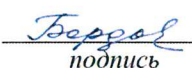


Министерство науки и высшего образования Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)  
Кафедра астрономии и космической геодезии

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК  
Руководитель ООП  
д-р физ.-мат. наук, профессор

 Т.В. Бордовицына  
подпись  
« 14 » июня 2021 г.

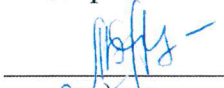
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА  
(МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

РАЗРАБОТКА ГРАФИЧЕСКОГО ИНТЕРФЕЙСА К ПРОГРАММНОМУ КОМПЛЕКСУ  
ДЛЯ ИССЛЕДОВАНИЯ НЕЛИНЕЙНОСТИ В ОБРАТНЫХ ЗАДАЧАХ  
СПУТНИКОВОЙ ДИНАМИКИ

по направлению подготовки 09.04.02 – Информационные системы и технологии  
направленность (профиль) «Информационные системы и технологии в геодезии  
и картографии»

Шмидт Екатерина Евгеньевна

Руководитель ВКР  
канд. физ.-мат. наук, доцент

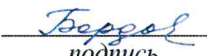
 М.А. Баньщикова  
подпись  
« 14 » июня 2021 г.

Автор работы  
студент группы № 051971

 Е.Е. Шмидт  
подпись  
« 14 » июня 2021 г.

Министерство науки и высшего образования Российской Федерации.  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)  
Кафедра астрономии и космической геодезии

УТВЕРЖДАЮ  
Руководитель ООП  
д-р физ.-мат. наук, профессор

 Т.В. Бордовицына  
подпись  
« 16 » сентября 2021 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра / специалиста / магистра  
обучающегося

Шмидт Екатерина Евгеньевна

*Фамилия Имя Отчество обучающегося*

по направлению подготовки 09.04.02 – Информационные системы и технологии,  
направленность (профиль) «Информационные системы и технологии в геодезии и  
картографии»

1 Тема выпускной квалификационной работы

Разработка графического интерфейса к программному комплексу для исследования  
нелинейности в обратных задачах спутниковой динамики

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) в учебный офис / деканат – июнь 2021 г. б) в ГЭК – июнь 2021 г.

3 Исходные данные к работе:

Объект исследования – далекие спутники Юпитера

Предмет исследования – нелинейность в обратных задачах спутниковой динамики

Цель исследования – оценить нелинейность в обратных задачах спутниковой  
динамики для всех новых внешних спутников Юпитера

Задачи:

разработать графический интерфейс для исследования нелинейности обратных задач  
динамики реальных и модельных объектов

Методы исследования:

анализ, моделирование, синтез

Организация или отрасль, по тематике которой выполняется работа, –  
Томский государственный университет (ТГУ)

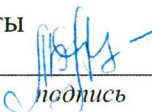
4 Краткое содержание работы

1 глава – Описание далеких спутников Юпитера, их орбит и численной модели движения; 2  
глава – Исследование нелинейности в обратных задачах спутниковой динамики; 3 глава –  
Разработка графического интерфейса JUNO

Руководитель выпускной квалификационной работы

канд. физ.-мат. наук, доцент, ТГУ

*должность, место работы*

  
подпись

М.А. Баньшикова

*И.О. Фамилия*

Задание принял к исполнению

студент группы № 051971

*должность, место работы*

  
подпись

Е.Е. Шмидт

*И.О. Фамилия*

## АННОТАЦИЯ

Выпускная квалификационная работа магистра «Разработка графического интерфейса к программному комплексу для исследования нелинейности в обратных задачах спутниковой динамики» содержит 50 страниц печатного текста, 18 рисунков, 2 таблицы, 15 формул и 31 источник использованной литературы.

Ключевые слова: внешние спутники Юпитера, нелинейность в обратных задачах спутниковой динамики.

Цель работы – оценка нелинейности в обратных задачах орбитальной динамики для всех новых внешних спутников.

В работе представлены результаты исследования полной и внутренней нелинейностей в обратных задачах для 63 новых далеких спутников Юпитера. Выявлена взаимосвязь между нелинейностью и условиями спутниковых наблюдений. В частности, показано, что полная нелинейность очень сильная, когда период наблюдений меньше 0.1 от орбитального периода и когда окна между наблюдательными данными составляют порядка несколько лет, в то время как внутренняя нелинейность достаточно слабая почти для всех спутников, что указывает на возможность применения нелинейных методов для адекватного моделирования их орбитальной неопределенности. А также подробно описана разработка графического интерфейса JUNO для расчета показателей нелинейности обратных задач динамики реальных и модельных объектов и представления результатов в виде таблицы и графика в самой программе и с сохранением их в файл.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1 Внешние спутники Юпитера.....	5
1.1 Описание орбит внешних спутников Юпитера.....	5
1.2 Численная модель движения спутников Юпитера.....	7
1.3 Определение орбитальных параметров из наблюдений.....	9
2 Исследования нелинейности в обратных задачах спутниковой динамики .....	12
2.1 Общий анализ нелинейности.....	12
2.2 Показатели полной и внутренней нелинейности .....	13
2.3 Численные результаты. Модельные объекты .....	14
2.4 Нелинейность для внешних спутников Юпитера .....	17
3 Разработка графического интерфейса JUNO.....	21
3.1 Этапы разработки.....	21
3.1.1 Вкладка Real.....	23
3.1.2 Вкладка Model.....	29
3.1.3 Окно Result .....	31
3.1.4 Вкладка Orbit.....	35
3.1.5 Вкладка Satellites.....	37
3.1.6 Вкладка Help .....	38
3.2 Описание интерфейса.....	41
3.3 Работа в JUNO. Представления результатов .....	42
ЗАКЛЮЧЕНИЕ .....	47
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ .....	48

## ВВЕДЕНИЕ

Большинство внешних спутников Юпитера было открыто в самом начале XXI столетия [1-3]. Между тем орбиты многих из этих спутников до сих пор плохо определены вследствие скудного состава их наблюдений, несмотря на то что со времени открытия спутников прошло уже почти два десятка лет. Для некоторых объектов орбиты определены настолько ненадежно, что даже не совсем ясно, являются ли они вообще спутниками или это — астероиды, временно захваченные Юпитером [4]. Ответить на этот вопрос можно только исходя из вероятностных оценок, полученных на основе стохастического моделирования орбитальной неопределенности [3], [5], [6], [7]. Поскольку для новых представителей спутниковой системы Юпитера неопределенность в орбитальных параметрах достаточно большая, подход к ее моделированию весьма деликатный. Исследователь становится перед выбором метода стохастического моделирования: линейного или нелинейного [3], [5], [6], [7]. Линейные методы — очень простые, но недостаточно точные при сильной нелинейности обратной задачи, которая как раз сопряжена с большой орбитальной неопределенностью. Все нелинейные методы — очень сложные и трудоемкие, но более точные. При этом всегда нужно иметь в виду, что адекватность нелинейных методов определяется так называемой внутренней нелинейностью [7-11].

Таким образом целью работы является оценка нелинейности в обратных задачах орбитальной динамики для всех новых внешних спутников.

Задача — разработать графический интерфейс для исследования нелинейности обратных задач динамики реальных и модельных объектов.

### Актуальность работы

Разработанный интерфейс для исследования нелинейности обратных задач динамики реальных и модельных объектов актуален для исследователей в области спутниковой динамике.

## Научная новизна

Впервые получены оценки полной и внутренней нелинейности для всех далеких спутников Юпитера, с использованием новых наблюдений, добавленных в базу данных NSDB [13] до 28 января 2021 г. Построен графический интерфейс JUNO для исследования этого вопроса.

## Практическая значимость работы

Используя графический интерфейс и зная только распределение наблюдений по дуге орбиты, можно оценить параметры внутренней и полной нелинейности и решить какой из методов стохастического моделирования линейный или нелинейный применять для исследования.

## Степень достоверности и апробация результатов

Теоретическая часть исследования нелинейности в обратных задачах спутниковой динамики была представлена на двух научных конференциях:

1. X Всероссийская научная конференция с международным участием «Актуальные проблемы современной механики сплошных сред и небесной механики» (18-20 ноября 2020 г., Томск, НИИ ПММ ТГУ).

2. 50-ая Международная студенческая научная конференция «Физика космоса» (1–5 февраля 2021 г., Екатеринбург).

А также сдана в печать статья (М.А. Баньщикова, В.А. Авдюшев, Е.Е. Шмидт «Нелинейность в обратных задачах динамики внешних спутников Юпитера») в Астрономический вестник, которая выйдет в 6 номере 2021 г.

Представленные в данной работе результаты включены в отчеты по проекту гранта Российского научного фонда (проект № 19-72-10022).

## Благодарности

Автор выражает благодарность своему научному руководителю, М.А. Баньщиковой за помощь в выборе темы магистерской диссертации, ценные советы и поддержку и В.А. Авдюшеву за консультации.

## 1 Внешние спутники Юпитера

### 1.1 Описание орбит внешних спутников Юпитера

Внешние спутники Юпитера – это весьма малые представители Солнечной системы. Их орбиты нерегулярны (с большими эксцентриситетами и наклонениями) и ввиду существенной удаленности от Юпитера далекие спутники сильно возмущаются Солнцем.

По данным 2019 года всего известно 79 спутников Юпитера. Из них 4 спутника внутренние, 4 – галилеевские и 71 – внешние. Далекие спутники нерегулярны. Все нерегулярные спутники можно разделить на определенные группы или классы. Это – спутники групп Карпо (прямое движение), Гималии и Ананке, Карме, Пасифе (обратное движение). Все они – малые небесные объекты. Их средние расстояния от центра Юпитера – в пределах 17–28 млн.км; наклонения — 140–165 град. (кроме Карпо: 51 град.; S/2018\_J1: 28 град; Гималии, S/2000\_J1, S/2017\_J4: 28 град.; Эллары и Лиситеи: 27 град.; Фемисто: 43 град.); периоды обращения – до трех лет. Ввиду удаленности от Юпитера внешние спутники испытывают достаточно сильное гравитационное влияние от Солнца. Количество измерений спутников варьируется от 238 (Эвропии) до 10 (S2003/J02) и они покрывают временные интервалы от 0.03 (S2003/J23) до 1.9 оборотов (Каллирое, Праксидике).

В Таблице 1 приведена основная информация о 63 новых далеких спутниках Юпитера: средние орбитальные элементы ( $a$  – большая полуось,  $i$  – наклонение,  $e$  – эксцентриситет,  $W$  – аргумент перицентра,  $\Omega$  – долгота восходящего узла,  $M$  – средняя аномалия,  $T$  – период) и  $L$  – количество наблюдений [12]. Наблюдения для внешних спутников Юпитера были взяты из базы данных NSDB [13] в январе 2021 г.

