

Санкт-Петербургский государственный университет

**Дорофеева Елизавета Алексеевна**

**Выпускная квалификационная работа магистра**  
***Интегрирование уравнений Кеплера с помощью***  
***полных полиномиальных систем УрЧП***

Направление 01.04.02 «Прикладная математика и информатика»  
Основная образовательная программа ВМ.5517 «Методы прикладной  
математики и информатики в задачах управления»

Научный руководитель: доцент,  
кафедра механики управляемого  
движения,  
кандидат физ. –мат. наук,  
Потоцкая И.Ю.

Рецензент: Ведущий специалист по  
тестированию, ООО "Воркми",  
магистр физ. –мат. наук,  
Брегман Анна Михайловна.

Санкт-Петербург  
2023

## Оглавление

Введение .....	3
Постановка задачи.....	5
Обзор литературы .....	6
Глава 1. Задача двух тел.....	7
1.1 Постановка задачи.....	7
1.2 Кеплеровы элементы орбиты.....	7
1.3 Уравнения движения задачи двух тел и их решения. ....	8
Глава 2. Система дифференциальных уравнений для решения уравнения Кеплера. ....	10
2.1 Полиномиальные системы.....	10
2.2 Подготовка к применению алгоритма методов рядов Тейлора .....	10
Глава 3. Реализация алгоритма.....	12
3.1 Описание программы .....	12
3.2 Результаты решения уравнения Кеплера .....	12
3.3 Результаты решения задачи двух тел .....	14
Заключение. ....	16
Литература.....	17
Приложение 1 .....	18
Приложение 2 .....	19
Приложение 3 .....	20
Приложение 4 .....	23
Приложение 5 .....	24
Приложение 6 .....	25
Приложение 7 .....	26

## Введение

Численные методы решения дифференциальных уравнений набрали большую популярность в современном мире. Компьютерные технологии достаточно продвинуты для проведения данных вычислений, однако иногда существующих мощностей может не хватать для достаточно точного или достаточно быстрого решения поставленных задач. Поэтому исследования в области изучения, сравнения и улучшения имеющихся численных методов крайне важны как для дальнейшего продвижения науки в целом, так и для прикладного использования наших математических знаний.

Подобное движение может быть описано системой обыкновенных дифференциальных уравнений. Такие системы хорошо изучены и для их решения были разработаны различные численные методы. Однако выбор наиболее оптимального метода напрямую зависит от конкретной задачи. Одним из методов решения систем дифференциальных уравнений является метод рядов Тейлора, который дает преимущество в точности вычислений. В общем случае этот метод требует значительных компьютерных мощностей, однако он работает намного быстрее если интегрируемая система имеет полиномиальные правые части. Из положительных особенностей метода Тейлора также стоит подчеркнуть его способность решать жесткие системы, в отличие от метода Рунге-Кутты.

Математическая модель задачи двух тел может быть представлена в виде полной полиномиальной системы уравнений в частных производных. Но до недавнего времени литературы, описывающей алгоритм метода рядов Тейлора для подобных систем, не было. Статья «Estimates for Taylor series method to polynomial total systems of PDEs» (Бабаджанянц Л.К, Потоцкая И.Ю., Пупышева Ю.Ю.), дающая необходимые математические инструменты для интегрирования полных полиномиальных систем УрЧП была опубликована в «Вестнике СПбГУ» только в начале мая 2021 года. Для программной реализации метода рядов Тейлора, в данной работе будет рассматриваться

система полиномиальных уравнений, полученная в своей выпускной квалификационной работе магистра Дмитриевой Анной.

В данной работе объектом исследования является задача двух тел, предметом исследования – методы решения системы дифференциальных уравнений, описывающих задачу двух тел.

Выполнение этой работы было разделено на несколько этапов: изучение литературы, связанной с задачей двух тел, анализ полученной информации, моделирование системы, программная реализация алгоритма решения полиномиальных уравнений методом рядов Тейлора, отладка программы на решении уравнения Кеплера и сравнение точности данного алгоритма, программная реализация решения задачи двух тел.

