Санкт-Петербургский государственный университет

**Кафедра механики управляемого движения**

**Саакян Артур Темиевич**

**Магистерская диссертация**

**Метод рядов Тейлора для дифференциальных**

**уравнений динамических моделей**

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа математическое моделирование в задачах естествознания

Научный руководитель,  
доктор физ.-мат. наук,  
профессор  
Бабаджанянц Л. К.

Санкт-Петербург

2017

Оглавление

**Введение 3**

0.1 Обзор работы и использованная литература3

0.2Сведение уравнений динамики к полиномиальной системе 4

0.3Пошаговый метод рядов Тейлора 6

0.4Кусочно-полиномиальное управление движением тела 10

**Глава 1. Постановка задачи 11**

1.1 Обозначения и определения 11

1.2 Формулировка задачи 11

**Глава 2. Построение минимальной оболочки 12**

**Глава 3. Сведение уравнений динамики к полиномиальной форме 13**

3.1 Задача N тел 13

3.2 Возмущенная и невозмущенная задача двух тел 15

**Глава 4. Численные эксперименты. Задача N тел 18**

**Глава 5. Численное интегрирование 21**

5.1 Возмущенная и невозмущенная задача двух тел 21

5.2 Задача N тел. Сравнение полиномиальных систем разного порядка22

**Заключение и Выводы 24**

**Список литературы 25  
Приложения I, II, III 26**

**Приложение I: Программа построения минимальной оболочки и схемы26**

**Приложение II: Задача N тел. Время вычисления правых частей системы 27**

**А. Время вычисления правых частей системы пятой степени 27**

**B. Время вычисления правых частей системы четвертой степени31**

**C. Время вычисления правых частей системы третьей степени 35**

**D. Время вычисления правых частей исходной системы 38**

**Приложение III. Программа оценки эффективности схем 39**

**Введение**

Метод рядов Тейлора является одним из самых популярных численных методов для решения обыкновенных дифференциальных уравнений динамики, которые можно свести к полиномиальной форме. В главах 1, 2 и 3 рассмотрены основные идеи ускорения численного интегрирования полиномиальных систем. В главах 4 и 5 проводятся непосредственно численные эксперименты и численное интегрирования, на примере задачи N тел.

* 1. **Обзор работы и использованная литература**

В разделе 0.2 рассматриваются разновидность и особенности классов уравнений и функций, при помощи которых, можно свести дифференциальные уравнения к полиномиальной форме. В разделе 0.3 рассматривается пошаговый метод решения обыкновенных дифференциальных уравнений методом рядов Тейлора. Магистерская диссертация состоит из введения, глав 1, 2, 3, 4, 5, выводов, списка использованной литературы и приложении. В параграфах 0.2 и 0.3 обосновывается актуальность обыкновенных дифференциальных уравнений в полиномиальной форме в задачах динамики, и описывается метод Рядов Тейлора для таких систем уравнений.

**Глава 1** содержит необходимые обозначения и определения, необходимые для формулировки поставленной задачи. Первая глава состоит из двух параграфов. В первой параграфе описаны такие понятия, как моном, степень монома и схема. Во втором параграфе сформулирована постановка задачи.

В **главе 2** представлено решение поставленной задачи, которую удалость свести к задаче бинарного линейного программирования.

**Глава 3** состоит из двух параграфов. В первом параграфе, при помощи ряда замен уравнение задачи N тел сводится к полиномиальной форме. Во втором параграфе сводится к полиномиальной форме задача двух тел с кусочно-полиномиальным управлением.

Использовалась литература: [1 -2].

В **главе 4** представлены численные эксперименты, показывающие эффективность использования схем для численного интегрирования полиномиальных систем.

Использовалась литература: [4 -5].

**Глава 5** состоит из двух параграфов. В первом параграфе рассмотрена задача двух тел с кусочно-полиномиальным управлением. Результаты ее численного интегрирования представлены в Таблице IV. Во втором параграфе рассмотрена задача трех внешних планет. Результаты ее численного интегрирования представлены в Таблице V.

Использовалась литература: [6 - 18].

**0.2 Сведение уравнений динамики к полиномиальной системе**

Материал, представленный ниже, взят из статьи [1 - 2].

Рассматривается полная система дифференциальных уравнений в частных производных первого порядка, разрешенная относительно производных:

, , , (1)

, ,

где , , ,

, , , , .

При данная система является системой обыкновенных дифференциальных уравнений вида:

, , . (2)

Класс полиномов по с коэффициентами, зависящими от параметра (возможно), обозначим или . Объединение этих классов по обозначим или . Система (1), где все функции называют полиномиальной системой.

**Классы функций**

Функция от аргумента удовлетворяет полиномиальной системе, если эта скалярная функция является одной из компонент вектор - функции (того же аргумента - решения некоторой полиномиальной системы). Класс – это множество скалярных функций аргумента , которые удовлетворяют полиномиальным системам. Если , тогда любую функцию аргумента можно считать функцией аргумента класса , т.е. .

Класс скалярных функций от определим условием тогда и только тогда, когда найдется какое-то число k скалярных функций таких, что

, , , ,

причем функции – алгебраические полиномы по .

Класс , , – это множество скалярных функций аргумента , которые можно получить из при помощи конечного числа операций и функций и их суперпозиций. Запись вида: означает, что величину можно получить как последний член последовательности , в которой величина , либо равна при каких-то , , либо является полиномом по переменным .

Класс - множество таких функций от аргумента , которые задаются формулой:

*, ;*

,, ,

где ; ; ;

; ; , а .

**Метод дополнительных переменных (формулировка)**

При помощи этого метода можно сводить систему вида (1) к системе с полиномиальными правыми частями. Необходимо найти такой набор дополнительных переменных – , для которых выполнено:

1. правыe части системы уравнений (1) – полиномы от

;

1. все производные для дополнительных переменных

в силу уравнений (1), тоже являются полиномами по

.

**0.3 Пошаговый метод рядов Тейлора**

Пошаговый метод численного интегрирования задачи Коши (1), (2) методом рядов Тейлора позволяет получать приближенные значения - решения в точках начальной задачи:

*,* (1)

, (2)

Числа называют узлами таблицы, а - шагами интегрирования. Сеткой называют совокупность узлов. Если , то говорят об интегрировании с фиксированным шагом, либо о равномерной сетке. Величины есть значения решения на узлах сетки.

Предположим, что права часть уравнения (1) голоморфна относительно переменных в окрестности начальной точки , тогда единственное решение задачи (1), (2) представляется рядом Тейлора с ненулевым радиусом сходимости .

Пусть

,

,, (3)

где – радиус сходимости ряда , а - круг его сходимости.

Метод рядов Тейлора заключается в построении таблицы приближенных значений

,

… …. …

, (4)

и т.д.,

где , , а для чисел выполняется:

. (5)

Величина - порядок метода на к -м шаге. Для вычисления , необходимо вычислить , .

Рассмотрим два способа рекуррентного нахождения коэффициентов Тейлора вектор - функции, удовлетворяющей полиномиальной системе.

**1-й способ:**

, (6)

, , (7)

где , (8)

; ;

; - комплекснозначные функции аргумента , голоморфные в окрестности точки . Пусть – последовательность коэффициентов Тейлора функции в её разложении по степеням , а - аналогичная последовательность для функции .

Введем обозначения:

, . (9)

Обозначим , как коэффициент при полинома по степеням . Вычисление этого полинома состоит из умножений и сложений полиномов; для получения его k-ого члена включительно при выполнении этих операций достаточно удерживать каждый раз в результате члены до k-ого порядка включительно.

Если подставить в (6) разложение Тейлора функций , привести подобные члены и приравнять к нулю все коэффициенты полученного степенного ряда, то из полученных таким образом уравнений можно получить следующие рекуррентные соотношения:

,

; , .

**2-й способ:** Нахождение явных рекуррентных формул для . Ограничимся рассмотрением квадратичной задачи:

, , (10)

где – комплексные постоянные. Подставляем в (10) разложения , приводя подобные члены и приравнивая к нулю коэффициенты полученного степенного ряда, получаем искомые формулы

;

; , , .

**0.4 Кусочно-полиномиальное управление движением тела**

Будем рассматривать кусочно-полиномиальные управления

,

которые представлены в виде:

где , а - функция Хевисайда.

**Глава 1. Постановка задачи**

* 1. **Обозначения и определения**

*Мономом* называют выражение вида , , ,

- неотрицательное, целое число.