Таблица 1 – средние орбитальные элементы далеких спутников Юпитера [12]

Спутник	a(тыс.км)	i °	e	W °	$\Omega$ °	M °	T	L
Callirhoe	24 102	147.1	0.283	49.3	281.1	321.7	758.8	195
Themisto	7 507	43.08	0.242	240.7	201.5	134.2	130.0	107
Megaclite	23 806	152.8	0.421	302.3	304.6	352.5	752.8	83
Taygete	23 360	165.2	0.252	241.1	313.3	317.9	732.2	58
Chaldene	23 179	165.2	0.251	282.5	148.7	131.6	723.8	72
Harpalyke	21 105	148.6	0.226	129.9	40.0	120.4	623.3	91
Kalyke	23 583	165.2	0.245	216.6	38.7	116.3	743.0	93
Iocaste	21 269	149.4	0.216	80.0	271.3	129.8	631.5	82
Erinome	23 279	164.9	0.266	356.0	321.7	131.1	728.3	75
Isonoe	23 217	165.2	0.246	145.6	149.8	345.4	725.5	96
Praxidike	21 147	149.0	0.230	209.7	285.2	21.8	625.3	85
Autonoe	23 039	152.9	0.334	60.2	275.6	182.7	762.7	52
Thyone	20 940	148.5	0.229	89.1	243.0	26.6	627.3	50
Hermippe	21 131	150.7	0.210	298.7	347.2	276.4	633.9	75
Aitne	23 231	165.1	0.264	122.2	24.5	153.9	730.2	59
Eurydome	22 865	150.3	0.276	241.6	307.4	340.7	717.3	47
Euanthe	20 799	148.9	0.232	316.0	271.0	130.5	620.6	26
Europie	19 302	145.8	0.144	74.6	64.9	293.0	550.7	238
Orthosie	20 721	145.9	0.281	230.5	223.6	326.8	622.6	34
Sponde	23 487	151.0	0.312	79.1	129.1	216.8	748.3	34
Kale	23 217	165.0	0.260	44.4	56.4	286.4	729.5	44
Pasithee	23 096	165.1	0.267	253.3	338.7	272.7	719.5	30
Hegemone	23 947	155.2	0.328	235.4	327.6	98.73	739.6	43
Mneme	21 069	148.6	0.227	41.7	18.1	338.9	620.0	81
Aoede	23 981	158.3	0.432	74.5	187.1	74.0	761.5	63
Thelxinoe	21 162	151.4	0.221	179.8	206.2	194.0	628.1	56
Arche	22 931	165.0	0.259	161.1	350.7	126.7	723.9	48
Kallichore	24 043	165.5	0.264	18.5	41.5	319.4	764.7	43
Helike	21 263	154.8	0.156	314.7	100.3	36.2	634.8	46
Carpo	16 989	51.4	0.430	90.0	60.9	242.8	456.1	43
Eukelade	23 661	165.5	0.272	325.6	206.3	98.4	746.4	63
Cyllene	24 349	149.3	0.319	214.0	266.4	000.4	737.8	28
Kore	24 543	145.0	0.325	152.4	324.7	247.1	779.2	35
Herse	23 097	164.2	0.200	355.7	329.0	41.9	715.4	54
S/2010_J1	23 314	163.2	0.320	000.0	000.0	00.0	723.2	171
S/2010_J2	20 307	150.4	0.307	000.0	000.0	00.0	588.1	116
Dia	12 118	28.23	0.211	178.0	290.9	169.9	287.0	23
S/2016_J1	20 651	139.8	0.141	328.2	293.8	25.5	602.7	15
S/2003_J18	20 274	146.4	0.105	98.15	215.5	220.7	588.0	53



Продолжение таблицы 1

S/2011_J2	23 464	148.8	0.332	341.9	105.2	292.5	730.5	36
S/2003_J5	23 732	163.1	0.220	160.0	256.3	284.4	759.7	34
S/2003_J15	22 820	143.6	0.194	52.08	286.5	161.6	701.3	23
S/2017_J1	23 547	149.2	0.397	79.73	326.3	161.3	734.2	18
S/2003_J3	21 200	148.0	0.253	109.0	292.0	168.7	627.8	26
S/2003_J19	22 757	166.7	0.257	284.1	105.7	205.7	697.6	24
S/2016_J2	18 980	34.0	0.222	116.8	240.3	160.3	533.3	21
S/2017_J2	23 303	166.4	0.236	231.9	74.7	344.4	723.1	16
S/2017_J3	20 694	147.9	0.148	171.6	82.5	91.2	606.3	19
S/2017_J4	11 525	28.15	0.180	238.8	228.4	282.4	252.1	20
S/2017_J5	23 232	164.3	0.284	11.9	118.2	77.5	719.5	15
S/2017_J6	22 455	155.2	0.557	77.3	26.3	184.5	683.0	16
S/2017_J7	20 627	143.4	0.215	323.5	321.5	236.9	602.6	18
S/2017_J8	23 233	164.7	0.312	45.1	160.3	70.8	719.6	13
S/2017_J9	21 487	152.7	0.229	317.7	306.4	259.5	639.2	15
S/2018_J1	11 483	30.61	0.094	346.7	93.3	356.6	252.0	17
S/2011_J1	20 155	162.8	0.296	000.0	000.0	00.0	580.7	35
S/2003_J2	28 570	151.8	0.380	000.0	000.0	00.0	982.5	10
S/2003_J4	23 258	144.9	0.204	000.0	000.0	00.0	723.2	11
S/2003_J9	22 442	164.5	0.269	000.0	000.0	00.0	683.0	17
S/2003_J10	24 250	164.1	0.214	000.0	000.0	00.0	767.0	11
S/2003_J12	19 002	145.8	0.376	000.0	000.0	00.0	533.3	11
S/2003_J16	21 000	148.5	0.230	000.0	000.0	00.0	595.4	98
S/2003_J23	24 055	149.2	0.309	000.0	000.0	00.0	759.7	16

## 1.2 Численная модель движения спутников Юпитера

Формально численную модель спутникового движения  $p_c$  в пространстве угловых координат  $p = (p_1, p_2)$  относительно стандартного земного экватора J2000.0 можно представить в виде

$$p^c = p^c(t, q^{DT}) = T(t, x(t, q^D), q^T), \quad (1.1)$$

где  $t$  – эфемеридное время;  $q^{DT} = (q^D, q^T)$  – вектор всех параметров модели;  $T$  – преобразование перехода от йовицентрической системы координат к топоцентрической;  $q^D = (x_0, \dot{x}_0, t_0, q_s, \dots)$  и  $q^T$  – параметрические векторы, связанные с движением спутника относительно йовицентра и с координатным преобразованием;  $x$  – йовицентрическое положение спутника;  $x_0$  и  $\dot{x}_0$  –

векторы динамического состояния спутника в начальный момент времени  $t_0$  [14].

В модели (1.1) положение  $x$  определяется путем численного интегрирования дифференциальных уравнений движения (1.2)

$$\frac{d^2x}{dt^2} = P(t, x, \dot{x}, q^D) = P_J + P_G + P_{SP} + P_R \quad (1.2)$$

с начальными условиями  $x_0 = x(t_0), \dot{x}_0 = \dot{x}(t_0)$ , где учитываются влияние гравитационного поля Юпитера  $P_J$ , от Солнца и планет  $P_{SP}$ , а также релятивистские эффекты  $P_R$  в рамках задачи Шварцшильда [14]. Уравнения интегрируются методом Гаусса-Эверхарта [14].

В модели Юпитер рассматривается как сфероид, притяжение которого учитывается с точностью до шестой зональной гармоники [15]. Формально притяжение планеты можно представить в виде (1.3)

$$P_J = \frac{\partial U}{\partial x}, U = -\mu \sum_i J_i \frac{b_J^i}{|x|^{i+1}} L_i(\sin \psi_J), \sin \psi_J = \frac{x_{J3}}{|x|}, \quad (1.3)$$

где  $\mu$  – гравитационный параметр планеты;  $J_i$  – коэффициенты зональных гармоник, причем  $J_0 = -1$ ;  $b_J$  – экваториальный радиус Юпитера;  $L_i$  – полиномы Лежандра;  $\psi_J$  и  $x_{J3}$  – широта и аппликата спутника соответственно относительно экватора планеты [14]. При этом аппликата  $x_{J3}$  определяется из координатного преобразования  $x_J = T_J x$  [13], где  $T_J$  – матрица перехода от гео- к йовиэкватору. Кроме того, при вычислении  $\psi_J$  учитывается прецессия экватора Юпитера [16]. Параметры гравитационного поля взяты из теории Лиске [17].

Ввиду малых размеров гравитирующих тел: галилеевых спутников, Солнца и планет (за исключением Юпитера), по отношению к их расстояниям до спутников, они рассматриваются как материальные точки, а притяжение

каждого тела в  $P_G$  и  $P_{SP}$  (1.1) вычисляется по формуле (1.4)

$$P_i = \mu_i \left( \frac{x_i - x}{|x_i - x|^3} - \frac{x_i}{|x_i|^3} \right), \quad (1.4)$$

где  $\mu_i$  и  $x_i$  – соответственно гравитационный параметр и йовицентрический вектор положения  $i$ -го тела [14]. Координаты галилеевых спутников вычисляются по теории Леней [18], а координаты Солнца и планет (Сатурн, Уран, Нептун) – из эфемериды DE405 [19].

Возмущения, связанные с релятивистскими эффектами, учитывались в рамках задачи Шварцшильда [20] (1.5):

$$P_R = 4 \frac{\mu^2}{c^2 |x|^4} x + \frac{\mu}{c^2 |x|^3} [4(x \cdot \dot{x}) \cdot \dot{x} - \dot{x}^2 x], \quad (1.5)$$

где  $c$  – скорость света.

### 1.3 Определение орбитальных параметров из наблюдений

Пусть имеем  $N$  наблюдаемых положений  $p_i^O$  в  $L$ -мерном пространстве на моменты времени  $t_i (i=1, \dots, N)$ . Требуется по наблюдениям  $p_i^O$  определить  $K$  орбитальных параметров  $q$  так, чтобы они наилучшим образом удовлетворяли переопределенной системе уравнений  $p_i^O = p_i^C(q) (i=1, \dots, N)$ , или  $P^O = P^C(q)$ , где  $p_i^C = p^C(t_i, q)$  – модельное представление (1.1) наблюдений  $p_i^O$  на момент  $t_i$ , и введено обозначение  $P = (p_1, \dots, p_N)^T$  [14].

Решение обратной задачи обычно сводится к минимизации функционала (1.6)

$$S(q) = \frac{1}{2} \sum_{i=1}^N \rho^2(p_i^O, p_i^C), \quad (1.6)$$

где  $\rho$  – метрика в пространстве  $p$ , а  $p_i^C = p^C(t_i, q)$  – численное представление наблюдений  $p_i^O$  по  $q$ , на основе моделирования орбитального движения.

Зададим метрику как  $\rho(p_1, p_2) = \sqrt{(p_{11} - p_{21})^2 + \dots + (p_{1L} - p_{2L})^2}$ , где  $p_{j1}, \dots, p_{jL}$  –

компоненты  $j$ -вектора  $p_j$ . Тогда (1.1) можно представить в виде

$$S(q) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^L (p_{ij}^o - p_{ij}^c)^2 = \frac{1}{2} (P^o - P^c)^2. \quad (1.7)$$

Минимум функционала (1.7) находится из необходимого условия экстремума по  $q = (q_1, \dots, q_K)$ :

$$\frac{\partial S}{\partial q} = - \sum_{i=1}^N \sum_{j=1}^L (p_{ij}^o - p_{ij}^c) \frac{\partial p_{ij}^c}{\partial q} = 0. \quad (1.8)$$

Чтобы решить систему уравнений (1.8) относительно параметров  $q$ , прибегают к итерационным методам. Согласно итерационной схеме Ньютона поправка к текущему приближению  $q$  определяется как

$$\Delta q = - \left( \frac{\partial^2 S}{\partial p^2} \right)^{-1} \frac{\partial S}{\partial q}, \quad (1.9)$$

где  $\partial^2 S / \partial q^2$  – матрица Гессе:

$$\frac{\partial^2 S}{\partial q^2} = \sum_{i=1}^N \sum_{j=1}^L \left[ \frac{\partial p_{ij}^c}{\partial q} \frac{\partial p_{ij}^c}{\partial q} - (p_{ij}^o - p_{ij}^c) \frac{\partial^2 p_{ij}^c}{\partial q^2} \right]. \quad (1.10)$$

В схеме (1.9) начальное приближение  $q$  обычно берется из предварительного определения орбиты по нескольким наблюдениям в рамках задачи двух тел [14].

Ввиду проблематичного вычисления вторых производных в (1.10) на практике их опускают и, таким образом, используют упрощенную схему так называемого метода Гаусса-Ньютона, который относится к широкому классу квазиньютоновских методов [14].

Введем матрицы (1.11)

$$A_{NL \times K} = \begin{pmatrix} \frac{\partial p_{11}^c}{\partial q_1} & \dots & \frac{\partial p_{11}^c}{\partial q_K} \\ \vdots & \ddots & \vdots \\ \frac{\partial p_{1L}^c}{\partial q_1} & \dots & \frac{\partial p_{1L}^c}{\partial q_K} \\ \frac{\partial p_{21}^c}{\partial q_1} & \dots & \frac{\partial p_{2L}^c}{\partial q_K} \\ \vdots & \ddots & \vdots \\ \frac{\partial p_{NL}^c}{\partial q_1} & \dots & \frac{\partial p_{NL}^c}{\partial q_K} \end{pmatrix} \text{ и } B_{NL \times 1} = \begin{pmatrix} p_{11}^o - p_{11}^c \\ \vdots \\ p_{1L}^o - p_{1L}^c \\ p_{21}^o - p_{21}^c \\ \vdots \\ p_{NL}^o - p_{NL}^c \end{pmatrix}. \quad (1.11)$$

Тогда поправку (1.9) можно переписать как (1.12)

$$\Delta q = -Q^{-1}G, \quad (1.12)$$

где  $Q = A^T A$  – так называемая нормальная матрица, которая в минимуме  $S(q)$  при достаточно малых невязках  $B$  близка к матрице Гессе, а  $G = -A^T B$  – градиент функции  $S$  по  $q$ . Элементы матрицы  $A$  являются изохронные производные, которые вычисляются численно из дифференциальных уравнений в вариациях совместно с уравнениями движения [14].