Структурно работа состоит из нескольких глав. В первой главе приводится описание задачи двух тел. Вторая глава посвящена построению системы полиномиальных уравнений в частных производных для решения уравнения Кеплера. В третьей главе представлено описание алгоритма программной реализации метода Тейлора для получения численного решения системы.

## Постановка задачи

Целью работы является применение метода рядов Тейлора для полной системы полиномиальных уравнений в частных производных для решения уравнения Кеплера и задачи двух тел.

Были поставлены следующие задачи:

- 1) Изучить литературу о задаче двух тел и методе дополнительных переменных.
- 2) Построить полиномиальную систему для решения уравнения Кеплера
- 3) Реализовать алгоритм решения полиномиальных систем методом рядов Тейлора в среде MATLAB для решения уравнения Кеплера
- 4) Сравнить полученные результаты с результатами метода Ньютона
- 5) Реализовать этот же алгоритм для решения задачи двух тел.

## Обзор литературы

Во время написания данной работы использовалась следующая научная и учебно-методическая литература.

Для ознакомления с задачей двух тел в первой главе были взяты книги [3] и [4].

Системы полиномиальных уравнений в частных производных для задачи двух тел были взяты из [2], [5].

Теоретическая информация по использованию метода Тейлора была взята из работы [1].

Полиномиальная система и схема алгоритма рядов Тейлора для задачи двух тел была взята из выпускной квалификационной работы магистра Дмитриевой Анны [6].

# Глава 1. Задача двух тел

## 1.1 Постановка задачи

Задача двух тел состоит в определении их движения под действием гравитационных сил взаимного притяжения, с учетом что никакие другие силы не воздействуют на эти тела. В данной работе вместо тел рассматривается движение двух материальных точек, расположенных в центре масс соответствующего тела и обладающих его массой. За начало системы отчета была взята одна из этих материальных точек, таким образом в этой работе рассматривается относительное движение.

## 1.2 Кеплеровы элементы орбиты

В задаче двух тел положение тела в пространстве может определяться с помощью Кеплеровых элементов орбиты. Они же, вместе с эксцентрической аномалией, используются для записи решения задачи двух тел. Этими элементами являются

- большая полуось,
- эксцентриситет,
- наклонение,
- долгота восходящего узла,
- аргумент перицентра,
- средняя аномалия.

Форма орбиты движения тела определяется первыми двумя элементами. Третий, четвертый и пятый отвечают за ориентацию плоскости орбиты по отношению к базовой системе координат. Шестой элемент определяет положение тела на орбите.

1. Большая полуось. В случае эллиптической орбиты, большая полуось положительна и равняется половине расстояния от перицентра до апоцентра.

2. Эксцентриситет – числовая характеристика конического сечения.

Эксцентриситет может быть выражен формулой:  $\varepsilon = \sqrt{1 - \frac{b^2}{a^2}}$ , где  $a$  и  $b$  – большие и малые полуоси орбиты соответственно.

Эксцентриситет определяет форму орбиты:

- 0 – окружность
- От 0 до 1 – эллипс
- 1 – парабола
- От 1 до бесконечности – гипербола, а малая полуось – мнимое число
- Бесконечность – прямая

3. Наклонение – угол между плоскостью орбиты тела и плоскостью отсчета (базовой плоскостью).

4. Долгота восходящего узла. Определяет угол в базовой плоскости, образуемый между базовым направлением на нулевую точку и направлением на точку восходящего узла орбиты, в которой орбита пересекает базовую плоскость в направлении с юга на север.

5. Аргумент перицентра – угол между направлениями из притягивающего центра на восходящий узел орбиты и на перицентр (ближайшую к притягивающему центру точку орбиты небесного тела), или угол между линией узлов и линией апсид.

6. Средняя аномалия – угловое расстояние от перицентра гипотетического тела, движущегося с постоянной угловой скоростью, равной среднему движению.