*Степенью монома* называют величину .

Набор будем называть *упорядоченным*, если .

*Схемой* назовём набор из пар натуральных чисел и , , такой, что из , : при , .

Если набор не имеет схемы, то его можно дополнить таким набором мономов, что полученный набор будет иметь схему.

* 1. **Формулировка задачи**

Пусть - упорядоченные наборы мономов. Тогда будем называть *оболочкой* для , если , и имеет схему.

*Минимальной оболочкой* для набора мономов будем называть оболочку с минимальным количеством мономов.

**Постановка задачи**: Построить минимальную оболочку для любого заданного набора мономов.

**Глава 2. Построение минимальной оболочки**

В данной главе предложен способ построения минимальной оболочки для мономов 3-й степени (). Поставленную задачу удалось свести к минимизации в рамках задачи бинарного программирования, так как переменные имеют ограничения вида: .

Задача состоит во введении по возможности меньшего количества таких мономов второй степени, чтобы любой несогласованный моном третьей степени исходного набора содержал по крайней мере один из них.

Каждый моном третьей степени содержит до трех мономов второй степени. Например, степень (3,0,0,…,0) кубического монома сoдержит oдин квадратичный степени (2,0,…,0), кубический моном степени (2,1,0,…,0) содержит два монома степеней (2,0,…,0) и (1,1,0,…,0) , а моном степени (1,1,1,0,…,0) - три монома степени (1,1,0,…,0), (1,0,1,0,…,0) и (0,1,1,0,…,0).

Задача бинарного линейного программирования для поставленной задачи имеет вид:

при известных , ,

найти целые ,

удовлетворяющие неравенствам

, ; ,

и доставляющие минимум сумме ,

где - количество квадратичных подиндексов для кубического индекса, – различные квадратичные индексы для всего набора кубических индексов. Каждой величине сопоставляем переменную , которая может принимать два значения – 0 и 1.

**Глава 3. Сведение уравнений динамики**

**к полиномиальной форме**

**3.1 Задача N тел**

Пусть массы материальных точек соответственно движутся по закону Ньютона, под действием взаимного притяжения; – прoизвольная инерциальная система координат; имеет координаты (в данной инерциальной системе). Рассмотрим относительные координаты с началом в точке с oсями , кoторые являются oдинаково нaправленными с . Пусть – кoординаты тoчки в дaнной сиcтеме. Тoгда . Функции удовлетворяют ОДУ:

(1)

где , , , k – постоянная Гаусса.

Сведем систему (1) к полиномиальной системе пятой, четвертой и третьей степени [1 - 4].

*Полиномиальная система пятой степени.*

Введем новые переменные: , ,

где , ,, , получаем полиномиальную систему относительно переменных , , . Система имеет вид:

*,*

*,* (2)

*, , , .*

Система (2) содержит ниже перечисленные нелинейные мономы:

*, , , ,* при  *, , .* (3)

Набор (3) не имеет схемы, поэтому необходимо построить оболочку для данного набора. Для получения минимальной оболочки для набора (3), необходимо добавить мономы: , .

*Полиномиальная система четвертой степени.*

Вводя дополнительные переменные

, ,

где , ,, , получаем полиномиальную систему четвертой степени относительно переменных , , , , :

*,*

*,  
 ,  
,* (4)

) ].

Система (4) содержит нелинейные мономы:

, , , , , , , , , , , , (5)

Набор (5) не имеет схемы, поэтому необходимо построить оболочку для данного набора. Для получения минимальной оболочки для набора (5), необходимо добавить мономы: .

*Полиномиальная система третьей степени.*

Введем новые переменные: , где , ,, , получаем полиномиальную систему относительно переменных , , , , , . Система имеет вид:

*,*

*,*

*,*

*,* (6)

*,*

) ].

Система (6) содержит нелинейные мономы:

, , , , , , , ,

, , , , , . (7)

Набор (7) имеет схему.

* 1. **Возмущенная и невозмущенная задача двух тел**

Рассматривается возмущенная (управлением – малой тягой) задача двух тел

, (8)

и невозмущенная задача двух тел

, , (9)

и общие начальные условия

, , , (10)

здесь - истинная аномалия в невозмущенном эллиптическом движении, , – постоянная Гаусса, - компоненты ускорения от этой тяги. Мы хотим переписать эти уравнения в терминах производных по вместо . Для этого надо вторые производные в (8), (9) переписать через производные (первого и второго порядка) по , а также переписать соответствующим образом начальные условия (10). Так как

, , ,

тогда для любой функции последовательно получаем:

,

.

Теперь перепишем (8) - (10) в искомом виде:

,

*Переход к полиномиальной системе.*

*Невозмущенная задача двух тел* имеет решение в аналитическом виде, поэтому нет необходимости переходить к полиномиальной системе в этом случае. Для вычисления необходимо знать начальные данные, которые вычисляем по формулам:

, , ,

;

,

, ;

,

, ;

*Возмущенная задача двух тел.*

;

Введем новые переменные:

, , ,

, ;

получаем полиномиальную систему относительно переменных:

, , , , , , .

Система имеет вид:

,

,

, ,

, .

**Глава 4. Численные эксперименты. Задача N тел.**

В данной главе, мы рассмотрим три численных эксперимента [4-5]. В первых двух из них сравнивается время вычисления правых частей дифференциальных уравнений без схемы и со схемой. В третьем эксперименте, мы выполняем те же вычисления и сравнения для произвольного набора мономов. Расчеты проводились с использованием шести программ, написанных на языке Wolfram Mathematica: *Mono3*, *Nbody5, Nbody4*, *Nbody3*, *RHS* и *RandomM*; Результаты всех трех экспериментов представлены в табличной форме с необходимыми пояснениями.

*Результаты численного эксперимента № 1*

Здесь используются следующие обозначения:

– количество тел;  
 – процессорное время, требуемое для вычисления правых частей уравнений (1), (2), (4) и (6) соответственно;  
 – процессорное время, требуемое для вычисления при помощи схем правых частей уравнений (1), (2), (4) и (6) соответственно;  
 – процессорное время, требуемое для вычисления при помощи схем правых частей уравнений (1), (2), (4) и (6) соответственно;

ТАБЛИЦА I

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 3 | 7.2 | 8.6 | 4.2 | 5.0 | 4.2 | 5.5 |
| 4 | 9.6 | 15.4 | 4.7 | 7.0 | 4.6 | 7.8 |
| 5 | 9.7 | 20.4 | 4.8 | 11.5 | 4.7 | 14.1 |
| 6 | 9.8 | 22.5 | 4.9 | 14.2 | 4.8 | 16.3 |
| 7 | 10.2 | 25.5 | 5.2 | 17.7 | 5.1 | 19.3 |
| 8 | 12.1 | 38.7 | 5.3 | 20.1 | 5.3 | 22.8 |
| 9 | 13.0 | 49.9 | 5.6 | 23.0 | 5.5 | 25.8 |
| 10 | 13.1 | 57.6 | 5.8 | 26.6 | 5.7 | 29.6 |

*Результаты численного эксперимента № 2*

Здесь используются следующие обозначения:

– количество тел;  
 – процессорное время, требуемое для вычисления всех добавленных мономов третьей степени и всех мономов лежащих в правых частях уравнений (2) (), (4) () и (6) () (с использованием и без использования схемы).

