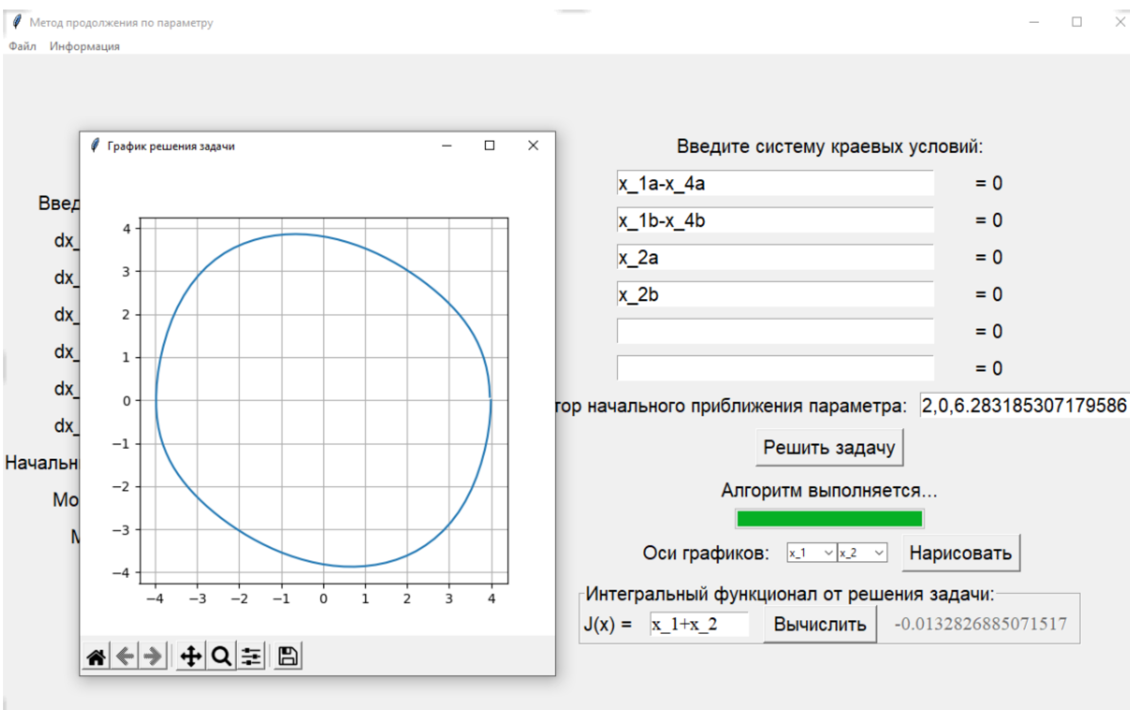

Рис. 3: График в осях x_1, x_2

3.3 Функционал типа "энергия"для трёхкратного интегратора

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = x_3, \\ \dot{x}_3 = \frac{1}{2}(\sqrt{\nu + (x_6+1)^2} - \sqrt{\nu + (x_6-1)^2}), \\ \dot{x}_4 = 0, \\ \dot{x}_5 = -x_4, \\ \dot{x}_6 = -x_5. \end{cases} \qquad x(0) = (1, 0, 0),\ x(T) = (0, 0, 0),\ T = 3.275,$$

При выборе параметров $t_* = 3.275,\ \nu = 10^{-10}$, для начального приближения

$$p_0 = [0, 0, 0, -2.9, 4.9, -2.9].$$

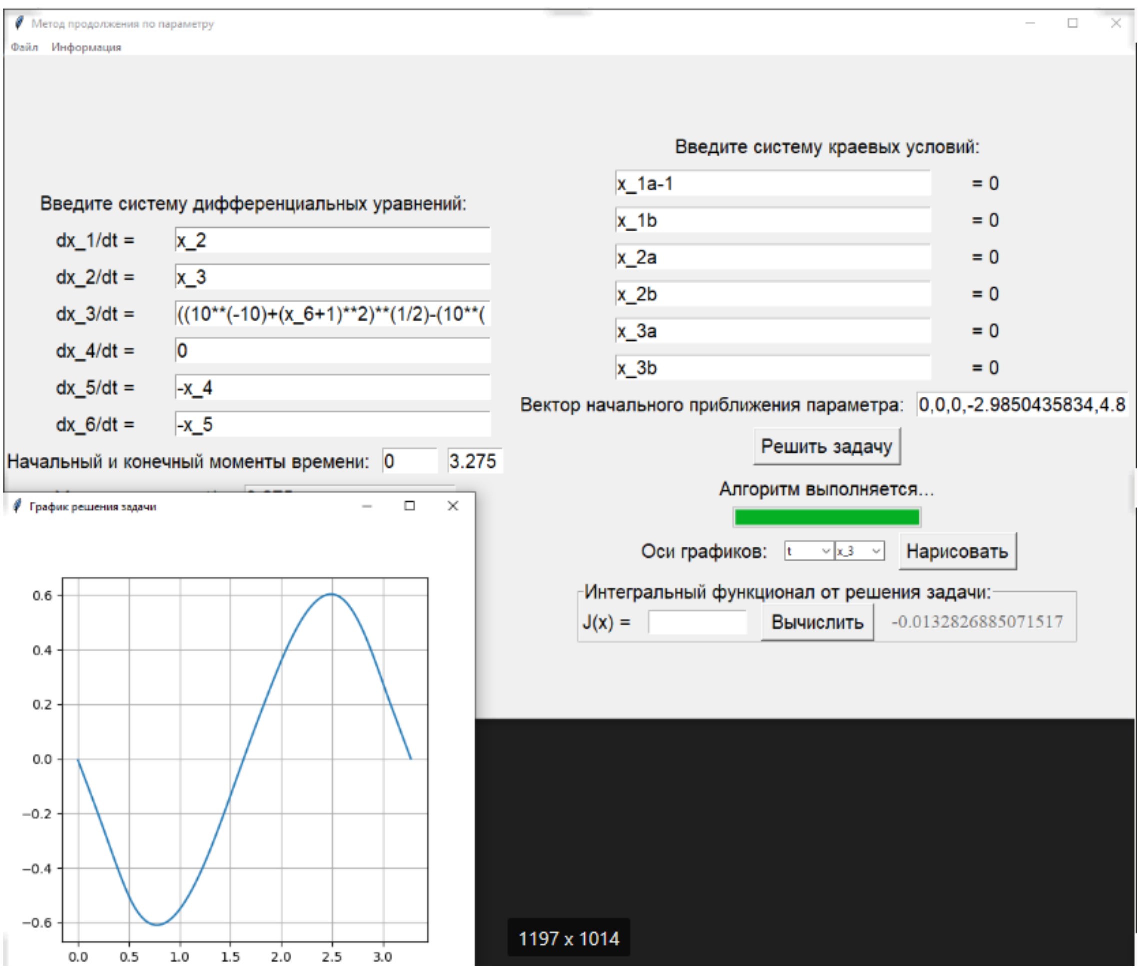

Рис. 4: График функции $x_3(t)$

Список литературы

[1] Ю.Н. Киселёв, С.Н. Аввакумов, М.В. Орлов *Оптимальное управление. Линейная теория и приложения* // 2007, Издательский отдел факультета ВМК МГУ имени М.В. Ломоносова, 270.

[2] Лекции А.А. Дряженкова по практикуму.

[3] С. М. Львовский *Набор и вёрстка в системе LaTeX* // Издательство Макс Пресс, 2003.

[4] Официальный сайт языка Python *python.org*.