### 1.3 Уравнения движения задачи двух тел и их решения.

Рассмотрим уравнения движения точки массы  $m$  в центральном Ньютоновском поле массы  $m^0$ , используя относительную декартову систему координат с центром в точке с массой  $m^0$ :

$$\ddot{\xi}_i = -\mu \xi_i r^{-3} \quad (\text{или } \dot{\xi}_i = \eta_i \quad \dot{\eta}_i = -\mu \xi_i r^{-3}) \quad i \in [1 : 3]$$



И решения этой системы в общем виде:

$$\xi_i/a = A_i\sqrt{1 - e^2} \sin E + B_i(\cos E - e), i \in [1 : 3], r/a = (1 - e \cos E),$$

$$A_1 = -\sin \omega \cos \Omega - \cos \omega \sin \Omega \cos i, B_1 = \cos \omega \cos \Omega - \sin \omega \sin \Omega \cos i,$$

$$A_2 = -\sin \omega \sin \Omega - \cos \omega \cos \Omega \cos i, B_2 = \cos \omega \sin \Omega - \sin \omega \cos \Omega \cos i,$$

$$A_3 = \cos \Omega \sin i, B_3 = \sin \omega \sin i,$$

$$E - e \sin E = M, M = M_0 + n(t - t_0), n = \sqrt{\frac{\mu}{a^3}}, \mu = \gamma(m^0 + m)$$

где  $a$  (большая полуось),  $e$  (эксцентриситет),  $M_0$  (средняя аномалия в момент  $t_0$ ),  $\Omega$  (долгота восходящего узла),  $i$  (наклонение),  $\omega$  (аргумент перицентра) – Кеплеровы элементы орбиты;  $E$  (эксцентрическая аномалия),  $M$  (средняя аномалия) – функции времени, а  $\gamma$  - гравитационная постоянная.

В этой системе уравнение  $E - e \sin E = M$  – уравнение Кеплера.

## Глава 2. Система дифференциальных уравнений для решения уравнения Кеплера.

### 2.1 Полиномиальные системы

Таблица 1 – Функции и аргументы полиномиальной системы для решения уравнения Кеплера.

N	Функции	Аргументы
1	$\varphi_1 = E, \varphi_2 = \sin E; \varphi_3 = \cos E; \varphi_4 = (1 - e \cos E)^{-1};$	$\tau_1 = e; \tau_2 = M;$

Будем рассматривать функции  $\varphi_i, i = 1, \dots, 4$  как функции от  $t$ . Теперь выпишем системы дифференциальных уравнений в частных производных.

Полиномиальная система:

$$\begin{aligned} \frac{\partial \varphi_1}{\partial \tau_1} = \varphi_2 \varphi_4, \quad \frac{\partial \varphi_1}{\partial \tau_2} = \varphi_4, \quad \frac{\partial \varphi_2}{\partial \tau_1} = \varphi_2 \varphi_3 \varphi_4, \quad \frac{\partial \varphi_2}{\partial \tau_2} = \varphi_3 \varphi_4, \quad \frac{\partial \varphi_3}{\partial \tau_1} = -\varphi_4 \varphi_2^2, \quad \frac{\partial \varphi_3}{\partial \tau_2} = \\ -\varphi_2 \varphi_4, \quad \frac{\partial \varphi_4}{\partial \tau_1} = \varphi_3 \varphi_4^2 - \tau_1 \varphi_4^3 \varphi_2^2, \quad \frac{\partial \varphi_4}{\partial \tau_2} = -\tau_1 \varphi_4^3 \varphi_2 \end{aligned} \quad (1)$$

### 2.2 Подготовка к применению алгоритма методов рядов Тейлора

Подробный алгоритм метода рядов Тейлора был описан в выпускной квалификационной работе магистра Дмитриевой Анны. Для реализации данного алгоритма для решения уравнения Кеплера была составлена схема S мономов системы (1).

Таблица 2 - Таблица всех мономов системы (1) и соответствующая им схема.