ТАБЛИЦА II

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 3 | 1.2 | 1.2 | 1.2 | 1.1 | 1.3 | 1.2 |
| 4 | 1.6 | 1.5 | 1.5 | 1.5 | 1.7 | 1.6 |
| 5 | 2.1 | 2.0 | 2.4 | 2.3 | 3.0 | 2.7 |
| 6 | 2.3 | 2.1 | 2.9 | 2.7 | 3.4 | 3.1 |
| 7 | 2.6 | 2.2 | 3.4 | 3.0 | 3.8 | 3.3 |
| 8 | 3.2 | 2.6 | 3.8 | 3.3 | 4.3 | 3.6 |
| 9 | 3.8 | 3.0 | 4.1 | 3.5 | 4.7 | 3.8 |
| 10 | 4.4 | 3.2 | 4.6 | 3.8 | 5.2 | 4.1 |

*Результаты численного эксперимента № 3*

Здесь используются следующие обозначения:

– количество мономов;  
 – количество переменных;  
 – эффективность схемы, где – процессорное время, требуемое для вычисления случайного набора мономов, с использованием и без использования схем;  
 – минимальное и максимальное значение соответственно;

ТАБЛИЦА III  
ЭФФЕКТИВНОСТЬ СХЕМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | |  | | |
|  |  |  |  |  |  |
| 50 | 1.6 | 1.7 | 1 000 | 4.5 | 4.7 |
| 100 | 2.6 | 2.8 | 2 000 | 4.9 | 5.2 |
| 150 | 3.5 | 3.8 | 3 000 | 5.1 | 5.3 |
| 200 | 4.5 | 4.7 | 4 000 | 5.3 | 5.8 |
| 250 | 5.2 | 5.5 | 5 000 | 5.6 | 5.9 |
| 300 | 6.3 | 6.7 | 6 000 | 5.8 | 6.2 |
| 350 | 7.5 | 8.2 | 7 000 | 5.6 | 6.0 |
| 400 | 8.6 | 9.7 | 8 000 | 5.7 | 6.6 |
| 450 | 9.9 | 11.0 | 9 000 | 5.7 | 6.9 |
| 500 | 11.2 | 13.5 | 10 000 | 5.8 | 7.0 |

**Глава 5. Численное интегрирование**

**5.1 Возмущенная и невозмущенная задача двух тел**

Рассмотрим возмущенную малой тягой задачу двух тел в полиномиальной форме, представленную в главе 3:

,

,

, ,

, .

Управление будем рассматривать в виде:

,

где

– полиномы степени :

, , ;

, , ;

, , ;

В качестве примера рассмотрим астероид Апофис:

, , , , ,

(массы Солнца), .

Здесь используются следующие обозначения:

– степень полинома (управление);  
 – коэффициент представленный в управлении;  
 – требуемая относительная погрешность;  
 – максимальное значения относительной погрешности;  
 , ;

ТАБЛИЦА IV  
ВОЗМУЩЕННАЯ И НЕВОЗМУЩЕННАЯ ЗАДАЧА ДВУХ ТЕЛ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |

**5.2 Задача N тел. Сравнение полиномиальных систем разного порядка**

Рассмотри задачу N тел в полиномиальной форме [6-15], т.е. полиномиальную систему пятой степени (2), четвертой степени (4) и третьей степени (6). Будем интегрировать их используя программы: TSMR и TIDES.

Рассмотрим задачу о движении трёх внутренних планет (Солнце – Меркурий – Венера), интегрируем эту задачу программами: TSMR [15] и TIDES [16]. Интегрируем на промежутке , где = 1 млн. суток, 2 млн. суток и 3 млн. суток по схеме «туда – обратно», т.е. на каждом промежутке сначала интегрируем уравнения от точки до точки , а затем приняв полученные данные за начальные, интегрируем уравнения от точки до точки . Начальные данные и массы планеты взяты из [17-18]. Вычисления производились на компьютере с процессором Intel Core i5, 2,6 GHz, 8 GB и Mac OS X El Capitan (Version 10.11.5), а в качестве компиляторов использовались: Intel Parallel Studio XE 2016 для TSMR и GNU Compiler Collection (Xcode Version 7.3.1) для TIDES. Результаты представлены в Таблице V.

ТАБЛИЦА V  
ЧИСЛЕННОЕ ИНТЕГИРОВАНИЕ (СРАНИТЕЛЬНЫЕ ДАННЫЕ; )

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Метод* |  |  |  |  |  |
| TIDES\_5 |  |  |  |  | 551 1458 2915 |
| TSMR\_5 |  |  |  | 227 659 1369 |
| TIDES\_4 |  |  |  | 1261 3368 6173 |
| TSMR\_4 |  |  |  | 381 1088 2272 |
| TIDES\_3 |  |  |  | 1233 3244 6096 |
| TSMR\_3 |  |  |  | 407 1252 2230 |
| TIDES\_5 |  |  |  |  | 1101 3006 5701 |
| TSMR\_5 |  |  |  | 484 1311 2698 |
| TIDES\_4 |  |  |  | 2425 6266 11855 |
| TSMR\_4 |  |  |  | 777 2214 4315 |
| TIDES\_3 |  |  |  | 2519 6491 12163 |
| TSMR\_3 |  |  |  | 756 2279 4258 |
| TSMR\_5 |  |  |  |  | 649 1993 3977 |
| TSMR\_4 |  |  |  | 1494 3203 6579 |
| TSMR\_3 |  |  |  | 1130 3370 6409 |

**Заключение и Выводы**

Перечислим полученные результаты:

1. Был получен алгоритм построения схемы оптимального вычисления системы мономов третьей степени и написана соответствующая программа на языке Wolfram Mathematica.
2. Задача тел сведена к полиномиальной системе четвертой и третьей степени, найдены оптимальные схемы для правых частей этих задач.
3. Проведены численные эксперименты для задачи тел (), которые показывают эффективность перехода к полиномиальной системе и эффективность использования схем для полиномиальных систем.
4. Продемонстрирована численно зависимость эффективности вычислений от количества переменных и количества мономов в полиномиальных системах.
5. Проведены численные интегрирование для задачи трёх внешних планет и сравнение представленных полиномиальных систем.
6. Проведены численные эксперименты для невозмущенной задачи двух тел, при кусочно-полиномиальном возмущении.

**Список литературы**

1. Бабаджанянц Л. Метод дополнительных переменных // Вестник СПБГУ Серия 10, 2010. 3 - 11.
2. Бабаджанянц Л., Брэгман К. Алгоритм метода дополнительных переменных // Вестник СПБГУ Серия 10. Вып. 2. 2012. 3 - 12.
3. Бабаджанянц Л. К. Метод рядов Тейлора // Вестник Санкт- Петербургского университета. Серия 10: Прикладная математика. Информатика. Процессы управления. 2010. No 3. C. 13–29.
4. И.М. Алесова, Л.К. Бабаджанянц, И.Ю. Потоцкая, А.Т. Саакян «Оптимизация пошагового интегрирования дифференциальных уравнений динамики», Устойчивость и колебания нелинейных систем управления: Материалы XIII Международной конференции/ Ред. В.Н. Тхай. – М.: ИПУ РАН, 2016, 17-19.
5. Саакян А.Т. «Ускорения численного интегрирования уравнений динамики при помощи схем», Процессы управления и устойчивость. Том 3(19). №1/ Под ред. тома Н.В. Смирнов. Санкт-Петербург, 2016, с.250-254.
6. L. Babadzanjanz, D. Sarkissian, ”Taylor series method for dynamical systems with control”, Journal of Mathematical Sciences, Springer New York, 6, Vol. 139, 7025-7046. 2006.
7. Babadzanjanz L.K.On the global solution of the N-body problem // Celestial Mechanics and Dynamical Astronomy. 1993. 56. 427 – 449.
8. Babadzhanyants L.K. Error estimates for numerical integration of the N-body problem. American Institute of Physics (Sov.Astron.Lett.7(6) Nov.-dec.-1981), 1982. 416 – 418
9. Irina M. Alesova, Levon K. Babadzanjanz, Irina Yu. Pototskaya, Yulia Yu. Pupysheva, Artur T. Saakyan «Taylor Series Method of Numerical Integration of the N-body Problem » (Предполагается опубликовать)
10. L. K. Babadzanjanz, ”On the global solution of the N-body problem”, Celestial Mechanics and Dynamical Astronomy, Kluwer Academic Publishers, 56, 427-449. 1993
11. J.F.Steffensen,”On the problem of three bodies in the planet”,Mat.-Fys. Medd. Danske, Videnskab. Selskab , 31, 3, 1957.
12. D. J. Bates, J. D. Hauenstein, A. J. Sommese, C. W. Wampler, ”Nu- merically solving polynomial systems with Bertini”, SIAM, Philadelphia, Software, Environments, and Tools, Vol. 25, 2015.
13. L. K. Babadzanjanz, ”Existence of the Continuations and Representation of the Solutions in Celestial Mechanics”, TRUDY ITA, vyp. XVII , 3-45, 1978 (in Russian).
14. D. C. Carothers, E. G. Parker, J. S. Sochacki, P. G. Warne, ”Some properties of solutions to polynomial systems of differential equations”, Electronic Journal of Differential Equations, 40, Vol. 2005, 117, 2005.
15. Бабаджанянц Л. К., Большаков А. И. Реализация метода рядов Тейлора для решения обыкновенных дифференциальных уравнений // Вычислительные методы и программирование. Научно-исследовательский вычислительный центр МГУ им. М.В. Ломоносова. 2012. Т. 13. C. 497–510.
16. TIDES webpage URL: gme.unizar.es/software/tides
17. C. Oesterwinter, C. J. Cohen, ”New orbital elements for Moon and planets”, Celestial Mech., 5, No. 3, 317395, 1972.
18. NASA Jet Propulsion Laboratory URL: ssd.jpl.nasa.gov/?constants.