На практике часто наблюдаются угловые координаты объекта: прямое восхождение  $p_{i1}^o = \alpha_i^o$  и  $p_{i2}^o = \delta_i^o$  склонение. В этом случае расстояния между наблюдаемыми и вычисленными положениями на небесной сфере вычисляются по формуле (1.13):

$$\rho(p_i^o, p_i^c) = \sqrt{(\alpha_i^o - \alpha_i^c)^2 \cos^2 \delta_i^o + (\delta_i^o - \delta_i^c)^2}, \quad (1.13)$$

тогда как в качестве функционала (1.7) рассматривают (1.14)

$$S(q) = \frac{1}{2} \sum_{i=1}^N [(\alpha_i^o - \alpha_i^c)^2 \cos^2 \delta_i^o + (\delta_i^o - \delta_i^c)^2]. \quad (1.14)$$

## 2 Исследования нелинейности в обратных задачах спутниковой динамики

### 2.1 Общий анализ нелинейности

Взаимосвязь между обстоятельствами наблюдений и нелинейностью можно наглядно представить на примере нормализованной задачи двух тел. Мы рассматривали движение спутника на круговой орбите единичного радиуса с периодом  $2\pi$ . В качестве определяемых параметров мы приняли компоненты вектора начального динамического состояния  $\mathbf{q} = (\mathbf{x}_0, \dot{\mathbf{x}}_0)^T$  ( $\dim \mathbf{q} = 6$ ), где  $\mathbf{x}_0$  и  $\dot{\mathbf{x}}_0$  — положение и скорость спутника на нулевой момент времени ( $t_0 = 0$ ). Моделируя  $L$  положений объекта  $\mathbf{x}_i$  ( $i = 1, \dots, L$ ) и отображая их на картинную плоскость  $(X, Y)$ , мы получали, таким образом, вектор наблюдаемых величин  $\mathbf{p} = (X_1, Y_1, \dots, X_L, Y_L)^T$  ( $\dim \mathbf{p} = 2L$ ).

Координаты спутника в картинной плоскости (Рисунок 2.1) рассчитывались как  $X_i = x_{1i}$ ,  $Y_i = x_{2i} \cos \psi - x_{3i} \sin \psi$  ( $i = 1, \dots, L$ ), где  $\psi \in [0, \pi/2]$  — угол между картинной и орбитальной плоскостями.

Наблюдения моделировались на трех орбитальных дугах:  $\Delta t/2\pi = 0.1, 0.01, 0.001$ , на четыре равномерно распределенных момента времени ( $L = 4$ ):  $t_i = \Delta t(i-1)/(L-1)$  ( $i = 1, \dots, L$ ).

Мы намеренно выбрали предельно минимальное количество наблюдений, с тем чтобы создать критические условия для обратной задачи, когда эффект нелинейности максимален. При  $L = 3$  задача уже сводится к предварительному определению орбиты по трем наблюдениям, и тогда внутренняя нелинейность становится нулевой.

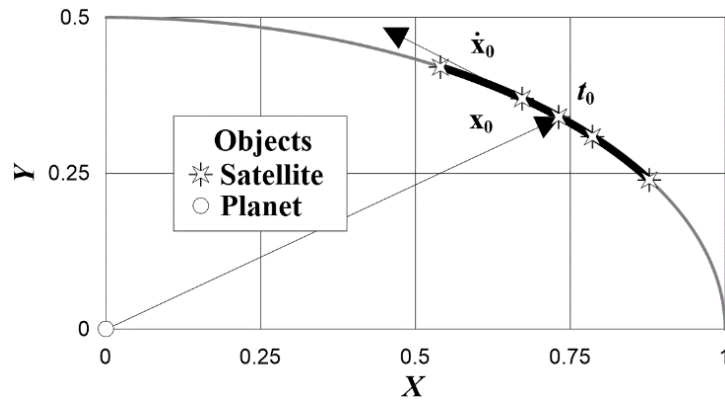


Рисунок 2.1 – Наблюдения спутника в картинной плоскости

## 2.2 Показатели полной и внутренней нелинейности

Показатели полной и внутренней нелинейности вычислялись по формулам (2.1)

$$\varepsilon = \frac{\|A\Delta q - \Delta p\|}{\|\Delta p\|}, \quad \varepsilon_t = \frac{\|\Pi\Delta p - \Delta p\|}{\|\Delta p\|}, \quad \Pi = A(A^T A)^{-1} A^T, \quad (2.1)$$

где  $A = \partial p / \partial q$  — матрица частных производных от модельных представлений наблюдений по параметрам;  $\Delta q = \sqrt{\chi_{6,0.997}^2 \lambda_{\max}} l_{\max}$  — вариация параметров; — 99.7%-квантиль функции распределения  $\chi_6^2$  с 6-ю степенями свободы;  $l_{\max}$  — собственный вектор ковариационной матрицы  $C = \sigma^2 (A^T A)^{-1}$ , который соответствует максимальному собственному числу  $\lambda_{\max}$ ;  $\sigma = 0.2''$  — среднеквадратическая ошибка наблюдений (выбранное значение типичное для ошибок наземных ПЗС-наблюдений);  $\Delta p = p(q + \Delta q) - p(q)$  — вариация в векторе наблюдений, вызванная вариацией параметрического вектора  $\Delta q$ .

Полная (параметрическая) нелинейность [8] проявляется, главным образом, в неравномерном изменении модельного отклика на равномерное изменение параметров модели, а также в тангенциальной (геодезической) кривизне параметрических линий в подпространстве оценок. Теоретически существуют такие (нелинейные) преобразования параметров, которые сводят параметрическую нелинейность к нулю. Однако представить их в аналитической форме практически невозможно.

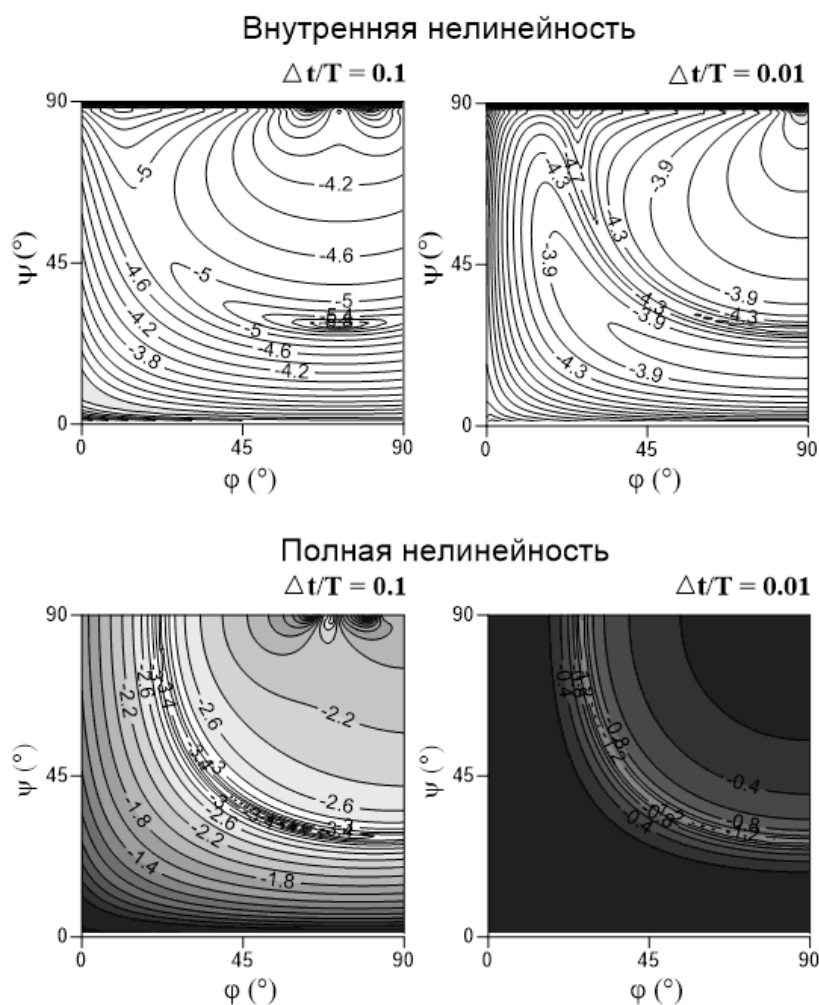
Внутренняя нелинейность [8] обусловлена непосредственно нормальной кривизной подпространства оценок и не зависит от преобразования параметров. Поэтому она неустранима и, следовательно, определяет предельный уровень минимальной нелинейности, которая может быть достигнута в результате преобразований параметров, т.е. при  $\varepsilon = 0$ . Если  $\varepsilon_i = 0$ , а  $\varepsilon \neq 0$ , то говорят, что обратная задача или модель внутренне линейна (intrinsically linear) или внешне нелинейная (nonintrinsically nonlinear).

### 2.3 Численные результаты. Модельные объекты

На рисунке 2.2 показаны значения показателей нелинейности (2.1) в зависимости от начального положения объекта, определяемого аномалией  $\varphi$ , и от угла между картиной и орбитальной плоскостями  $\psi$  на двух орбитальных дугах:  $\Delta t / 2\pi = 0.1, 0.01$ . Заливка полутонами серого цвета на рисунке применяется только к уровням показателей, превышающих критическое значение  $10^{-3}$  [8].

Как видно из рисунка 2.2, внутренняя нелинейность при всех обстоятельствах наблюдений довольно слабая, что является практическим основанием для применения нелинейных методов стохастического моделирования при исследовании орбитальной неопределенности. Полная нелинейность значительно сильнее и ее показатели в большинстве случаев превышают критическое значение. Слабая полная нелинейность имеет место лишь при определенных обстоятельствах, когда  $\Delta t / 2\pi = 0.1$ . На рисунке им соответствуют угловые величины около оврага, проходящего через значения  $\varphi = 30^\circ \& \psi = 75^\circ$ ,  $\varphi = 45^\circ \& \psi = 45^\circ$ ,  $\varphi = 75^\circ \& \psi = 30^\circ$ .





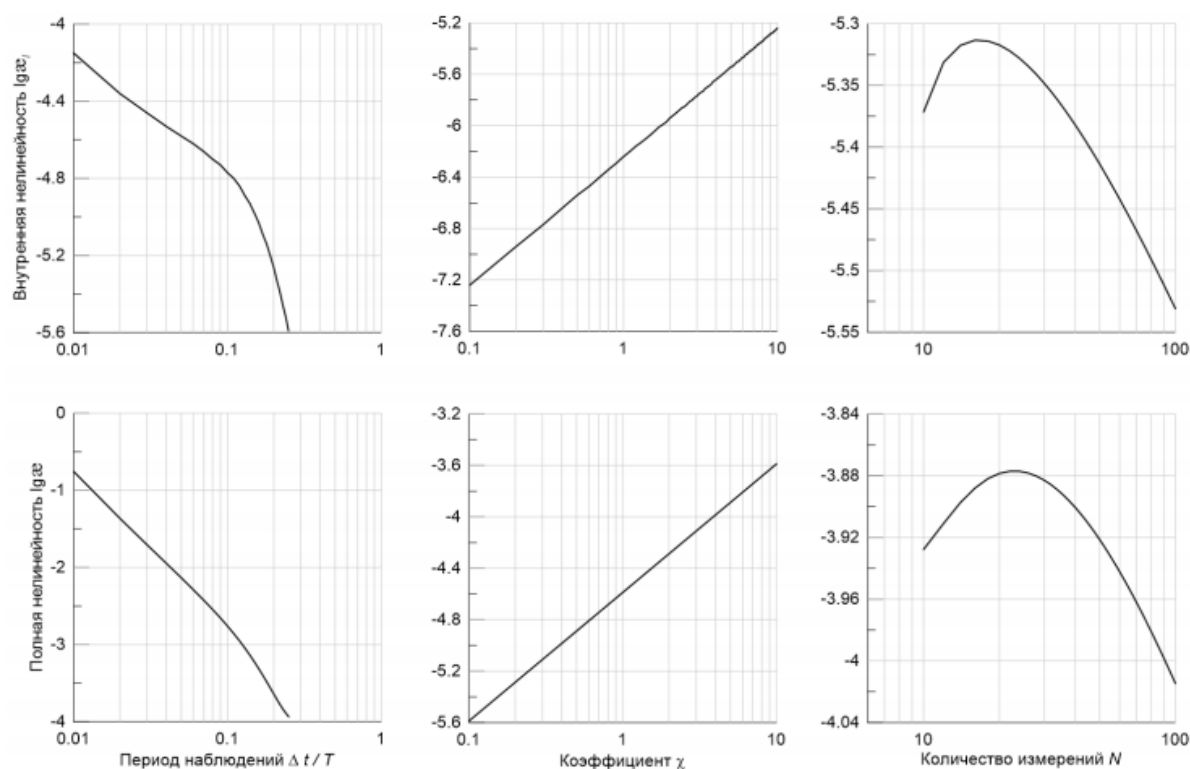


Рисунок 2.3 – Показатели внутренней  $\lg \alpha_i$  и полной  $\lg \alpha$  нелинейности в зависимости от изменения длины наблюдаемой орбитальной дуги  $\Delta t / T$ , коэффициента  $\chi_{6,0.997}$  и количества измерений  $N$

Как видно из графиков увеличивая продолжительность наблюдений  $\Delta t / T$  (Рисунок 2.3, первый столбец) видно что внутренняя нелинейность менее чувствительна к вариациям длины орбитальной дуги, нежели полная, она даже при очень малой дуге еще достаточно слабая, тогда как полная нелинейность становится чрезвычайно сильной.