№	Одночлен	S
5	$\varphi_2\varphi_4$	2, 4
6	$\varphi_3\varphi_4$	3, 4
7	$\varphi_4^2$	4, 4
8	$\varphi_2\varphi_3\varphi_4$	2, 6
9	$\varphi_2^2\varphi_4$	2, 5
10	$\varphi_3\varphi_4^2$	4, 6
11	$\varphi_2\varphi_4^3$	5, 7
12	$\varphi_2^2\varphi_4^3$	9, 7

Также была составлена матрица  $A = \{a_{j,k,v} \mid j = \overline{1, n}, k = \overline{1, u}, v = \overline{1, s}\}$ , которая состоит из нулей за исключением элементов:

$$\begin{aligned}
 a_{1,5,1} = 1, a_{1,4,2} = 1, a_{2,8,1} = 1, a_{2,6,2} = 1, a_{3,9,1} = -1, \\
 a_{3,5,2} = -1, a_{4,10,1} = 1, a_{4,12,1} = -t_1, a_{4,11,2} = -t_1 \quad (2)
 \end{aligned}$$

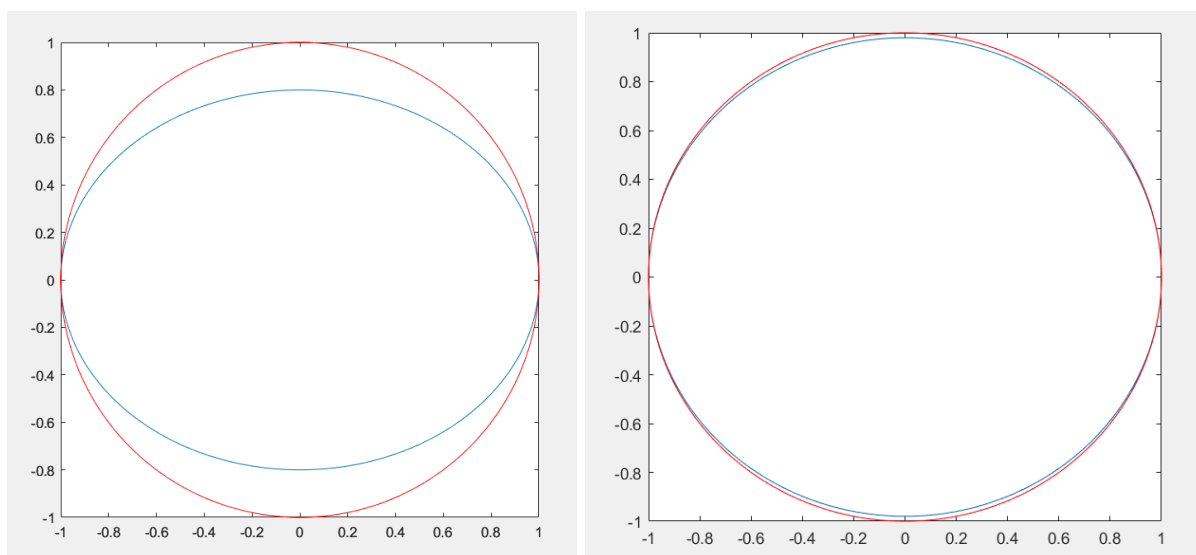
## Глава 3. Реализация алгоритма

### 3.1 Описание программы

Программа реализована с помощью математического пакета MatLab. Функция `x_j_1_calc` рассчитывает новые производные, функция `step(x0,A)` рассчитывает шаг для каждой новой точки. Основной код содержится в функции `main`. Составленная программа является универсальной и подходит как для решения уравнения Кеплера, так и для решения уравнений задачи двух тел с тем учетом, что необходимо заменить соответствующие матрицы  $S$  и  $A$  алгоритма Тейлора для конкретной системы. Для ускорения работы программы часть выкладок по поиску шага для конкретных задач была сделана аналитически.

### 3.2 Результаты решения уравнения Кеплера

На рисунке 1 представлены результаты работы программы для решения уравнения Кеплера описанным методом с 5 членами ряда Тейлора. Синим цветом на графиках обозначена орбита при эксцентриситете  $e=0.6$  и  $e=0.2$  соответственно. Красным цветом очерчен круг ( $e=0$ ) для сравнения.



*Рисунок 1 – Орбиты тела при разных эксцентриситетах*

Далее на рисунке 2 представлен подсчет эксцентрической аномалии  $E$  для разных эксцентриситетов.