**Приложения I, II, III**

**Приложение I: Программа построения минимальной оболочки и схемы**

Программа №1: ***Mono3***

Код в Wolfram Mathematica, выдаёт минимальную оболочку и схему для произвольного набора мономов третьей степени.

Входные данные. Например:   
m = DeleteDuplicates [ { x[1] x[2] , x[1] x[4] x[5], x[1] x[2] x[3] }]

Выходные данные. Минимальная оболочка и схема:  
{x[1]x[2], x[1]x[4], x[1]x[4]x[5], x[1]x[2]x[3]};   
{{1,2}, {1,4}, {5,7}, {3,6}}

(\* Чтобы использовать предложенные программы, скопируйте ниже код и вставьте в Wolfram Mathematica \*)

im[t\_] := Times @@ (x[#1] & ) /@ t;  
mi[t\_] := Flatten[t /. {Power -> ConstantArray, Times -> List} /.  
x -> Identity];  
m = DeleteDuplicates[{x[1]\*x[2], x[1]\*x[4]\*x[3]}];  
monomial1 = im /@ Transpose[{Sort[DeleteDuplicates[Catenate[ mi /@ m]]]}];  
monomial2 = Select[m, Length[mi[#1]] == 2 & ];  
monomial3 = Select[m, Length[mi[#1]] == 3 & ];  
m3 = DeleteCases[Complement[m, monomial2],

t\_ /; Length[Intersection[DeleteDuplicates[Subsets[mi[t], {2}]], mi /@ monomial2]] > 0];  
m3Subsets = (DeleteDuplicates[Subsets[mi[#1], {2}]] & ) /@ m3; m3AllSubsets = DeleteDuplicates[Catenate[m3Subsets]]; m3OptimalSubsets = If[m3AllSubsets == {}, {},

im /@ m3AllSubsets[[Transpose[Position[LinearProgramming[ Table[1., {i, Length[m3AllSubsets]}],

(ReplacePart[Table[0, {i, Length[m3AllSubsets]}], Transpose[{#1}] -> 1] & ) /@ Table[FirstPosition[m3AllSubsets, m3Subsets[[r,j]]][[1]],

{r, Length[m3]},  
{j, Length[m3Subsets[[r]]]}], Table[1, {i, Length[m3]}],

Table[{0, 1}, {i, Length[m3AllSubsets]}], Integers], 1]][[1]]]]]; monomial2 = Join[monomial2, m3OptimalSubsets]; allmonomial3degree = Join[monomial1, monomial2, monomial3] monomial1forscheme = ({#1} & ) /@ Flatten[mi /@ {monomial1}]; monomial2forscheme = Sort[mi /@ monomial2]; monomial3forscheme = Sort[mi /@ monomial3]; allmonomialscheme = Join[monomial2sсheme =

(Catch[Do[If[#1 == Join[monomial1forscheme[[i]], monomi- al1forscheme[[j]]],

Throw[{i, j}]], {i, Length[monomial1forscheme]},

{j, Length[monomial1forscheme]}]] & ) /@ mi /@ monomi- al2forscheme, monomial3sсheme =

(Catch[Do[If[#1 == Sort[Join[monomial1forscheme[[i]], monomi- al2forscheme[[j]]]],  
Throw[{i, Length[monomial1forscheme] + j}]],  
{i, Length[monomial1forscheme]},

{j, Length[monomial2forscheme]}]] & ) /@ monomial3forscheme]

**Приложение II: Задача N тел. Время вычисления правых частей системы**

**А. Время вычисления правых частей системы пятой степени**

Программа №2: ***Nbody5***

Входные данные. Количество планет и количество добавленных мономов:

Например : N= 3; add = 3;

Выходные данные.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 3 | 15 | 24 | 27 | 1.2 |

im[t\_] := Times @@ (x[#1] & ) /@ t;  
mi[t\_] := Flatten[{t /. {Power -> ConstantArray, Times -> List} /. x -> Identity}];  
bodiesN = 3; L = bodiesN - 1; add = 3;  
monomial5degree = DeleteDuplicates[Flatten[{Table[  
Subscript[g, i, j]\*Subscript[d, 0, i]^3, {i, L}, {j, 3}],

Table[If[s == i, {}, {Subscript[g, i, j]\*If[s > i, Subscript[d, i, s]^3, Subscript[d, s, i]^3]}], {i, L}, {j, 3}, {s, L}],

Table[If[s == i, {}, {Subscript[g, i, j]\*If[s > i, Subscript[d, i, s]^3, Subscript[d, s, i]^3]\*Subscript[p, i, j]}], {i, L}, {j, 3},

{s, L}], Table[If[s == i, {}, {Subscript[g, i, j]\*If[s > i, Subscript[d, i, s]^3, Subscript[d, s, i]^3]}\*Subscript[p, s, j]], {i, L},

{j, 3}, {s, L}]}]];  
monomial5degreeAdd = DeleteDuplicates[Flatten[{Table[If[s < i, Subscript[d, s, i]^2, {}], {i, L}, {s, L}],

Table[Subscript[d, 0, i]^2, {i, L}], Table[If[s < i, Subscript[d, s, i]^3, {}], {i, L}, {s, L}], Table[Subscript[d, 0, i]^3, {i, L}]}]]; allmonomial5degree = Join[monomial5degree, monomi- al5degreeAdd];

monomial5degreeNewVariable = SortBy[monomial5degree /. MapIn- dexed[#1 -> x[#2[[1]]] & ,

DeleteDuplicates[Flatten[monomial5degree /. {Power -> Con- stantArray, Times -> List}]]], mi[#1] & ];  
allmonomial5degree = SortBy[allmonomial5degree /. MapIndexed[#1 -> x[#2[[1]]] & ,

DeleteDuplicates[Flatten[allmonomial5degree /. {Power -> ConstantArray, Times -> List}]]], mi[#1] & ];  
monomial5degree1 = im /@ Trans- pose[{Sort[DeleteDuplicates[Catenate[mi /@ allmonomi- al5degree]]]}];

rules = Apply[Rule, Transpose[{monomial5degree1, RandomReal[{-1, 1}, Length[monomial5degree1], WorkingPrecision -> 16]}], {1}]; monomial5degree2 = Select[allmonomial5degree, Length[mi[#1]] == 2 & ];

monomial5degree3 = Select[allmonomial5degree, Length[mi[#1]] == 3 & ];  
m = im /@ RandomInteger[{1, Length[monomial5degree1]}, {add, 3}]; m3 = DeleteCases[Complement[m, monomial5degree2], t\_ /; Length[Intersection[DeleteDuplicates[Subsets[mi[t], {2}]], mi /@ monomial5degree2]] > 0];  
m3Subsets = (DeleteDuplicates[Subsets[mi[#1], {2}]] & ) /@ m3; m3AllSubsets = DeleteDuplicates[Catenate[m3Subsets]]; m3OptimalSubsets = If[m3AllSubsets == {}, {},

im /@ m3AllSubsets[[Transpose[Position[LinearProgramming[Table[1., {i, Length[m3AllSubsets]}],

(ReplacePart[Table[0, {i, Length[m3AllSubsets]}], Trans- pose[{#1}] -> 1] & ) /@ Table[FirstPosition[m3AllSubsets, m3Subsets[[r,j]]][[

1]], {r, Length[m3]}, {j, Length[m3Subsets[[r]]]}], Ta- ble[1, {i, Length[m3]}], Table[{0, 1}, {i, Length[m3AllSubsets]}],