Очевидно, что с удалением от оценки в параметрическом пространстве влияние нелинейности усиливается. Показатели нелинейности вводятся для максимально удаленной от оценки точки на границе доверительного эллипсоида вероятности  $\alpha$ . Размер доверительного эллипсоида задается квантилем  $\chi_{K,\alpha}$  и среднеквадратической ошибкой  $\sigma$ , точнее, их произведением. Чтобы оценить зависимость нелинейности от удаленности от оценки, мы проварьировали лишь значения квантиля  $\chi_{6,0.997}$ . Как видно из Рисунка 2.3 (второй столбец), десятичный логарифм показатели нелинейности

увеличиваются и стремятся к нулю с увеличением размера принятой доверительной области, на границе которой оценивается нелинейность. В частности, поэтому высокоточные наблюдения являются залогом слабой нелинейности. Кроме того, размер доверительного эллипсоида зависит от количества измерений  $N/2L$ : чем больше измерений, тем больше собственные числа нормальной матрицы и тем меньше размер эллипсоида. Как видно из Рисунка 2.3 (третий столбец), в спутниковой динамике, увеличение количества наблюдений ослабляют нелинейность, но при этом количество измерений влияет на нелинейность меньше, чем размер наблюдаемой орбитальной дуги (первый столбец).

## 2.4 Нелинейность для внешних спутников Юпитера

В данном разделе мы приводим результаты исследования нелинейности для 63 представителей групп внешних спутников Юпитера.

В Таблице 2 приведены значения показателей нелинейности для всех новых спутников Юпитера. Светло-серым цветом отмечены спутники с умеренной нелинейностью, а темно-серым — с сильной нелинейностью. Результаты в таблице без заливки — нелинейность слабая. Показатели нелинейности для всех спутников также представлены графически на Рисунке 2.4: белая гистограмма — полная нелинейность; серая — внутренняя. Числа на белой гистограмме —  $\Delta t/T$ .

Таблица 2 — Показатели нелинейности для новых далеких спутников Юпитера

Спутник	$\varepsilon$	$\varepsilon_l$	$L$	$\Delta t/T$	Спутник	$\varepsilon$	$\varepsilon_l$	$L$	$\Delta t/T$
<b>2017J02</b>	0.0003	0.0000	16	1.11	<b>Kalyke</b>	0.0074	0.0001	93	8.16
<b>Cyllene</b>	0.0003	0.0000	28	7.27	<b>Kale</b>	0.0074	0.0000	44	8.17
<b>Arche</b>	0.0005	0.0000	48	7.69	<b>Hermippe</b>	0.0083	0.0001	75	9.27
<b>2017J03</b>	0.0009	0.0005	19	1.30	<b>Eurydome</b>	0.0088	0.00001	47	8.81
<b>2017J01</b>	0.0009	0.0000	18	1.14	<b>Praxidike</b>	0.0092	0.0001	85	10.00
<b>2003J19</b>	0.0012	0.0001	24	7.41	<b>Taygete</b>	0.0093	0.0003	58	9.05
<b>Mneme</b>	0.0013	0.0000	81	8.73	<b>Herse</b>	0.0105	0.0000	54	7.91
<b>Magaclite</b>	0.0017	0.0001	83	7.87	<b>2003J18</b>	0.0162	0.0001	53	9.06
<b>2016J01</b>	0.0017	0.0002	15	0.66	<b>2011J02</b>	0.0162	0.0001	36	3.29
<b>Kallichore</b>	0.0018	0.0000	43	6.86	<b>Euporie</b>	0.0173	0.0003	38	10.40
<b>Erinome</b>	0.0018	0.0001	75	8.43	<b>Chaldene</b>	0.0178	0.0002	72	9.52
<b>Autonoe</b>	0.0020	0.0000	52	7.71	<b>2017J05</b>	0.0212	0.0000	14	0.59
<b>Kore</b>	0.0022	0.0000	35	3.82	<b>2017J07</b>	0.0251	0.0002	18	0.70
<b>Aitne</b>	0.0023	0.0000	59	7.80	<b>Aoede</b>	0.0276	0.0000	63	7.67
<b>Isonoe</b>	0.0024	0.0001	96	8.99	<b>Pasithee</b>	0.0289	0.0006	30	7.79
<b>Orthosie</b>	0.0025	0.0000	34	8.91	<b>2017J06</b>	0.0300	0.0009	16	0.60
<b>Callirrhoe</b>	0.0026	0.0003	195	8.21	<b>Hegemone</b>	0.0305	0.0002	43	8.41
<b>Sponde</b>	0.0028	0.0000	34	8.08	<b>2003J23</b>	0.0317	0.0012	16	0.04
<b>2010J02</b>	0.0032	0.0003	116	0.66	<b>2017J04</b>	0.0326	0.0004	20	1.66
<b>Dia</b>	0.0032	0.0000	23	8.72	<b>Themisto</b>	0.0379	0.0021	107	119.40
<b>Carpo</b>	0.0036	0.0000	43	12.14	<b>2003J16</b>	0.0399	0.0003	98	5.28
<b>Helike</b>	0.0040	0.0000	46	8.96	<b>2017J08</b>	0.0487	0.0001	13	0.60
<b>2017J09</b>	0.0041	0.0002	15	0.66	<b>Iocaste</b>	0.0530	0.0003	82	10.44
<b>Thelxinoe</b>	0.0047	0.0002	56	9.13	<b>2011J01</b>	0.0781	0.0001	35	3.23
<b>2010J01</b>	0.0050	0.0001	171	7.49	<b>2003J15</b>	0.0848	0.0025	23	7.42
<b>Euanthe</b>	0.0056	0.0000	26	8.95	<b>2003J03</b>	0.1600	0.0001	26	9.53
<b>2016J02</b>	0.0057	0.0000	21	1.56	<b>2003J09</b>	0.1600	0.0035	17	0.07
<b>2003J05</b>	0.0063	0.0002	34	7.23	<b>2003J12</b>	0.8100	0.0018	11	0.06
<b>Harpalyke</b>	0.0069	0.0001	91	10.46	<b>2003J02</b>	0.9000	0.0114	8	0.03
<b>Thyone</b>	0.0070	0.0000	50	8.84	<b>2003J10</b>	0.9100	0.0047	11	0.04
<b>Eukelade</b>	0.0070	0.0001	51	7.46	<b>2003J04</b>	1.0000	0.0008	11	0.05
<b>2018J01</b>	0.0073	0.0001	17	1.79					

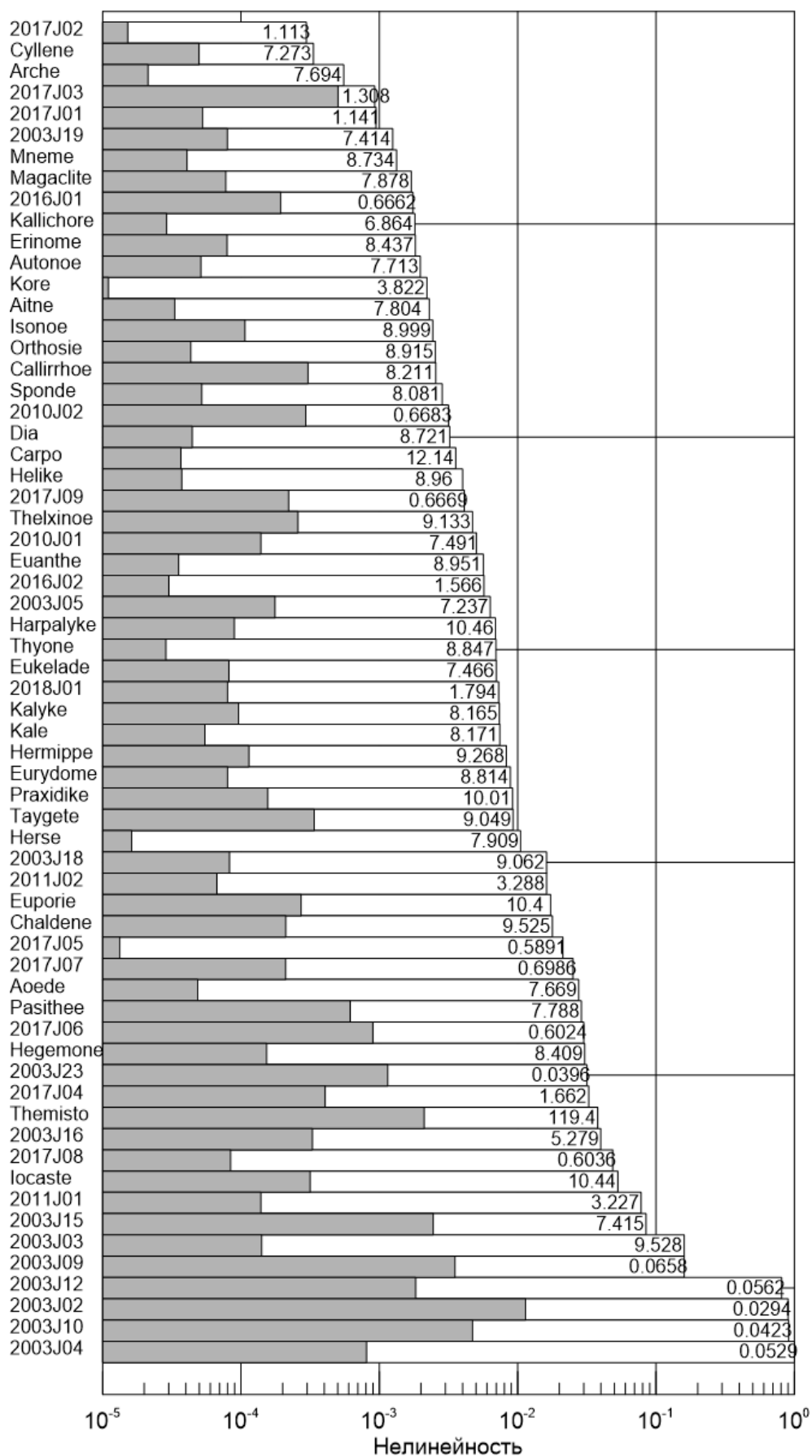


Рисунок 2.4 – Нелинейность для далеких спутников Юпитера

На рисунке 2.4 спутники отсортированы по возрастанию коэффициента полной нелинейности  $\alpha$ . Как видно из Рисунка 2.4 строгой корреляции между нелинейностью и периодом наблюдений не наблюдается. Дело в том, что нелинейность зависит не только от длины дуги, но и от количества наблюдений и их распределения в картинной плоскости и во времени. Т.к. далекие спутники Юпитера – это малые тела Солнечной системы, и не несут угрозы человечеству, поэтому их наблюдают очень редко. За одну наблюдательную ночь получают наблюдательные данные всего на ~2-10 момента времени, а следующие моменты наблюдений возможно получают только через несколько лет. Получается, что у многих спутников окна между наблюдениями порядка нескольких оборотов спутника (если учитывать, что период обращения спутников ~ 1-2 года), таким образом этот факт влияет на улучшение точности определения параметров орбит из наблюдений и на значения показателей нелинейности. Рисунок 2.4 также показывает, что для подавляющего большинства спутников внутренняя нелинейность слабая ( $\alpha < 10^{-3}$ ) (серая гистограмма). Следовательно, для стохастического моделирования неопределенностей в спутниковых орбитах уверенно можно применять нелинейные методы.