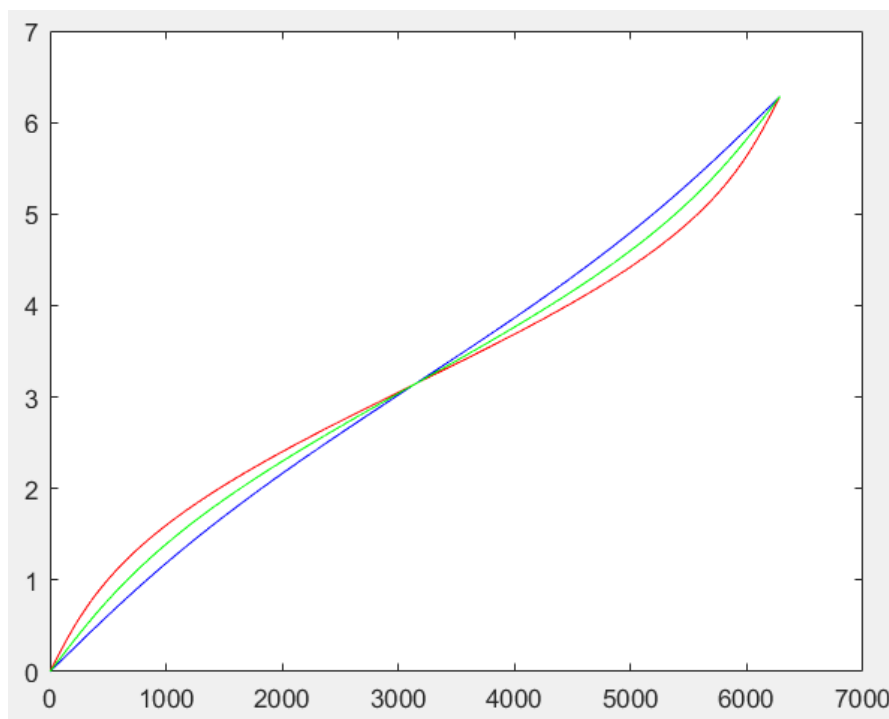


Рисунок 2 – Эксцентрисическая аномалия при разных эксцентриситетах.  
 Синяя линия – аномалия при  $e = 0,2$ , зеленая линия – аномалия при  $e = 0,4$ ,  
 красная линия – аномалия при  $e = 0,6$

Так как в общем случае для уравнения Кеплера не существует аналитического решения, для сравнения точности результатов был взят метод Ньютона точности  $10^{-20}$ . Результаты работы программы для разного количества членов ряда Тейлора представлено на рисунке 3.

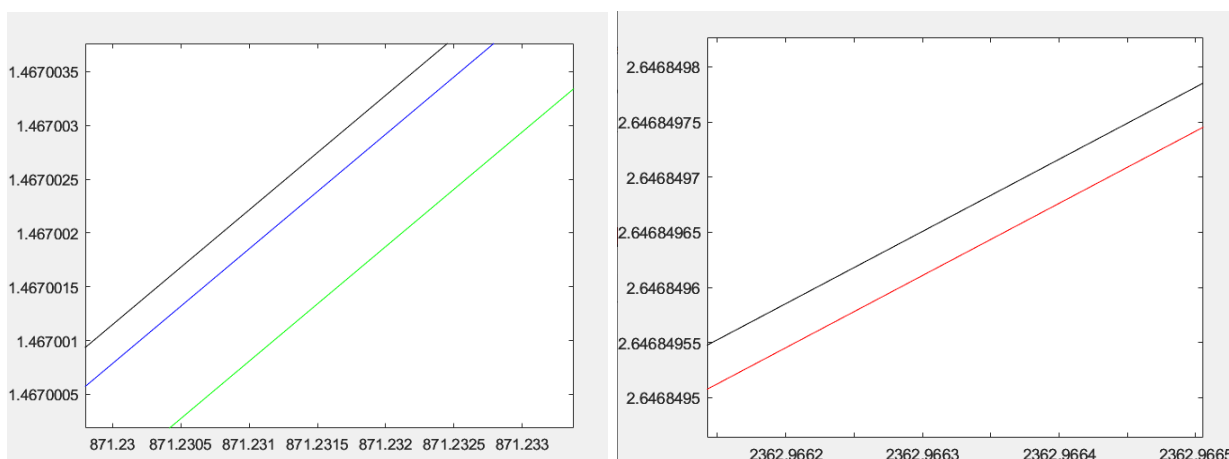


Рисунок 3 – Точность решения уравнения Кеплера для разного количества членов ряда Тейлора. Синяя линия – решение при 3 членах ряда Тейлора, зеленая линия – решение при 2 членах ряда, красная линия – решение при 5 членах ряда, черная линия – решение методом Ньютона.