Integers], 1]][[1]]]]];  
monomial5degree2 = Join[monomial5degree2, m3OptimalSubsets]; monomial5degree3 = Join[monomial5degree3, m]; monomial5degree4 = Select[allmonomial5degree, Length[mi[#1]] == 4 & ];  
monomial5degree5 = Select[allmonomial5degree, Length[mi[#1]] == 5 & ];  
monomial5degree1ForScheme = ({#1} & ) /@ Flatten[mi /@ {mono- mial5degree1}];  
monomial5degree2ForScheme = Sort[mi /@ monomial5degree2]; monomial5degree3ForScheme = Sort[mi /@ monomial5degree3]; monomial5degree4ForScheme = Sort[mi /@ monomial5degree4]; monomial5degree5ForScheme = Sort[mi /@ monomial5degree5]; monomial5sсheme = {(Catch[Do[If[#1 == Sort[Join[monomial5degree1ForScheme[[i]], monomi- al5degree2ForScheme[[j]]]],

Throw[{i, Length[monomial5degree1ForScheme] + j}]], {i, Length[monomial5degree1ForScheme]},

{j, Length[monomial5degree2ForScheme]}]] & ) /@ monomi- al5degree3ForScheme,

(Catch[Do[If[#1 == Sort[Join[monomial5degree2ForScheme[[i]], monomial5degree2ForScheme[[j]]]],

Throw[Length[monomial5degree1ForScheme] + {i, j}]], {i, Length[monomial5degree2ForScheme]}, {j, Length[monomial5degree2ForScheme]}];

Do[If[#1 == Sort[Join[monomial5degree1ForScheme[[i]], monomial5degree3ForScheme[[j]]]],

Throw[{i, Length[monomial5degree1ForScheme] + Length[monomial5degree2ForScheme] + j}]], {i, Length[monomial5degree1ForScheme]},

{j, Length[monomial5degree3ForScheme]}]; ] & ) /@ mono- mial5degree4ForScheme,

(Catch[Do[If[#1 == Sort[Join[monomial5degree1ForScheme[[i]], monomial5degree4ForScheme[[j]]]],  
Throw[{i, Length[monomial5degree1ForScheme] + Length[monomial5degree2ForScheme] + Length[monomial5degree3ForScheme] + j}]],

{i, Length[monomial5degree1ForScheme]}, {j, Length[monomial5degree4ForScheme]}];  
Do[If[#1 == Sort[Join[monomial5degree2ForScheme[[i]],

monomial5degree3ForScheme[[j]]]], Throw[Length[monomial5degree1ForScheme] + {i, Length[monomial5degree2ForScheme] + j}]], {i, Length[monomial5degree2ForScheme]},  
{j, Length[monomial5degree3ForScheme]}]; ] & )

/@ monomial5degree5ForScheme}; n = 100;  
result = {(1/n)\*AbsoluteTiming[Do[monomial5degreeNewVariable /. rules, {n}]][[1]],

(1/n)\*AbsoluteTiming[Do[a = Join[monomial5degree1, monomi- al5degree2] /. rules; Do[a = Join[a, (a[[#1[[1]]]]\*a[[#1[[2]]]] & ) /@ i],

{i, monomial5sсheme}]; , {n}]][[1]]}; result[[1]] Grid[{{"N", "\!\(\\*SubscriptBox[\(N\), \(v\)]\)", "\!\(\\*SubscriptBox[\(N\), \(m\)]\)", "\!\(\\*SubsuperscriptBox[\(N\), \(m\), \(+\)]\)",

If[add == 0, "t/\!\(\\*SubscriptBox[\(t\), \(s\)]\)", "\!\(\\*SuperscriptBox[\(t\),

\(+\)]\)/\!\(\\*SubsuperscriptBox[\(t\), \(s\), \(+\)]\)"]}, {bodiesN, Length[monomial5degree1], Length[monomial5degree],

Length[monomial5degree] + add, result[[1]]/result[[2]]}}, Frame -> All, Spacings -> {2, 2}]

**В. Время вычисления правых частей системы четвертой степени**

Программа №3: ***Nbody4***

Входные данные. Количество планет и количество добавленных мономов:  
Например : N= 3; add = 3;

Выходные данные.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 3 | 21 | 50 | 53 | 1.1 |

im[t\_] := Times @@ (x[#1] & ) /@ t;  
mi[t\_] := Flatten[{t /. {Power -> ConstantArray, Times -> List} /. x -> Identity}];  
bodiesN = 3; L = bodiesN - 1; add = 3;  
monomial4degree = DeleteDuplicates[Flatten[{Table[  
Subscript[g, i, j]\*Subscript[v, 0, i], {j, 3}, {i, L}],

Table[Subscript[g, i, j]\*If[s == i, {}, If[s > i, Subscript[v, i, s], Subscript[v, s, i]]], {j, 3}, {i, L}, {s, L}],

Table[Subscript[g, s, j]\*Subscript[v, 0, s], {j, 3}, {s, L}], Table[Subscript[v, 0, i]\*Subscript[w, 0, i], {i, L}],

Table[If[s == i, {}, If[s > i, Subscript[v, i, s]\*Subscript[w, i, s], Subscript[v, s, i]\*Subscript[w, s, i]]], {s, L}, {i, L}],

Table[{Subscript[p, i, j]^2, Subscript[p, i, j]\*Subscript[p, s, j], Subscript[p, s, j]^2}, {j, 3}, {i, L}, {s, L}],

Table[{Table[{Subscript[d, s, i]^2\*Subscript[v, s, i]\* Subscript[w, s, i], Table[{Subscript[g, i, j]^2\*Subscript[v, 0, i],

Subscript[g, i, j]\*Subscript[g, s, j]\*Subscript[v, 0, i], Subscript[g, i, j]\*Subscript[g, s, j]\*Subscript[v, 0, s],

Subscript[g, s, j]^2\*Subscript[v, 0, s], Subscript[g, i, j]\* Table[If[s == i, {}, {Subscript[g, s, j]\*If[s > i, Subscript[v, i, s],

Subscript[v, s, i]], Subscript[g, i, j]\*If[s > i, Subscript[v, i, s], Sub- script[v, s, i]], Subscript[g, s, j]\*

Subscript[v, 0, s]}], {s, L}], Subscript[g, s, j]\*Table[If[s == i, {}, {Subscript[g, s, j]\*If[s > i, Subscript[v, i, s],

Subscript[v, s, i]], Subscript[g, i, j]\*If[s > i, Subscript[v, i, s], Sub- script[v, s, i]], Subscript[g, s, j]\*

Subscript[v, 0, s]}], {s, L}]}, {j, 3}]}, {s, i - 1}], {Subscript[d, 0, i]\*Subscript[v, 0, i]\*Subscript[w, 0, i],

Subscript[d, 0, i]^2\*Subscript[v, 0, i]\*Subscript[w, 0, i], Ta- ble[{Subscript[g, i, j]\*Subscript[g, i, j]\*Subscript[v, 0, i],

Subscript[g, i, j]\*Table[If[s == i, {}, {Subscript[g, s, j]\*Subscript[v, 0, s], Subscript[g, s, j]\*If[s > i, Subscript[v, i, s],  
Subscript[v, s, i]], Subscript[g, i, j]\*If[s > i, Subscript[v, i, s], Sub- script[v, s, i]]}], {s, L}]}, {j, 3}]}}, {i, L}]}]];

monomial4degreeAdd = DeleteDuplicates[Flatten[{Table[If[s < i, Subscript[d, s, i]^2, {}], {i, L}, {s, L}], Table[Subscript[d, 0, i]^2, {i, L}]}]];  
data = Join[{{0.8588336752475201, -0.4935399919452301, - 0.2139978024915201, 0.874428008809101, 1.377761306977401, 0.602870470591201}, {0.3014126682538901, 0.1285949471782201, 0.03784152564682, -1.681668078098101, 2.133460037844401, 1.422459237170901}, {0.5359773796741901, 0.4544720093036001, 0.1708366717263001, -1.363414227994001, 1.328526692131601, 0.684597990991701}, {0.8561698332307001, -0.4940630457237901, -0.2143090507623301, 0.887983439180701, 1.330002537802801, 0.5768781098964001},

{1.205405982723601, 0.7272027017407001, 0.3012692776955001, -0.7092835268749001, 1.1656043764443, 0.5541349661214001},

{1.5413706201915, -4.5249958636625, -1.9791722571698, 0.7124662329600001, 0.2456541072521801, 0.0879628631403001},

{2.9214711965716003, 7.9536057652255, 3.1615878484025, - 0.5578190953901901, 0.1571326847358001, 0.0890456574959601},

{11.7455704480840001`17.06987411404159, -14.5716641002640001`17.163509151515584, -6.55117579443,