### **3 Разработка графического интерфейса JUNO**

JUNO (абб. от англ. JUpiter & Nonlinearity & Observations) – это приложение для вычисления показателей внутренней и полной нелинейности для далеких спутников Юпитера и модельных объектов. Также с помощью JUNO можно визуализировать орбиты спутников относительно Юпитера и строить графики с показателями нелинейности и выводить результаты в таблицу или в файл.

Графический интерфейс JUNO разработан на основе программы, написанной М.А. Баныщиковой на языке Fortan. Для разработки были предоставлены: exe-файл, файл с входными данными, фонд больших планет DE405. Для создания графического интерфейса, с помощью которого можно оценивать показатели нелинейности, использовалась среда Turbo Delphi 2006.

#### **3.1 Этапы разработки**

Создание графического интерфейса JUNO можно разбить на шесть этапов. На первом этапе стояла задача разработать графическую оболочку для вычисления показателей нелинейности для реальных спутников Юпитера (Real). На втором этапе – тоже что и на первом, но для модельных объектов (Model). На третьем – вывод графических и табличных результатов в отдельном окне интерфейса (Result). Четвертый этап заключался в визуализации орбит спутников (Orbits). На пятом этапе представление средних орбитальных элементов спутников Юпитера в виде таблицы (Satellites). На шестом – создание инструкции по работе с JUNO (Help).

Для реализации всех этапов сначала нужно поместить необходимые компоненты на форму.

На форму помещен компонент PageControl, расположенный на вкладке Win32. Этот компонент представляет собой многострочную панель, которая позволяет экономить пространство окна приложения, размещая на одном и

том же месте страницы разного содержания. Чтобы задавать и редактировать страницы этого компонента, необходимо щелкнуть на нем правой кнопкой мыши и выбрать пункт New Page, который отвечает за создание новой страницы. Каждая создаваемая страница является объектом типа TTabSheet. Это панель, на которой можно размещать любые управляющие компоненты, окна редактирования и т.д. Страница имеет основные свойства, которые можно изменять в окне Инспектора Объектов<sup>1</sup>: Caption (надпись, которую будет видно на закладке) и Name (имя, по которому можно ссылаться на страницу). Наиболее значимым свойством у PageControl является ActivePage, которое отвечает за имя активной страницы [27].

Компонент PageControl содержит следующие страницы (Рисунок 3.2):

- Real;
- Model;
- Orbits.

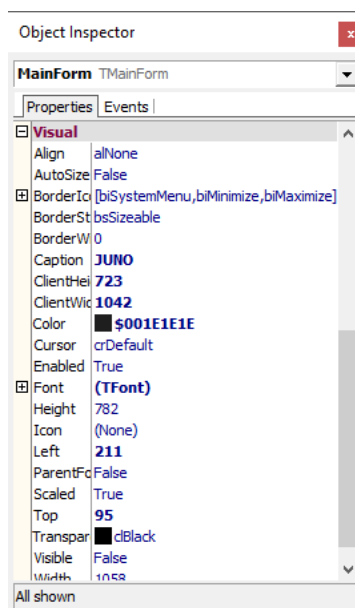


Рисунок 3.1 – Инспектор Объектов

---

<sup>1</sup> Инспектор Объектов (Object Inspector) – основной инструмент (Рисунок 3.1), с помощью которого вы будете задавать свойства компонентов и писать обработчики событий. Окно Инспектора Объектов имеет две страницы. В верхней части окна расположен выпадающий список всех компонентов, помещенных на форме. В нем вы можете выбрать тот компонент, свойства и события которого вас интересуют [27].



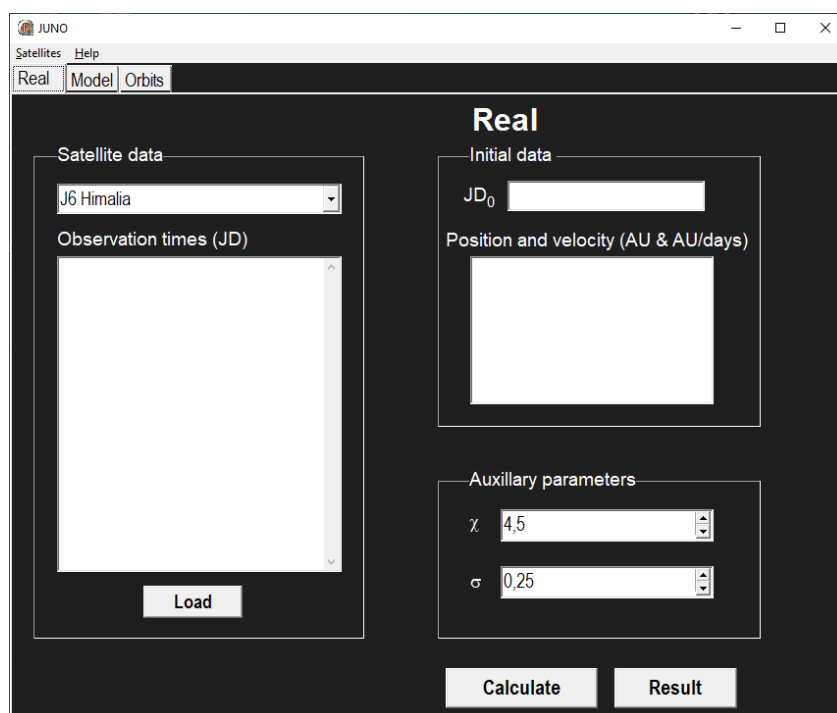


Рисунок 3.2 – Главное окно графического интерфейса (вкладка Real)

Также на форму помещен компонент MainMenu, расположенный на вкладке Standart. Данный компонент не визуальный, т.е. в процессе проектирования место его размещения на форме не представляет никакого значения, в связи с тем, что мы увидим меню, сгенерированное нами, а не сам компонент. Основное свойство данного компонента – Items, которое заполняется с помощью Конструктора Меню, вызываемое двойным щелчком левой кнопкой мыши на MainMenu или нажатием кнопки многоточием рядом со свойством Items в окне Инспектора Объектов<sup>1</sup> [27].

Компонент MainMenu заполнен следующими разделами (Рисунок 3.2):

- Satellites;
- Help.

### 3.1.1 Вкладка Real

На вкладке Real помещены следующие компоненты (Рисунок 3.2): три компонента Bevel (располагается на вкладке Additional и носит лишь декоративный характер [28]. Основное свойство Shape, которое отвечает за

тип фигуры, выбрано bsBox – прямоугольник), десять Label, один Combobox, два Memo, один Edit, два FloatSpinEdit, три Button.

У десяти компонентов Label в свойство Caption заносится информация, которая будет отображаться на форме, т.е. Real, Satellite data, Observation times (JD), Initial data, JD, 0, Position and velocity (AU & AU/days), Auxiliary parameters,  $\chi$  и  $\sigma$ . У всех компонентов, кроме Real, 0,  $\chi$  и  $\sigma$  в свойстве Font устанавливаем параметры выводимого текста, т.е. цвет, размер и стиль шрифта (стиль шрифта – Arial, цвет – clWhite (белый), размер – 14). Отличительной особенностью Label Real и 0 является размер шрифта, который устанавливается 24 и 11, а у Label  $\chi$  и  $\sigma$  – стиль шрифта Symbol.

ComboBox нужен для выбора далеких спутников Юпитера из списка. Отличительным свойством является Style, который при значении csDropDownList запрещает вводить информацию в строке ввода, а может принимать значения только одной из строк, сохранённых в компоненте. В свойство Items заносятся 71 строк, которые соответствуют названию всех внешних спутников Юпитера, по умолчанию стоит J6 Himalia, который соответствует индексу 0, заносимый в свойство ItemIndex.

Два компонента Memo для ввода многострочной информации, в один вводятся даты наблюдений либо вручную, либо с помощью кнопки Load (для удобства просматриваемой информации в обработчике событий OnCreate главной формы у Memo в свойстве ScrollBars установлено ssVertical, которое представляет собой вертикальную полосу прокрутки), в другой – начальный вектор положения и скорости.

Edit необходим для ввода начальной даты.

FloatSpinEdit является сторонним компонентом [29] и представляет собой аналог компонента SpinEdit, но имеет возможность работать с вещественными числами. Для того, чтобы этот компонент отображался в среде Turbo Delphi 2006: необходимо создать пакет и загрузить его со всеми данными в директорию библиотеки Delphi. Один компонент отвечает за

данные для  $\chi$  и в основных свойствах, отвечающих за шаг, максимальное, минимальное и значение по умолчанию вводятся следующие данные: Increment = 10, MaxValue = 10, MinValue = 0.1, Value = 4.5. Другой – за данные для  $\sigma$ : Increment = 0.01, MaxValue = 2, MinValue = 0.01, Value = 0.25.

У трех кнопок в свойстве Caption заносится информация, отображающееся на форме, т.е. Load, Calculate и Result. В свойстве Font устанавливаем параметры выводимого текста, т.е. цвет, размер и стиль шрифта (стиль шрифта – Arial, цвет – clblack (черный), размер – 16).

Для написания обработчика события OnClick кнопки Load помещен на форму компонент OpenFileDialog,<sup>2</sup> расположенный на вкладке Dialogs. После выполнения метода Execute, получаем полный адрес выбранного файла. Затем информация, содержащаяся в файле, записывается в компонент Memo, отвечающий за даты наблюдений.

В обработчике события OnClick кнопки Calculate записываются данные в два текстовых файла и запускается сторонний exe-файл, написанный на языке Fortran, который производит вычисление показателей нелинейности. В первый файл «sat4kate.in» записываются данные  $\chi$  и  $\sigma$  (компонент FloatSpinEdit), название выбранного спутника (компонент ComboBox), но так как это название в дальнейшем будет использоваться в результирующем файле с показателями нелинейности, необходимо убрать лишние символы «/» и пробел. Для этого используется процедура Insert, которая добавляет символ «\_» в нужное место строки, а процедура Delete удаляет символы, от которых нам необходимо было избавиться. Затем записывается название спутника и расширение файла «.dat», а также начальный вектор положения и скорости (компонент Memo). Во второй файл «название спутника.dat» записываются даты наблюдений (компонент Memo). Далее необходимо запустить exe-файл «koefNelin\_model.exe» и дождаться его завершения по следующему

---

<sup>2</sup> Компонент OpenFileDialog предназначен для открытия файлов.

алгоритму: определяем путь к exe, заполняем нулями параметры будущего процесса, после этого выставляем в нем некоторые параметры: cb – содержит количество байтов, занимаемых структурой TStartupInfo (инициализируем его как SizeOf(TStartupInfo)), dwFlags – содержит набор флагов, позволяющих управлять созданием дочернего процесса (показываем окно, курсор – часики), wShowWindow – определяет как должно выглядеть окно запущенного приложения (прячем окно), запускаем сам exe с параметрами с помощью функции CreateProcess, ждем завершения инициализации и процесса, получаем код завершения, закрываем дескрипторы процесса и потока. Также следует заметить, что чем больше данных введено в Мемо с датами наблюдениями, тем больше времени занимать работа exe-файла «koefNelin\_model.exe».

Код реализации запуска exe и его завершения представлен ниже:

```

...
1. AppToRun := 'koefNelin_model.exe'; // определяем путь к exe
2. FillChar(StartupInfo, SizeOf(TStartupInfo), 0);
3. with StartupInfo do
4. begin
5.   cb := SizeOf(TStartupInfo);
6.   dwFlags:=STARTF_USESHOWWINDOW or STARTF_FORCEONFEEDBACK;
7.   wShowWindow := SW_HIDE
8. end;
9. CreateProcess(PChar(AppToRun),"nil, nil, false, CREATE_NEW_CONSOLE or
   NORMAL_PRIORITY_CLASS, nil, nil, StartupInfo, ProcessInfo);
10. with ProcessInfo do
11. begin
12.   WaitForInputIdle(hProcess, INFINITE);
13. WaitForSingleObject(ProcessInfo.hProcess, INFINITE);
14. GetExitCodeProcess(ProcessInfo.hProcess, EXITCODE);
15. CloseHandle(hThread);
16. CloseHandle(hProcess);
17. end;
...