Таким образом, для решения уравнения Кеплера программа дает достаточно точные результаты: для решения построенного на основе 2 членов ряда Тейлора отклонение от решения методом Ньютона было порядка  $10^{-6}$ , для решения полученного методом Тейлора с 5 членами ряда отклонение составило  $10^{-8}$ . Исходя из полученных результатов, для решения задачи двух тел будет использоваться метод рядов Тейлора с 4 членами ряда, в силу достаточной его точности.

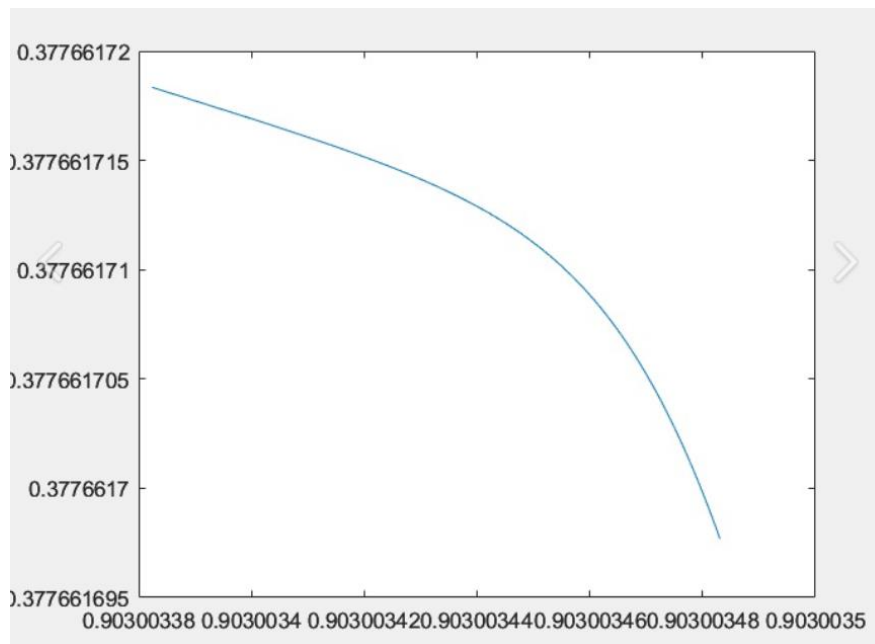
### 3.3 Результаты решения задачи двух тел

Были заданы следующие начальные условия:

Таблица 3 – Начальные условия для решения уравнений задачи двух тел

Элемент орбиты	Значение
$\mu$	$9.8*(1.9889169e+10 + 1.89863e+2)$
$t$	0
$a_0$	0.778194564622574
$e$	0.048425472281527694
$M_0$	0
$i$	1.305185101881517
$\Omega$	1.049431512298086
$\omega$	1.736540415423883

В программной реализации был использован метод рядов Тейлора с 4 членами ряда, число шагов составило 700. Результаты представлены на рисунке 4.



*Рисунок 4 – часть полученной орбиты по первым двум координатам системы  $\varphi_8$  и  $\varphi_9$ .*

На графике выведена часть орбиты по первой и второй координате. Можно заметить, что благодаря маленькому шагу и высокой точности метода Тейлора график орбиты получается очень гладким.

## **Заключение.**

В ходе работы была изучена литература о задаче двух тел интегрировании систем дифференциальных уравнений методом Тейлора, построена полная полиномиальная система уравнений в частных производных для решения уравнения Кеплера, проведена программная реализация алгоритма в среде MatLab и сравнение полученных результатов решения уравнения Кеплера с методом Ньютона точности  $10^{-20}$ . На основе программы для решения уравнения Кеплера была реализована программа для решения уравнения задачи двух тел.