0.3142105227680801, 0.1985913565926601, 0.0825679767743901}, {-13.3527443024030001`17.125570532585, 24.7256358237900001`17.393147468366813,10.464581607667000 1`17.01972186916645, -0.2823271047024101, -0.1298339633633901, -0.0460698226012701}, {-0.1968083804770001, 42.6773912708560001`17.630197864564565, 13.5391651708380001`17.131591886441, -0.2240184739979901,- 0.0686097157222501, 0.0464070770042101}}, If[L < 9, {}, Random- Real[{-1, 1}, {L - 9, 6}, WorkingPrecision -> 16]]][[1 ;; L + 1]];

rules = Flatten[Table[{Subscript[g, i, j] -> data[[i + 1,j]] - data[[1,j]], Subscript[p, i, j] -> data[[i + 1,3 + j]] - data[[1,3 + j]]}, {i, L}, {j, 3}]]; rules = Join[rules, Flatten[Join[Table[{Subscript[d, s, i] -> Sum[(Subscript[g, s, j] - Subscript[g, i, j])^2, {j, 3}]^(-(1/2)), Subscript[w, s, i] -> Sum[(Subscript[g, i, j] - Subscript[g, s, j])\*(Subscript[p, i, j] - Subscript[p, s, j]), {j, 3}]}, {i, 2, L},

{s, i - 1}], Table[{Subscript[d, 0, i] -> Sum[Subscript[g, i, j]^2, {j, 3}]^(-(1/2)), Subscript[w, 0, i] ->  
Sum[Subscript[g, i, j]\*Subscript[p, i, j], {j, 3}]}, {i, L}]]] /. rules];  
rules = Join[rules, Flatten[Table[{Subscript[v, s, i] -> Subscript[d, s, i]^3}, {i, L}, {s, 0, i - 1}]] /. rules];

allmonomial4degree = Join[monomial4degree, monomi- al4degreeAdd];  
map = MapIndexed[#1 -> x[#2[[1]]] & , DeleteDupli- cates[Flatten[allmonomial4degree /. {Power -> ConstantArray, Times -> List}]]];

allmonomial4degree = DeleteDupli- cates[Join[SortBy[allmonomial4degree /. map, mi[#1] & ]]]; monomial4degreeNewVariable = DeleteDupli- cates[Join[SortBy[monomial4degree /. map, mi[#1] & ]]];  
rules = rules /. map;  
monomial4degree1 = im /@ Transpose[{Sort[DeleteDuplicates[Catenate[mi  
/@ allmonomial4degree]]]}];  
monomial4degree2 = Select[allmonomial4degree,  
Length[mi[#1]] == 2 & ];  
monomial4degree3 = Select[allmonomial4degree,  
Length[mi[#1]] == 3 & ];  
monomial4degree4 = Select[allmonomial4degree,  
Length[mi[#1]] == 4 & ];  
m = im /@ RandomInteger[{1, Length[monomial4degree1]}, {add, 3}];

m3 = DeleteCases[Complement[m, monomial4degree2], t\_ /; Length[Intersection[DeleteDuplicates[Subsets[mi[t], {2}]], mi /@ monomial4degree2]] > 0];  
m3Subsets = (DeleteDuplicates[Subsets[mi[#1], {2}]] & ) /@ m3; m3AllSubsets = DeleteDuplicates[Catenate[m3Subsets]]; m3OptimalSubsets = If[m3AllSubsets == {}, {},

im /@m3AllSubsets[[Transpose[Position[ LinearProgramming[Table[1., {i, Length[m3AllSubsets]}], (ReplacePart[Table[0, {i, Length[m3AllSubsets]}],

Transpose[{#1}] -> 1] & ) /@ Table[FirstPosition[m3AllSubsets, m3Subsets[[r,j]]][[1]], {r, Length[m3]}, {j, Length[m3Subsets[[r]]]}], Table[1, {i, Length[m3]}], Table[{0, 1}, {i, Length[m3AllSubsets]}],

Integers], 1]][[1]]]]];  
monomial4degree2 = Join[monomial4degree2, m3OptimalSubsets]; monomial4degree3 = Join[monomial4degree3, m]; monomial4degree4 = Select[allmonomial4degree,  
Length[mi[#1]] == 4 & ];  
monomial4degree1ForScheme = ({#1} & ) /@ Flatten[mi /@ {mono- mial4degree1}];  
monomial4degree2ForScheme = Sort[mi /@ monomial4degree2]; monomial4degree3ForScheme = Sort[mi /@ monomial4degree3]; monomial4degree4ForScheme = Sort[mi /@ monomial4degree4]; monomial4sсheme = {(Catch[Do[If[#1 == Sort[Join[monomial4degree1ForScheme[[i]], monomi- al4degree2ForScheme[[j]]]],

Throw[{i, Length[monomial4degree1ForScheme] + j}]], {i, Length[monomial4degree1ForScheme]},

{j, Length[monomial4degree2ForScheme]}]] & ) /@ monomi- al4degree3ForScheme,

(Catch[Do[If[#1 == Sort[Join[monomial4degree2ForScheme[[i]], monomial4degree2ForScheme[[j]]]],

Throw[Length[monomial4degree1ForScheme] + {i, j}]], {i, Length[monomial4degree2ForScheme]}, {j, Length[monomial4degree2ForScheme]}];

Do[If[#1 == Sort[Join[monomial4degree1ForScheme[[i]], monomial4degree3ForScheme[[j]]]],

Throw[{i, Length[monomial4degree1ForScheme] + Length[monomial4degree2ForScheme] + j}]], {i, Length[monomial4degree1ForScheme]},

{j, Length[monomial4degree3ForScheme]}]; ] & ) /@ mono- mial4degree4ForScheme};

n = 1000; result = {(1/n)\*AbsoluteTiming[Do[Join[monomial4degreeNewVariable, m] /. rules, {n}]][[1]],

(1/n)\*AbsoluteTiming[Do[a = Join[monomial4degree1, monomi- al4degree2] /. rules; Do[a = Join[a, (a[[#1[[1]]]]\*a[[#1[[2]]]] & ) /@ i],

{i, monomial4sсheme}]; , {n}]][[1]]}; result[[1]] Grid[{{"N", "\!\(\\*SubscriptBox[\(N\), \(v\)]\)", "\!\(\\*SubscriptBox[\(N\), \(m\)]\)", "\!\(\\*SubsuperscriptBox[\(N\), \(m\), \(+\)]\)",

If[add == 0, "t/\!\(\\*SubscriptBox[\(t\), \(s\)]\)", "\!\(\\*SuperscriptBox[\(t\),

\(+\)]\)/\!\(\\*SubsuperscriptBox[\(t\), \(s\), \(+\)]\)"]}, {bodiesN, Length[monomial4degree1], Length[monomial4degree],

Length[monomial4degree] + add, result[[1]]/result[[2]]}}, Frame -> All, Spacings -> {2, 2}]

**C. Время вычисления правых частей системы третьей степени**

Программа №4: ***Nbody3***

Входные данные. Количество планет и количество добавленных мономов.  
Например : N= 3; add = 3;

Выходные данные.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 3 | 24 | 51 | 54 | 1.3 |

im[t\_] := Times @@ (x[#1] & ) /@ t;  
mi[t\_] := Flatten[{t /. {Power -> ConstantArray, Times -> List} /. x -> Identity}];  
bodiesN = 3; L = bodiesN - 1; add = 3;  
{m2 = DeleteDuplicates[Flatten[Table[{Table[{Subscript[g, i, j]\* Subscript[v, 0, i], Table[If[s == i, {}, {Subscript[g, s, j]\*If[s > i,

Subscript[v, i, s], Subscript[v, s, i]], Subscript[g, i, j]\*If[s > i, Subscript[v, i, s], Subscript[v, s, i]], Subscript[g, s, j]\*Subscript[v, 0, s]}], {s, L}]}, {j, 3}], Table[{Subscript[v, s, i]\*Subscript[w, s, i], Table[{Subscript[p, i, j]^2, Subscript[p, i, j]\*Subscript[p, s, j], Subscript[p, s, j]^2}, {j, 3}]}, {s, i - 1}], {Subscript[v, 0, i]\*  
Subscript[w, 0, i], Table[Subscript[p, i, j]^2, {j, 3}]}},  
{i, L}]]], m3 = DeleteDuplicates[Flatten[Table[{Table[  
{Subscript[d, s, i]\*Subscript[v, s, i]\*Subscript[w, s, i],