```

После выполнения этого процесса формируется файл «KOEFNELIN.dat», с которого считываются все данные:  $\kappa_{ppaI}$ ,  $\kappa_{ppa}$ ,  $d\kappa_{Riv}$ ,  $\sigma_{sat}$ ,  $\sigma_{sat1}$ ,  $\sigma_{pan}$ ,  $L$ ,  $Name_{Sat}$ , но нам необходимы только  $\kappa_{ppaI}$  и  $\kappa_{ppa}$  – внутренняя и полная нелинейность,  $L$  – количество наблюдений,  $\sigma_{pan}$  – доли

дуги орбит, покрывающие наблюдения и Name\_Sat – название спутника. Так как в название спутника мы вносили изменения, необходимо с помощью тех же процедур Insert и Delete преобразовать его в первоначальный вид. У компонента StringGrid<sup>3</sup>, который назван в программе Table\_StringGrid в свойстве RowCount (отвечает за количество строк в таблице) стоит 1, затем мы прибавляем еще одну строку для того, чтобы выводить в нее данные из файла, и в ячейки с номерами, в которых меняется только номер столбца, заносим эти данные. Так как ячейки представляют собой строки, вводимую информацию из чисел необходимо перевести в строковый тип с помощью процедур FloatToStr (для вещественных чисел) и IntToStr (для целых чисел).

Ниже представлен код реализации заполнения StringGrid:

```
...
1. with AdditionalForm.Table_StringGrid do
2. begin
3.   RowCount:=RowCount+1;
4.   cells[0,RowCount-1]:=Name_Sat;
5.   cells[1,RowCount-1]:=FloatToStr(kappa);
6.   cells[2,RowCount-1]:=FloatToStr(kappaI);
7.   cells[3,RowCount-1]:= IntToStr(L);
8.   cells[4,RowCount-1]:=FloatToStr(span);
9. end;
...
```

Далее необходимо отобразить показатели нелинейности на графике, которые будут представлены в виде прямоугольников (гистограмм). Гистограммы будут изменять размер в зависимости от того, сколько спутников было выбрано, т.е. с увеличением количества спутников, уменьшаются прямоугольники. Так как все данные берутся из таблицы Table\_StringGrid, то необходимо проверять условие о количестве строк таблицы, если оно превышает значение 2, то компонент Image, названный в программе Image1 необходимо отчистить. У данного компонента есть свойство Picture, которое содержит изображение. Данному свойству присваивается значения nil, т.е. нулевое значение. После этого на нем

---

<sup>3</sup> Компонент StringGrid используется для создания в приложении таблицы, которая может содержать в себя текстовую и графическую информацию [27].

необходимо построить ось  $x$ , которая представляет собой логарифмическую шкалу с помощью процедуры `axisX (Image:TImage)`, рисуем горизонтальную линию с помощью процедур `MoveTo` (устанавливает курсор, откуда будет происходить рисование) и `LineTo` (рисует линию), начиная с  $x=172$  пикселей (данные пиксели отводятся на текст, соответствующий названию спутника) до конца `Image1`, через каждые 100 пикселей с помощью этих процедур рисуем вертикальные большие штрихи на оси  $x$ , которые будут соответствовать значениям показателей нелинейности от  $10^{-7}$  до  $10^1$ , сами значения записываются с использованием процедуры `TextOut`. Для добавления делений между значения используется принцип построения логарифмического масштаба<sup>4</sup>. С помощью процедуры `axisten (Image: TImage; i: integer; count: integer)` строим ось  $y$  и вертикальные линии, проходящие через диапазон значений показателей нелинейности. `Count` содержит информацию о количестве спутников, и в зависимости от этого значения определяется ширина прямоугольников. Также с помощью процедур `MoveTo` и `LineTo` через каждые 100 пикселей строим вертикальные линии, которые на 5 пикселей будут выше построенных гистограмм и над ними строим горизонтальную линию. Затем строим сами прямоугольники с помощью процедуры `RectangleBuilding (Image: TImage; kappa: extended; Name_Sat: string; span: extended; color: TColor; i: integer; count: integer)` (`kappa` – значения показателя нелинейности, `Name_Sat` – название спутника, `span` – доли дуг, покрывающие наблюдения, `color` – цвет прямоугольника, `i` – какой спутник и `count` – сколько всего выбрано спутников). От количества спутников также изменяется ширина прямоугольников и размер шрифта. Процедура `TextOut` позволяет записать название спутника на `Image1`, затем в зависимости от значения показателей нелинейности в переменную `border` заносится округленное значение, умноженное на 10 в степени, которая позволит получиться целочисленное

---

<sup>4</sup> Логарифмический масштаб (шкала) — шкала, длина отрезка которой пропорциональна логарифму отношения величин, отмеченных на концах этого отрезка, в то время как на шкале в линейном масштабе длина отрезка пропорциональна разности величин на его концах [30].

значение, по принципу построения логарифмического масштаба определяем правую границу прямоугольника и присваиваем border это значение. Цвет окантовки прямоугольника устанавливаем черный, а цвет самого прямоугольника зависит от того, какую нелинейность мы рассматриваем (полная нелинейность – белый, внутренняя нелинейность – серый). С помощью процедуры Rectangle строим сами прямоугольники. Используя процедуру TextOut записываем значение span в правой части белого прямоугольника.

Ниже представлен код реализации построения графика на Image1:

```
...
1. with AdditionalForm.Table_StringGrid do
2. begin
3. I :=RowCount-1;
4. if RowCount>2 then
5. begin
6. AdditionalForm.Image1.Picture:=nil; //очистка Image
7. axisX(AdditionalForm.Image1);
8. end;
9. for j := 1 to i do
10. axisten(AdditionalForm.Image1,j,i);
11. for j := 1 to i do
12. begin
13. RectangleBuilding( AdditionalForm.Image1, StrToFloat(cells[1,j]), cells[0,j],
    StrToFloat(cells[4,j]),clwhite,j,i);
14. RectangleBuilding( AdditionalForm.Image1, StrToFloat(cells[2,j]), cells[0,j],
    StrToFloat(cells[4,j]), clsilver,j,i);
15. end;

```

...  
После выполнения всех процедур отчищаем два Мемо и Edit, используя метод Clear.

В обработчике события OnClick кнопки Result вызывается как немодальное окно с помощью метода Show форма с результатами в виде графика и таблицы.

### 3.1.2 Вкладка Model

На вкладке Model помещены следующие компоненты (Рисунок 3.3): четыре компонента Bevel, двадцать два Label, два Мемо, восемь Edit, два

FloatSpinEdit, четыре Button.

Данная вкладка по своему функционалу похожа на вкладку Real (см. подраздел 3.1.1), отличительные особенности: название спутника вводится вручную в компонент Edit, есть возможность вводить элементы орбиты в шести Edit (для этого используется сторонняя процедура, позволяющая вводить данные с разделителем «.») для вычисления начального вектора положения и скорости с помощью задачи двух тел, а также добавлена кнопка Clear, которая очищает все данные с помощью метода Clear у всех Edit и Memo.

The screenshot shows the 'JUNO' application window with the 'Model' tab active. The interface includes a menu bar with 'Satellites' and 'Help'. The main area is divided into three panels: 'Satellite data' on the left with a name input field, observation times text area, and a 'Load' button; 'Initial data' in the center with 'JD0' input, position/velocity text area, and auxiliary parameters (chi = 4.5, sigma = 0.25); and 'Elements' on the right with six orbital parameter inputs (a, e, i, omega, Omega, M) with units. Action buttons 'Calculate', 'Result', and 'Clear' are at the bottom right.

Рисунок 3.3 – Вкладка Model

У Edit, отвечающего за название спутника, в свойстве Text прописываем «Name Satellite» и устанавливаем цвет шрифта серым. Для того, чтобы ввести данные в черном цвете и убрать изначальную надпись, обработчике событий



OnKeyDown проверяем условие: если цвет шрифта серый, то заменяем его на черный и в свойствоText заносим пустую строку.

Код реализации отчистки Edit и ввод данных черным цветом:

- ```
...
1. with (Sender as TEdit) do
2. if Font.Color = clSilver then
3. begin
4. Font.Color:= clBlack;
5. Text:= "";
6. end;
```

Для того, чтобы кнопка Calculate была неактивной, пока не введены элементы орбиты или начальной вектор положения и скорости, на форме помещен компонент ActionList<sup>5</sup>, который располагается на вкладке Standart. Добавляем в ActionList новое действие и связываем его с кнопкой Calculate с помощью свойства Action. В обработчике событий кнопки Action прописываем, что активация возможна только в том случае, если нужные нам компоненты заполнены.

Ниже представлен код, реализующий активацию кнопки Calculate:

- ```
...
1. TAction(Sender).Enabled:= (memo2_sat4Kate.Lines.Text <>"") or (Edit2.Text <>
") and (Edit3.Text <> ") and (Edit4.Text <> ") and (Edit5.Text <> ") and (Edit6.Text
<> ") and (Edit7.Text <> ");
...
```

### 3.1.3 Окно Result

На форму помещены следующие компоненты (Рисунок 3.4): два Bevel, один StringGrid, три Image, семь Label и три Button.

У StringGrid в свойстве ColCount установлено значение 5, что соответствует количеству столбцов таблицы. Для того, чтобы отобразить в этих столбцах информацию, на форму помещен компонент ImageList<sup>6</sup>, так как Delphi не имеет возможности записывать некоторые символы. Чтобы в

---

<sup>5</sup> ActionList заносит тот список действий, которые могут быть компонентами и элементы управления [27].

<sup>6</sup> ImageList представляет собой набор изображений одинаковых размеров, на которые можно ссылаться по индексам, начиная с 0 [24].

Table\_StringGrid отобразить не только текстовую информацию, но и графическую нужно произвести рисование на канве. В обработчике событий OnDrawCell данного компонента в нужной ячейке отображением картинку из ImageList, также центрируем данные таблицы и делаем разделители таблиц черные.