## Литература

[1] Л.К. Бабаджанянц, И.Ю. Потоцкая, Ю.Ю. Пупышева Оценки в методе рядов Тейлора для линейных полных УрЧП // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2020. Т. 16. Вып. 2. С. 00–00.

[2] Бабаджанянц, Л. К., Брэгман, А. М., Брэгман, К. М., Касикова, П. В., & Петросян, Л. А. (2016). Полные системы уравнений для задачи двух тел. ТЕХНИЧЕСКИЕ НАУКИ - ОТ ТЕОРИИ К ПРАКТИКЕ, 8(56), 13-20.

[3] Холшевников К.В., Титов В.Б. Задача двух тел: Учеб. пособие. – СПб., 2007. – 180 с.

[4] Емельянов Н. В. Практическая небесная механика. – М.: Физический факультет МГУ, 2018. 270 с.

[5] Брэгман А.М. Движение тела, управляемого малой тягой в поле Ньютона: Магистерская диссертация. СПб., Санкт-Петербургский Государственный университет, 2014, 145 с.

## Приложение 1

*Функция all\_Q(r, s)*

```
function sum = all_Q(r, s)
if r<0
    sum = 0;
else
if r==0
    sum=1;
else
    sum = Q(r, s)+all_Q(r-1, s);
end
end
end
```

## Приложение 2

Функция *find\_mu(s, indx, d\_x)*

```
function Q = find_mu(s, indx, d_x)
Q = [];
l = d_x(indx).l_vec;
for i=1:indx
    temp = [d_x(i).l_vec];
    c=0;
    for nu=1:s
        if temp(nu)>l(nu)
            c = c+1;
        end
    end
    if c==0
        Q = [Q i];
    end
    temp = [];
end
```

## Приложение 3

### Функция main

```
clc;
clear all;
%% Начальные данные
r_max = 5; %глубина дифференцирования
u = 12;
s = 2;
n = 4;
t0 = [0.6, 0];
x0 = zeros(1, n);
x0(2) = sin (x0(1));
x0(3) = cos (x0(1));
x0(4) = (1 - t0(1)*cos(x0(1)))^(-1);

%e_true = zeros(1,6285);
%e_true(1)=0;
%e_temp = 0.1;
%% Шаг
dt = [0, 0.001];
%% Вычисление x в новой точке
c = 2*pi/dt(2);
res_x = zeros(fix(c)+2, n);
res_x(1, :) = x0;

for i = 2:fix(c)+2
    temp = x_calc(x0, dt, r_max, s, u, n, t0);
    res_x(i, :) = temp;

    x0 = temp;
    t0 = t0+dt;

    %flag = 1;
    %while flag>10^(-10)
    %e_true(i) = e_temp + ((t0(2)+0.6*sin(e_temp)-e_temp))/(1-0.6*cos(e_temp));
    %flag = e_temp - e_true(i);
    %e_temp=e_true(i);
    %end
end

%% Результат
x = res_x(:, 3);
y = sqrt(1 - t0(1)^2)*res_x(:, 2);

%figure;
%plot (x, y);
%hold on;
%plot (res_x(:, 3), res_x(:, 2), 'r')
%hold on;

figure;
plot (res_x(:, 1),'r')
hold on;

r_max = 5; %глубина дифференцирования
u = 12;
```

```

s = 2;
n = 4;
t0 = [0.2, 0];
x0 = zeros(1, n);
x0(2) = sin (x0(1));
x0(3) = cos (x0(1));
x0(4) = (1 - t0(1)*cos(x0(1)))^(-1);

%e_true = zeros(1,6285);
%e_true(1)=0;
%e_temp = 0.1;
%% Шаг
dt = [0, 0.001];
%% Вычисление x в новой точке
c = 2*pi/dt(2);
res_x = zeros(fix(c)+2, n);
res_x(1, :) = x0;

for i = 2:fix(c)+2
    temp = x_calc(x0, dt, r_max, s, u, n, t0);
    res_x(i, :) = temp;

    x0 = temp;
    t0 = t0+dt;

    %flag = 1;
    %while flag>10^(-10)
    %e_true(i) = e_temp + ((t0(2)+0.6*sin(e_temp)-e_temp))/(1-0.6*cos(e_temp));
    %flag = e_temp - e_true(i);
    %e_temp=e_true(i);
    %end
end

plot (res_x(:, 1),'b')
hold on;

r_max = 5; %глубина дифференцирования
u = 12;
s = 2;
n = 4;
t0 = [0.4, 0];
x0 = zeros(1, n);
x0(2) = sin (x0(1));
x0(3) = cos (x0(1));
x0(4) = (1 - t0(1)*cos(x0(1)))^(-1);

%e_true = zeros(1,6285);
%e_true(1)=0;
%e_temp = 0.1;
%% Шаг
dt = [0, 0.001];
%% Вычисление x в новой точке
c = 2*pi/dt(2);
res_x = zeros(fix(c)+2, n);
res_x(1, :) = x0;

for i = 2:fix(c)+2
    temp = x_calc(x0, dt, r_max, s, u, n, t0);
    res_x(i, :) = temp;

    x0 = temp;