Subscript[q, s, i]\*Subscript[v, s, i]\*Subscript[w, s, i], Table[{Subscript[g, i, j]^2\*Subscript[v, 0, i],  
Subscript[g, i, j]\*Subscript[g, s, j]\*Subscript[v, 0, i], Subscript[g, i, j]\* Subscript[g, s, j]\*Subscript[v, 0, s],  
Subscript[g, s, j]^2\*Subscript[v, 0, s], Subscript[g, i, j]\*Table[  
If[s == i, {}, {Subscript[g, s, j]\*If[s > i, Subscript[v, i, s],  
Subscript[v, s, i]], Subscript[g, i, j]\*If[s > i, Subscript[v, i, s], Subscript[v, s, i]], Subscript[g, s, j]\*Subscript[v, 0, s]}], {s, L}], Subscript[g, s, j]\*Table[If[s == i, {}, {Subscript[g, s, j]\*  
If[s > i, Subscript[v, i, s], Subscript[v, s, i]], Subscript[g, i, j]\*If[s > i, Subscript[v, i, s], Subscript[v, s, i]], Subscript[g, s, j]\*Subscript[v, 0, s]}], {s, L}]}, {j, 3}]}, {s, i - 1}], {Subscript[d, 0, i]\*  
Subscript[v, 0, i]\*Subscript[w, 0, i], Subscript[q, 0, i]\*  
Subscript[v, 0, i]\*Subscript[w, 0, i], Table[  
{Subscript[g, i, j]\*Subscript[g, i, j]\*Subscript[v, 0, i], Subscript[g, i, j]\* Table[If[s == i, {}, {Subscript[g, s, j]\*Subscript[v, 0, s],  
Subscript[g, s, j]\*If[s > i, Subscript[v, i, s], Subscript[v, s, i]], Subscript[g, i, j]\*If[s > i, Subscript[v, i, s], Subscript[v, s, i]]}], {s, L}]}, {j, 3}]}}, {i, L}]]]}; monomial = Join[m2, m3];  
monomial1 = im /@ Transpose[{Sort[DeleteDuplicates[Catenate[  
mi /@ monomial]]]}];  
data = Join[{{0.8588336752475201, -0.4935399919452301, -0.2139978024915201, 0.874428008809101, 1.377761306977401, 0.602870470591201}, {0.3014126682538901, 0.1285949471782201, 0.03784152564682, -1.681668078098101, 2.133460037844401, 1.422459237170901}, {0.5359773796741901, 0.4544720093036001, 0.1708366717263001, -1.363414227994001, 1.328526692131601, 0.684597990991701}, {0.8561698332307001, -0.4940630457237901, -0.2143090507623301, 0.887983439180701, 1.330002537802801, 0.5768781098964001},  
{1.205405982723601, 0.7272027017407001, 0.3012692776955001, -0.7092835268749001, 1.1656043764443, 0.5541349661214001}, {1.5413706201915, -4.5249958636625, -1.9791722571698, 0.7124662329600001, 0.2456541072521801, 0.0879628631403001},

{2.9214711965716003, 7.9536057652255, 3.1615878484025, -0.5578190953901901, 0.1571326847358001, 0.0890456574959601},{11.7455704480840001`17.06987411404159, -14.5716641002640001`17.163509151515584, -6.55117579443, 0.3142105227680801, 0.1985913565926601, 0.0825679767743901}, {-13.3527443024030001`17.125570532585, 24.7256358237900001`17.393147468366813, 10.4645816076670001`17.01972186916645, -0.2823271047024101, -0.1298339633633901, -0.0460698226012701}, {-0.1968083804770001, 42.6773912708560001`17.630197864564565, 13.5391651708380001`17.131591886441, -0.2240184739979901,

-0.0686097157222501, 0.0464070770042101}},  
If[L < 9, {}, RandomReal[{-1, 1}, {L - 9, 6}, WorkingPrecision -> 16]]] [[1 ;; L + 1]];  
rules = Flatten[Table[{Subscript[g, i, j] -> data[[i + 1,j]] - data[[1,j]], Subscript[p, i, j] -> data[[i + 1,3 + j]] - data[[1,3 + j]]}, {i, L}, {j, 3}]]; rules = Join[rules, Flatten[Join[Table[{Subscript[d, s, i] -> Sum[(Subscript[g, s, j] - Subscript[g, i, j])^2, {j, 3}]^(-(1/2)), Subscript[w, s, i] -> Sum[(Subscript[g, i, j] - Subscript[g, s, j])\* (Sub- script[p, i, j] - Subscript[p, s, j]), {j, 3}]}, {i, 2, L},  
{s, i - 1}], Table[{Subscript[d, 0, i] -> Sum[Subscript[g, i, j]^2,  
{j, 3}]^(-(1/2)), Subscript[w, 0, i] -> Sum[Subscript[g, i, j]\* Subscript[p, i, j], {j, 3}]}, {i, L}]]] /. rules];  
rules = Join[rules, Flatten[Table[{Subscript[q, s, i] -> Subscript[d, s, i]^2, Subscript[v, s, i] -> Subscript[d, s, i]^3}, {i, L}, {s, 0, i - 1}]]

/. rules]; map = MapIndexed[#1 -> x[#2[[1]]] & , DeleteDuplicates[Flatten[monomial /. {Power -> ConstantArray, Times -> List}]]]; monomial = DeleteDuplicates[ Join[SortBy[monomial /. map, mi[#1] & ]]]; rules = rules /. map; m = im /@ RandomInteger[{1, Length[monomial1]}, {add, 3}]; monomial3degree2 = Select[monomial, Length[mi[#1]] == 2 & ]; monomial3degree3 = Select[monomial, Length[mi[#1]] == 3 & ]; m3 = DeleteCases[Complement[m, monomial3degree2], t\_ /; Length[Intersection[DeleteDuplicates[Subsets[mi[t], {2}]], mi /@ monomial3degree2]] > 0];

m3Subsets = (DeleteDuplicates[Subsets[mi[#1], {2}]] & ) /@ m3; m3AllSubsets = DeleteDuplicates[Catenate[m3Subsets]]; m3OptimalSubsets = If[m3AllSubsets == {}, {},  
im /@ m3AllSubsets[[Transpose[Position[LinearProgramming[ Table[1., {i, Length[m3AllSubsets]}], (ReplacePart[Table[ 0,

{i, Length[m3AllSubsets]}], Transpose[{#1}] -> 1] & )  
/@ Table[FirstPosition[m3AllSubsets, m3Subsets[[r,j]]][[1]],  
{r, Length[m3]}, {j, Length[m3Subsets[[r]]]}], Table[1, {i, Length[m3]}], Table[{0, 1}, {i, Length[m3AllSubsets]}], Integers], 1]] [[1]]]]];  
monomial23OptimalSubsets = Join[monomial3degree2, m3OptimalSubsets];  
monomial3degree3 = Join[monomial3degree3, m];  
monomial1 = im /@ Transpose[{Sort[DeleteDuplicates[Catenate[ mi /@ monomial]]]}];  
monomial3degree1ForScheme = ({#1} & ) /@ Flatten[mi  
/@ {monomial1}];  
monomial3degree2ForScheme = Sort[mi /@ Select[monomial, Length[mi[#1]] == 2 & ]];  
monomial3degree3ForScheme = Sort[mi /@ Select[monomial, Length[mi[#1]] == 3 & ]];  
monomial3scheme = (Catch[Do[If[#1 == Sort[Join[monomial3degree1ForScheme[[i]], monomi- al3degree2ForScheme[[j]]]],  
Throw[{i, Length[monomial3degree1ForScheme] + j}]], {i, Length[monomial3degree1ForScheme]},  
{j, Length[monomial3degree2ForScheme]}]] & ) /@ monomi- al3degree3ForScheme;  
result = Mean[Table[ClearSystemCache[]; {AbsoluteTiming[ monomial /. rules][[1]],  
AbsoluteTiming[With[{monomial2 = Join[monomial1 /. rules, monomial23OptimalSubsets /. rules]}, (monomial2[[#1[[1]]]]\*monomial2[[#1[[2]]]] & )  
/@ monomial3scheme]][[1]]}, {100}]]; result[[1]]  
Grid[{{"N", "\!\(\\*SubscriptBox[\(N\), \(v\)]\)", "\!\(\\*SubscriptBox[\(N\), \(m\)]\)", "\!\(\\*SubsuperscriptBox[\(N\), \(m\), \(+\)]\)",