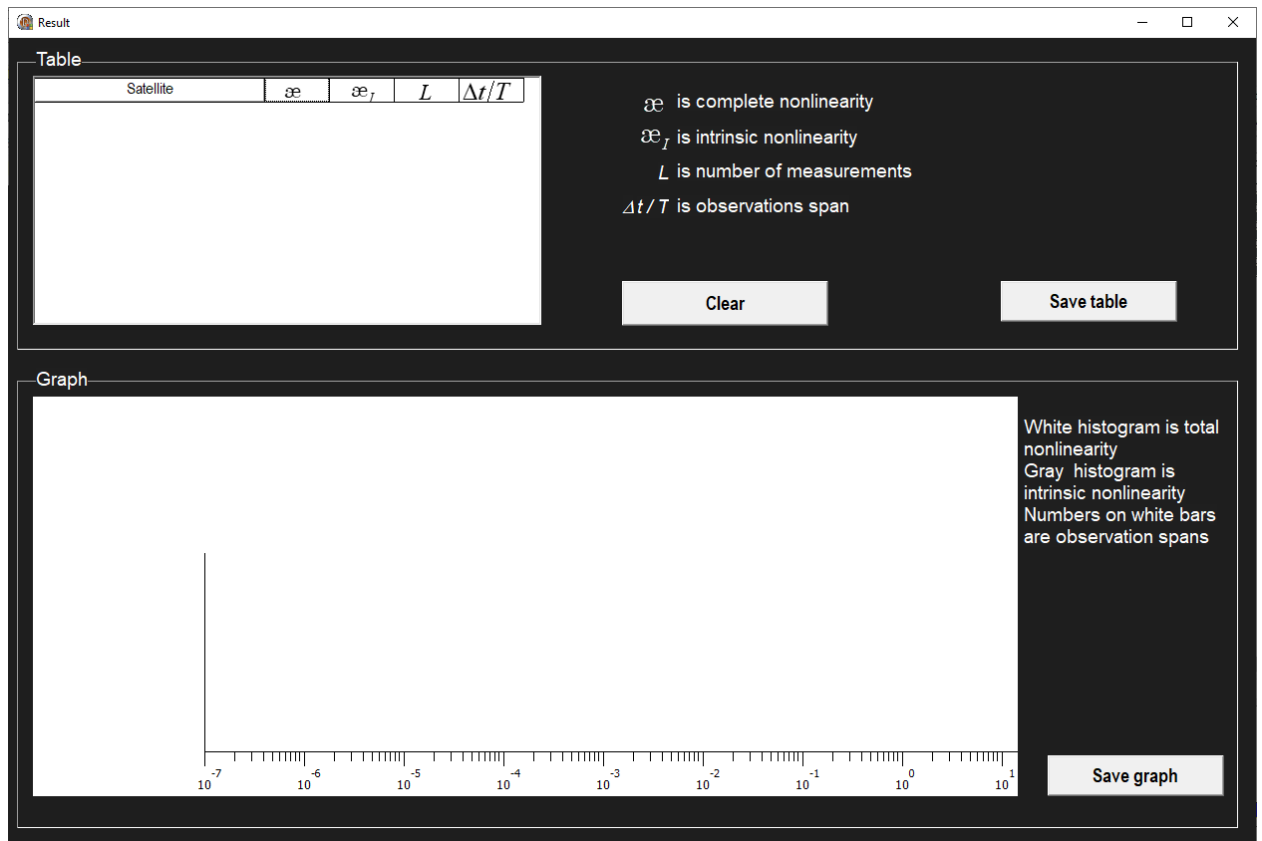


Рисунок 3.4 – Окно Result

Ниже представлен код, реализующий разделители таблицы и ее центрирование:

```

...
1. with Sender as TStringGrid do
2. begin
3.   Canvas.Pen.Color := clBlack;
4.   Canvas.MoveTo(0,Rect.Top);
5.   Canvas.LineTo(Rect.Right,Rect.Top);
6.   if ARow=RowCount-1 then
7.   begin
8.     Canvas.MoveTo(0,Rect.Bottom);
9.     Canvas.LineTo(Rect.Right,Rect.Bottom);
10.  end;
11. Canvas.MoveTo(Rect.Right,Rect.Top);
12. Canvas.LineTo(Rect.Right,Rect.Bottom);

```

```

13. if ACOL=0 then
14. begin
15. Canvas.MoveTo(Rect.Left,Rect.Top);
16. Canvas.LineTo(Rect.Left,Rect.Bottom);
17. end;
18. end;
19. with Table_StringGrid.Canvas do
20. begin
21. cr:=Table_StringGrid.CellRect(acol,arow) ;
22. FillRect(cr);
23. TextOut(cr.left+((cr.Right-cr.Left) div 2)-
    (TextWidth(Table_StringGrid.Cells[acol,arow]) div 2),
    cr.Top,Table_StringGrid.Cells[acol,arow]);
24. end;
...

```

Семь Label нужны для отображения пояснительной информации на форме, которая записывается в свойстве Caption в обработчике событий формы OnCreate. У Font устанавливаем параметры выводимого текста: стиль шрифта – Arial, цвет – clWhite (белый), размер – 14.

Два Image имеют небольшой размер и в них загружены картинки с использованием метода LoadFromFile с обозначениями æ и æ<sub>i</sub> (в Delphi нет возможности вывести символ æ), еще один Image необходим для отображения графика.

В обработчике событий OnClick кнопки Clear прописываем отчистку Table\_StringGrid с помощью метода Clear у Rows и устанавливаем количество строк равное 1. Image1 отчищаем с использованием метода Picture, которому присваиваем значение nil и рисуем с помощью процедуры axisX ось x.

Для реализации сохранения графика на форму помещен компонент SavePictureDialog<sup>7</sup>, расположенный на вкладке Dialogs. В обработчике событий OnClick кнопки SaveGraph после выполнения метода Execute, присваиваем строковой переменной, которая отвечает за название файла SavePictureDialog1.FileName, далее преобразовываем картинку из формата bmp в jpg. После этих действия обязательно необходимо отчистить созданные переменные bmp и jpg.

---

<sup>7</sup> Компонент SavePictureDialog предназначен для сохранения графических файлов.

Ниже представлен код реализации сохранения графика:

```
...
1. if SavePictureDialog1.Execute then
2. begin
3. SavePictureDialog1.Filter:='Bitmap      Map      (*.bmp)|*.bmp|JPEG      Image
   (*.jpg)|*.jpg';
4. FileName:=SavePictureDialog1.FileName;
5. jpg:=TJPEGImage.Create;
6. bmp:=TBitmap.Create;
7. bmp.Assign(Image1.Picture.Bitmap);
8. jpg.Assign(bmp);
9. jpg.SaveToFile(FileName);
10. jpg.Free;
11. bmp.Free;
12. Image1.Picture.SaveToFile(Filename);
13. end;
...
```

Сохранение таблицы Table\_StringGrid происходит аналогичным образом с использованием метода Execute и строковой переменной, только для этого используется компонент SaveDialog<sup>8</sup>, расположенный также на вкладке Dialogs. У SaveDialog есть метод Filter, позволяющий отображать и выбирать только некоторые типы файлов и метод DefaultExt, отвечающий за расширение, которое будет добавляться к имени файлу. В файл записываем строку с данными, которая соответствует шапке нашей таблицы, т.е. Sattelite, Карпа, KappaI, L, delta t/T, а затем все данные из таблицы.

Ниже представлен код, реализующий сохранение таблицы в текстовый файл:

```
...
1. SaveDialog_Table.Filter := 'Text file|*.txt';
2. SaveDialog_Table.DefaultExt := 'txt';
3. if SaveDialog_Table.Execute then
4. assignfile(f, SaveDialog_Table.FileName);
5. rewrite(f);
6. setlength(a, Table_StringGrid.ColCount);
7. for i := 0 to Table_StringGrid.ColCount - 1 do
8. begin
9. a[i] := 0;
10. for j := 0 to Table_StringGrid.rowcount - 1 do
```

---

<sup>8</sup> Компонент SaveDialog предназначен для сохранения любых файлов.

```

11. if length(Table_StringGrid.Cells[i, j]) > a[i] then
12. a[i] := length(Table_StringGrid.Cells[i, j]);
13. end;
14. write(f, 'Sattelite', ' ', 'Kappa', ' ', 'KappaI', ' ', 'L', ' ', 'delta t/T');
15. for i := 1 to Table_StringGrid.rowcount - 1 do
16. begin
17. writeln(f);
18. for j := 0 to Table_StringGrid.colcount - 1 do
19. write(f, Table_StringGrid.Cells[j, i]: a[j], ' ');
20. end;
21. closefile(f);
22. setlength(a, 0);
...

```

### 3.1.4 Вкладка Orbit

На вкладке Orbits помещены следующие компоненты (Рисунок 3.5): один Bevel, двенадцать Label, семь Edit, и три Button.

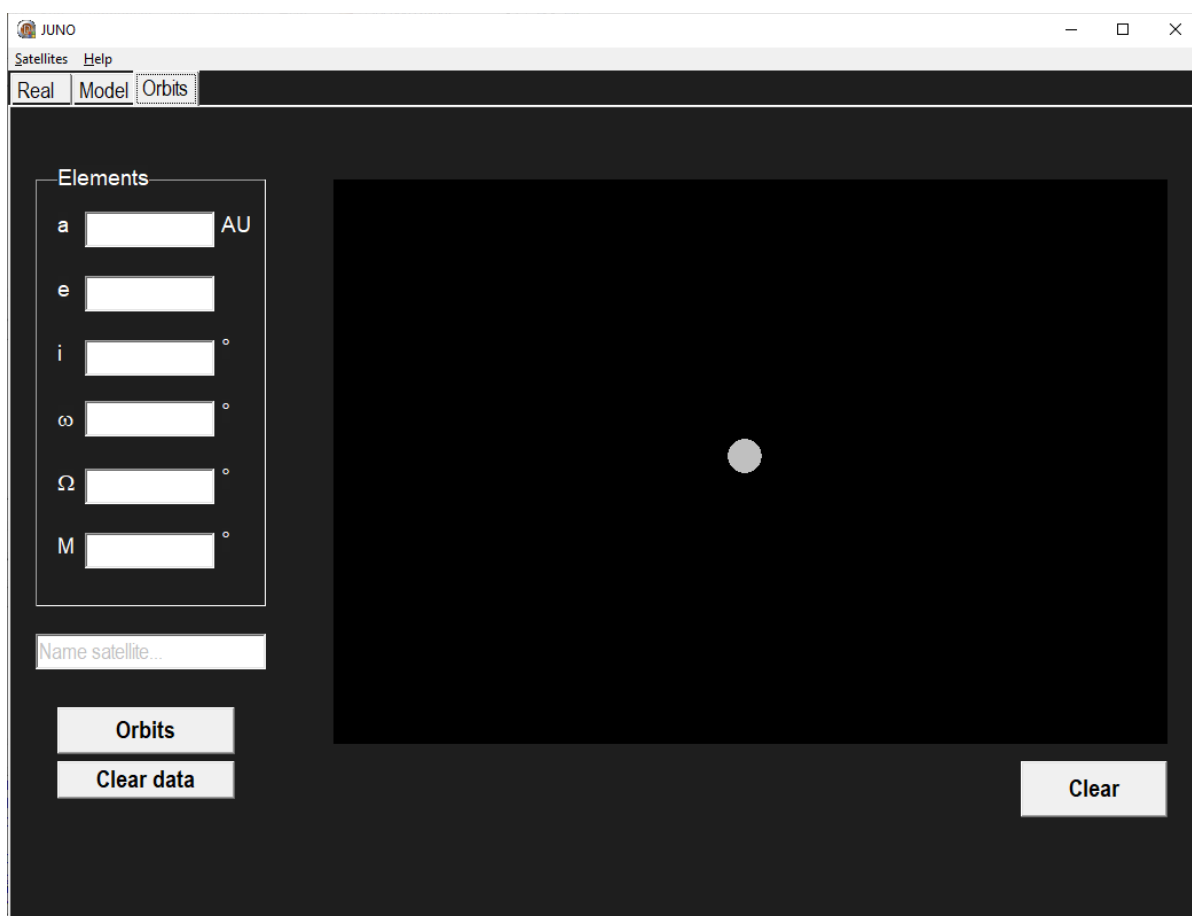


Рисунок 3.5 – Вкладка Orbits

В шесть Edit записываются элементы орбиты и в еще один название спутника. У Edit, отвечающего за название спутника в свойстве Text

прописываем «Name Satellite» и устанавливаем цвет шрифта серым. Для того, чтобы ввести данные в черном цвете и убрать изначальную надпись, в обработчике событий OnKeyDown проверяем условие: если цвет шрифта серый, то заменяем его на черный и в Text заносим пустую строку.

В обработчике событий главной формы OnCreate рисуем серый эллипс в центре Orbits\_Image, который будет представлять собой Юпитер, используя стандартную процедуру построения эллипса.