```

```
t0 = t0+dt;

%flag = 1;
%while flag>10^(-10)
%e_true(i) = e_temp + ((t0(2)+0.6*sin(e_temp)-e_temp))/(1-0.6*cos(e_temp));
%flag = e_temp - e_true(i);
%e_temp=e_true(i);
%end
end

plot (res_x(:, 1),'g')
hold on;
```

## Приложение 4

*Функция  $Q(r,s)$*

```
function sizeQ= Q(r,s)
if r==0
    sizeQ=1;
else
    if r==1
        sizeQ=s;
    else
        sizeQ = 0;
        for k=1:s
            sizeQ = sizeQ+Q(r-1,k);
        end
    end
end
end
```

## Приложение 5

Функция  $Ql(r,s)$

```
function [Ql] = Ql(r,s)

if r==0
    Ql = zeros(1,s+1);
    Ql(s+1) = 1;
else

Qfirst=zeros(Q(1,s),s+1);
for i = 1:s
    Qfirst(i,i)=1;
    Qfirst(i,s+1)=i;
end
if r == 1
    Ql=Qfirst;
else
for r_flag=2:r
Qnext = zeros(Q(r_flag,s),s+1);
h=1;
for j=1:Q(r_flag-1,s)
    for i = Qfirst(j,s+1):s
        Qnext(h,:)=Qfirst(j,:);
        Qnext(h,i)=Qnext(h,i)+1;
        Qnext(h,s+1)=i;
        h=h+1;
    end
end
Qfirst = zeros(Q(r_flag,s),s+1);
Qfirst = Qnext;
end

Ql=Qnext;
end
end
end
```



## Приложение 6

*Функция step(x0,A)*

```
function step = step(x0,A )

gamma = max(abs(x0));
step = zeros(1,4);
step(1)=1/(11*gamma^6*abs(A(4,61,1)));
step(2)=1/(11*max(gamma^8*(abs(A(4,71,2))), gamma^2*abs(A(5,42,2))));

t3m = [ gamma^2*abs(A(2,37,3)) , gamma^2*abs(A(4,40,3))+gamma^4*abs(A(4,50,3)),
gamma^2*abs(A(7,43,3)) ];
step(3)=1/(11*max(t3m));
step(4)=1/(11*gamma^3*abs(A(4,44,4)));

end
```

## Приложение 7

Функция  $x\_j\_l\_calc(j, indx, Q, d\_x)$

```
function sum = x_j_l_calc(j , indx, Q, d_x)
    sum = 0;
    [p, q] = S(j);
    l = d_x(indx).l_vec;
    Q_size = length(Q);
    for i = 1:Q_size
        mu = d_x(Q(i)).l_vec;
        mu2 = d_x(Q(Q_size - i+1)).l_vec;
        x1 = d_x(Q(i)).x_vec;
        x2 = d_x(Q(Q_size - i+1)).x_vec;
        sum = sum +
        (prod(factorial(l))/prod(factorial(mu))/prod(factorial(mu2)))*x1(p)*x2(q);
    end
end
```