If[add == 0, "t/\!\(\\*SubscriptBox[\(t\), \(s\)]\)", "\!\(\\*SuperscriptBox[\(t\),

\(+\)]\)/\!\(\\*SubsuperscriptBox[\(t\), \(s\), \(+\)]\)"]}, {bodiesN, Length[monomial1], Length[monomial], Length[monomial] + add, result[[1]]/result[[2]]}}, Frame -> All, Spacings -> {2, 2}]

**D. Время вычисления правых частей исходной системы**

Программа №5: ***RHS***

Входные данные. Количество планет и количество добавленных мономов:  
Например: N= 3;  
Выходные данные. t = 0.00117

im[t\_] := Times @@ (x[#1] & ) /@ t;  
mi[t\_] := Flatten[{t /. {Power -> ConstantArray, Times -> List} /. x -> Identity}];  
bodiesN = 3;  
L = bodiesN - 1;  
k = 0.1311567724137797;  
mass = Join[{3.694329711493759/10^8, 1.6601367952719304/10^7, 2.4478395979668246/10^6, 3.00348971264827/10^6, 3.227149362153928/10^7, 0.0009547906621473, 0.0000285877644368, 0.0000435540069686, 0.0000517759138448, 7.692307692307693/10^9},

If[L < 9, {}, Flatten[RandomReal[{-1, 1}, L - 9,  
WorkingPrecision -> 16], 2]]];  
mData = Table[Subscript[m, i] -> mass[[i + 1]], {i, 0, L}];  
system = Join[Flatten[Table[k^2\*(((-(Subscript[m, 0] +  
Subscript[m, i]))\*Subscript[g, i, j])/Sum[(Subscript[g, 0, j] – Subscript[g, i, j])^2,  
{j, 3}]^(3/2) + Sum[If[s == i, 0, Subscript[m, s]\*((Subscript[g, s, j] - Subscript[g, i, j])\*If[s > i, Sum[(Subscript[g, i, j] - Subscript[g, s, j])^2, {j, 3}]^(-(3/2)), Sum[(Subscript[g, s, j] - Subscript[g, i, j])^2, {j, 3}]^(- (3/2))] - Subscript[g, s, j]/Sum[(Subscript[g, 0, j] - Subscript[g, i, j])^2, {j, 3}]^(3/2))], {s, L}]), {i, L}, {j, 3}, {s, 0, i - 1}], 2]];  
data = Join[{{0.8588336752475201, -0.4935399919452301, - 0.2139978024915201, 0.874428008809101, 1.377761306977401,

0.602870470591201}, {0.3014126682538901, 0.1285949471782201, 0.03784152564682, -1.681668078098101, 2.133460037844401, 1.422459237170901}, {0.5359773796741901, 0.4544720093036001, 0.1708366717263001, -1.363414227994001, 1.328526692131601, 0.684597990991701}, {0.8561698332307001, -0.4940630457237901, -0.2143090507623301, 0.887983439180701, 1.330002537802801, 0.5768781098964001},

{1.205405982723601, 0.7272027017407001, 0.3012692776955001, -0.7092835268749001, 1.1656043764443, 0.5541349661214001}, {1.5413706201915, -4.5249958636625, -1.9791722571698, 0.7124662329600001, 0.2456541072521801, 0.0879628631403001}, {2.9214711965716003, 7.9536057652255, 3.1615878484025, -0.5578190953901901, 0.1571326847358001, 0.0890456574959601},{11.7455704480840001`17.06987411404159, -14.5716641002640001`17.163509151515584, -6.55117579443, 0.3142105227680801, 0.1985913565926601, 0.0825679767743901}, {-13.3527443024030001`17.125570532585, 24.7256358237900001`17.393147468366813,

10.4645816076670001`17.01972186916645, -0.2823271047024101, -0.1298339633633901, -0.0460698226012701}, {-0.1968083804770001, 42.6773912708560001`17.630197864564565, 13.5391651708380001`17.131591886441, -0.2240184739979901, -0.0686097157222501, 0.0464070770042101}},

If[L < 9, {}, RandomReal[{-1, 1}, {L - 9, 6}, WorkingPrecision -> 16]]] [[1 ;; L + 1]];  
rules = Join[Flatten[Table[  
{Subscript[g, i, j] -> data[[i + 1,j]] –data[[1,j]],

Subscript[p, i, j] -> data[[i + 1,3 + j]] - data[[1,3 + j]]}, {i, L}, {j, 3}]], Flatten[Table[{Subscript[g, 0, j] -> 1, Subscript[p, 0, j] -> 1}, {j, 3}]]]; n=100;  
(1/n)\*AbsoluteTiming[Do[system /. rules /. mData, {n}]][[1]]

**Приложение III. Программа оценки эффективности схем**

Программа №6: ***RandomM***

Входные данные. Количество переменных и мономов.  
Например: variable = 200; amount3degree = 1 000;

im[t\_] := Times @@ (x[#1] & ) /@ t;  
mi[t\_] := Flatten[t /. {Power -> ConstantArray, Times -> List}  
/. x -> Identity];  
all = Table[variable = 50; amount2degree = 0; amount3degree = 500; m = Join[Take[DeleteDuplicates[im /@ RandomInteger[{1, variable}, {amount2degree\*5, 2}]], amount2degree], Take[DeleteDuplicates[  
im /@ RandomInteger[{1, variable}, {amount3degree\*5, 3}]], amount3degree]]; monomial1 =

im /@ Transpose[{Sort[DeleteDuplicates[Catenate[  
mi /@ m]]]}]; rules = Apply[Rule, Transpose[{monomial1, RandomReal[{-1, 1}, Length[monomial1], WorkingPrecision -> 16]}], {1}]; monomial2 = Select[m, Length[mi[#1]] == 2 & ];  
monomial3 = Select[m, Length[mi[#1]] == 3 & ];  
m3 = DeleteCases[Complement[m, monomial2], t\_ /; Length[Intersection[DeleteDuplicates[Subsets[mi[t], {2}]], mi /@ monomial2]] > 0]; m3Subsets = (DeleteDuplicates[Subsets[  
mi[#1], {2}]] & ) /@ m3; m3AllSubsets = DeleteDuplicates[Catenate[m3Subsets]]; m3OptimalSubsets = If[m3AllSubsets == {}, {}, im/@m3AllSubsets[[Transpose[Position[LinearProgramming[ Table[1., {i, Length[m3AllSubsets]}], (ReplacePart[Table[  
0, {i, Length[m3AllSubsets]}], Transpose[{#1}] -> 1] & ) /@ Ta- ble[FirstPosition[m3AllSubsets, m3Subsets[[r,j]]][[1]],  
{r, Length[m3]}, {j, Length[m3Subsets[[r]]]}],  
Table[1, {i, Length[m3]}], Table[{0, 1}, {i, Length[m3AllSubsets]}], Integers], 1]][[1]]]]]; n = 100;  
{(1/n)\*AbsoluteTiming[Do[m /. rules, {n}]][[1]], Length[m], Length[m3OptimalSubsets]}, {10}];   
numberMin = Catch[Do[If[((#1[[1]] & ) /@ all)[[i]] == Min[all[[All,1]]], Throw[i], {0}], {i, Length[all]}]];  
numberMax = Catch[Do[If[((#1[[1]] & ) /@ all)[[i]] == Max[all[[All,1]]], Throw[i], {0}], {i, Length[all]}]];  
minimum = all[[numberMin]]; maximum = all[[numberMax]]; resultMinimum = minimum[[1]]/((minimum[[2]] + mini- mum[[3]])\*5.15\*^-6); resultMaximum = maximum[[1]]/((maximum[[2]] + maximum[[3]])\*5.15\*^-6); Grid[{{"", "\!\(\\*SubscriptBox[\(N\), \(v\)]\)", "\!\(\\*SubscriptBox[\(N\), \(m\)]\)",

"\!\(\\*SubsuperscriptBox[\(N\), \(m\), \(+\)]\)",

"t/\!\(\\*SubscriptBox[\(t\), \(s\)]\)"}, {min, variable, amount3degree, minimum[[3]], resultMinimum}, {max, variable, amount3degree, maximum[[3]], resultMaximum}}, Frame -> All, Spacings -> {2, 2}]