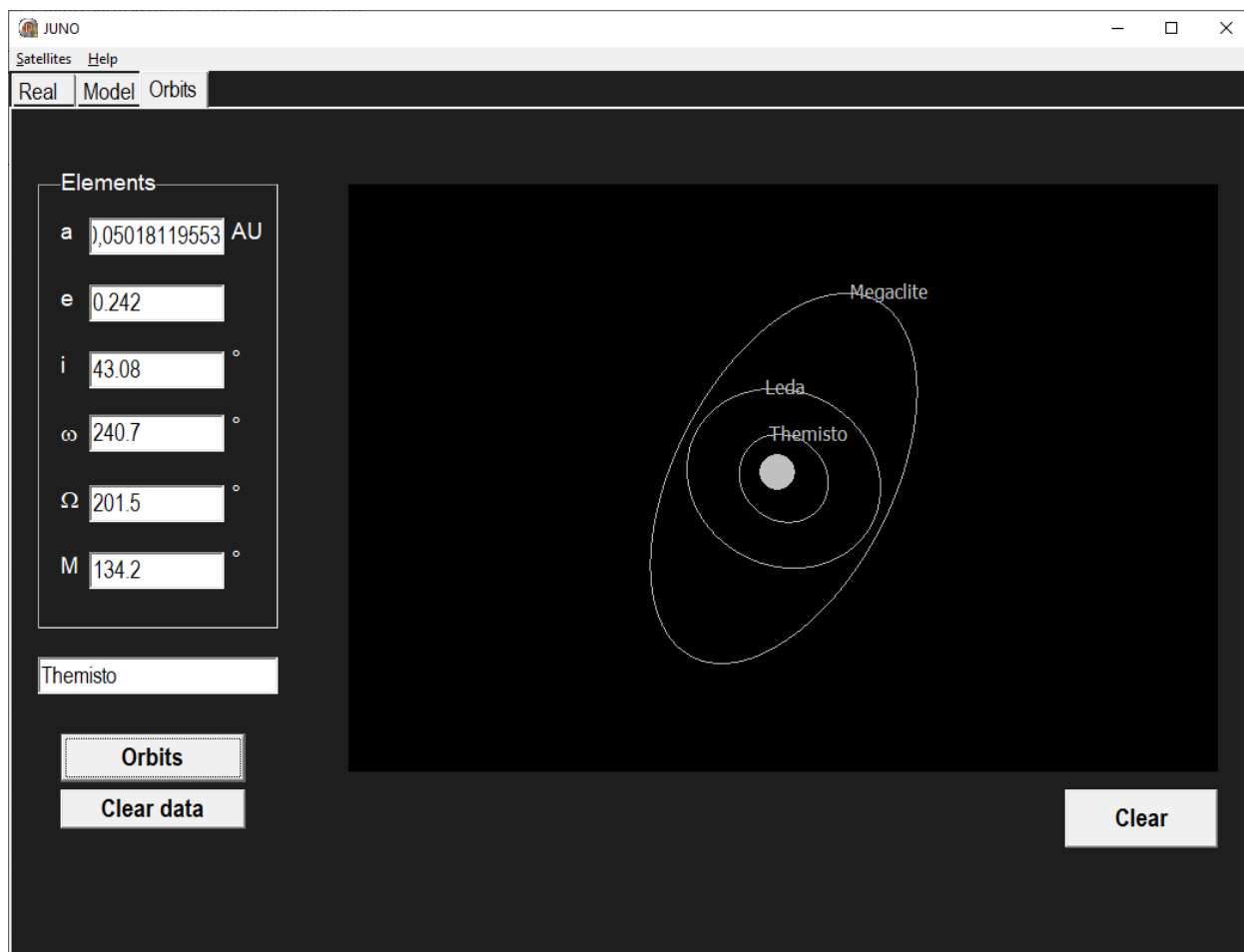


Рисунок 3.6 – Вкладка Orbits после нажатия на кнопку Orbits

В обработчике событий OnClick кнопки Orbits прописываем решение задачи двух тел, которую решаем 1000 раз с изменением времени, зависящим от периода. У полученных координат находим максимальное значение по x и y. Это необходимо сделать для того, чтобы найти максимальную точку на большой и малой полуосях, потом с помощью процедуры Ellipse (Canvas: TCanvas; name: string; X, Y, A, B: integer; Angle: single), где name – название

спутника, X, Y – координаты центра, A – большая полуось, B – малая полуось и Angle – угол поворота эллипса, строим эллипс на Orbits\_Image по точкам, цвет карандаша устанавливаем серым, а цвет кисти – невидимым (Рисунок 3.6).

### 3.1.5 Вкладка Satellites

Для вкладки Satellites создаётся новая форма, которая вызывается в OnClick, как немодальное окно, на нее помещаются следующие компоненты: один Bevel, три Label, один StringGrid и один Image (Рисунок 3.7).

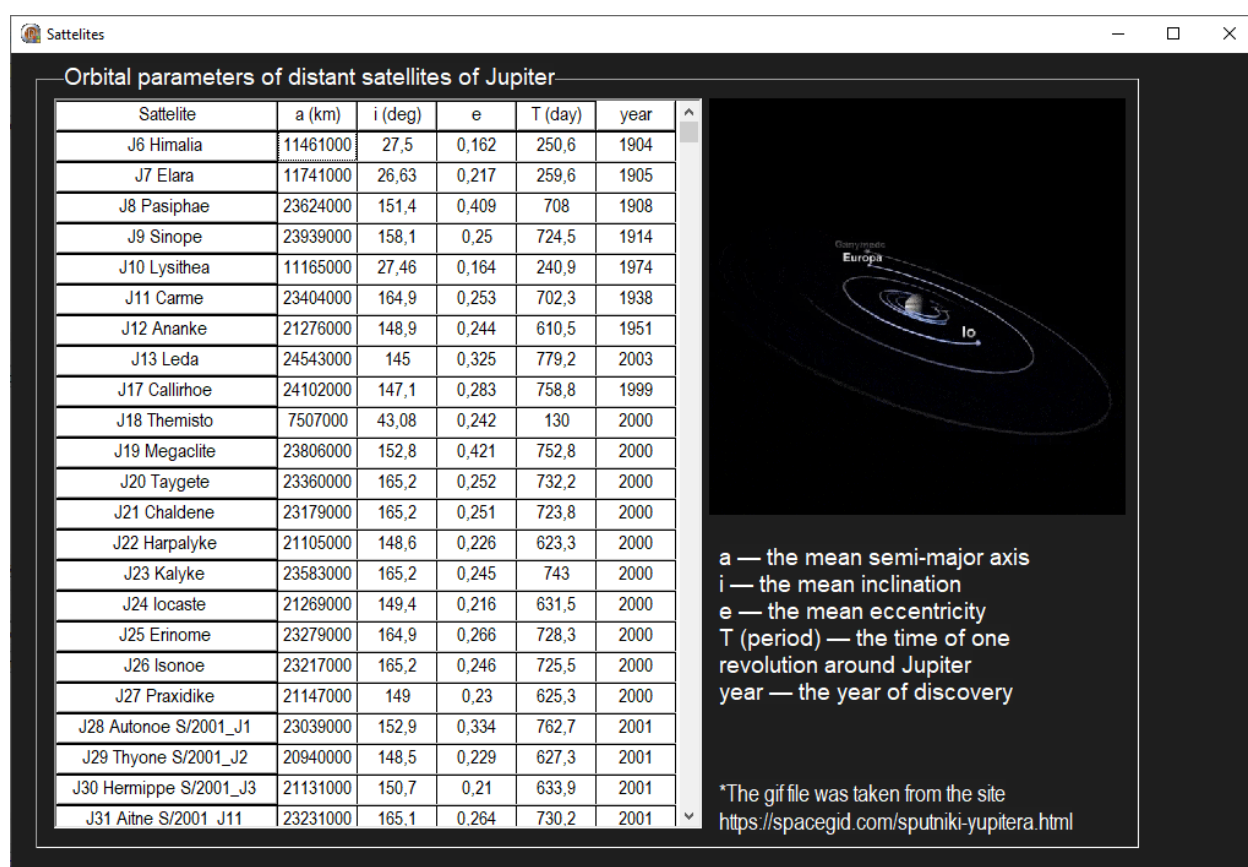


Рисунок 3.7 – Satellites

Два Label необходимы для отображения пояснительной информации, еще один для замечания, в котором написано откуда взят gif-файл.

StringGrid содержит средние орбитальные элементы далеких спутников Юпитера [12]. Все данные изначально были записаны в текстовый файл, затем в обработчике событий OnCreate формы они считываются в соответствующую

ячейку таблицы. В обработчике событий OnDrawCell данного компонента также центрируем данные таблицы и делаем разделители таблиц черные.

Image отображает giv-файл, который визуализирует орбиты всех спутников Юпитера [31]. Для того, чтобы иметь возможность отображать giv-файл, в модуль формы подключается сторонняя библиотека GIFImage. В обработчике событий OnCreate формы методу Proportional присваиваем значение true для того, чтобы автоматически масштабировать загружаемое изображение без искажений, чтобы это выполнилось необходимо методу AutoSize присвоить значение false. Затем с помощью метода LoadFromFile загружаем наш giv.

Ниже представлен код реализации загрузки giv в Image:

```
...
1. Image1.AutoSize:=False;
2. Image1.Proportional:=True;
3. Image1.Picture.LoadFromFile('sattelite.gif'); //загрузка gif
...
```

### **3.1.6 Вкладка Help**

Help включает в себя разделы (Рисунок 3.8):

- Help
- About



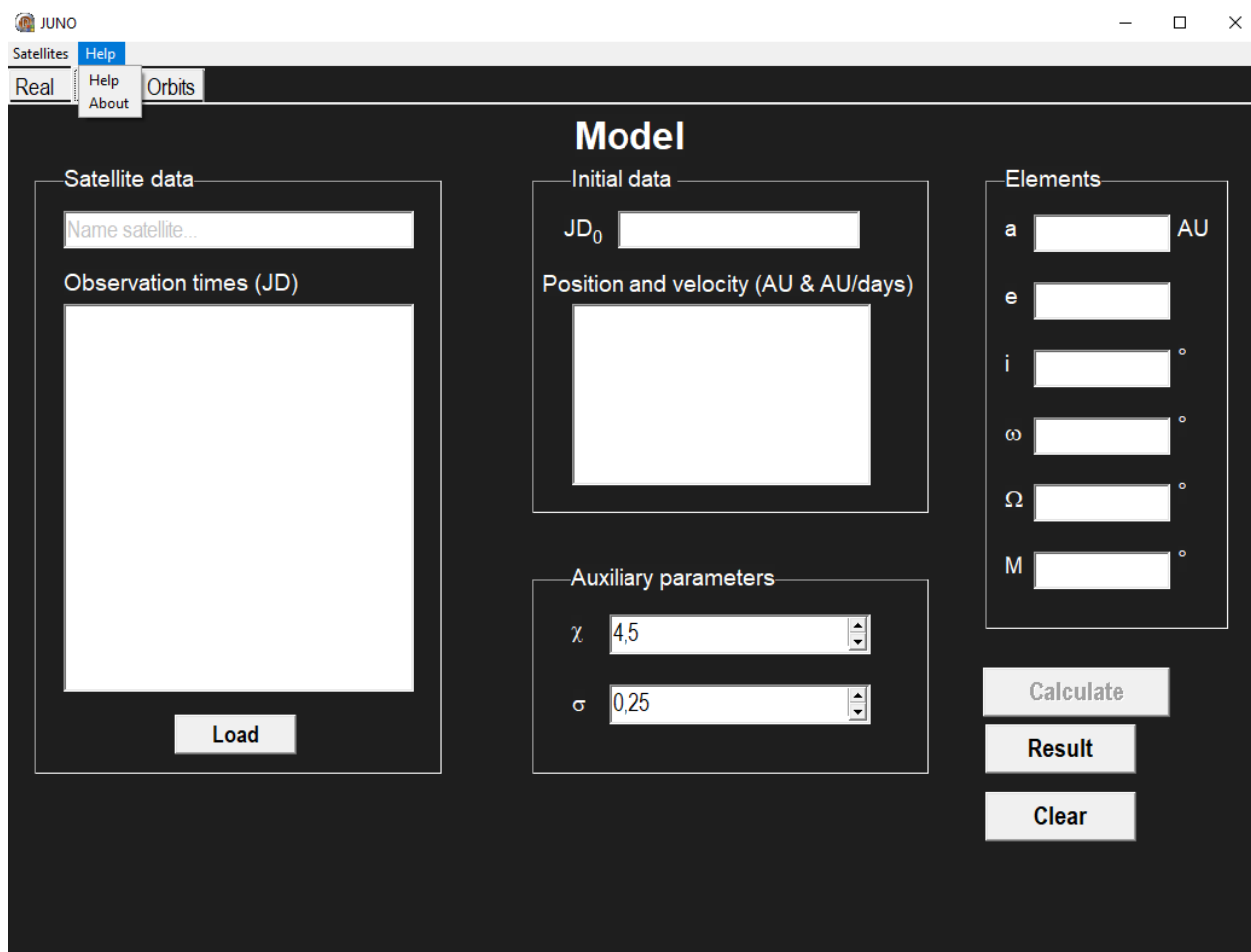


Рисунок 3.8 – Help

В разделе Help содержится информация, поясняющая все действия, производимые в JUNO (Рисунок 3.9). Создается форма, которая вызывается как немодальное окно в обработчике событий OnClick. На форму помещаются два Bevel и десять Label. В свойстве Caption Label заносится необходимая информация. Help представляет собой небольшую пошаговую инструкцию заполнения данных для реальных и модельных объектов. У Label с названиями Real, Model, Calculate, Result и Satellites в свойстве Font начертание шрифта устанавливаем жирным для того, чтобы акцентировать на этом внимание.

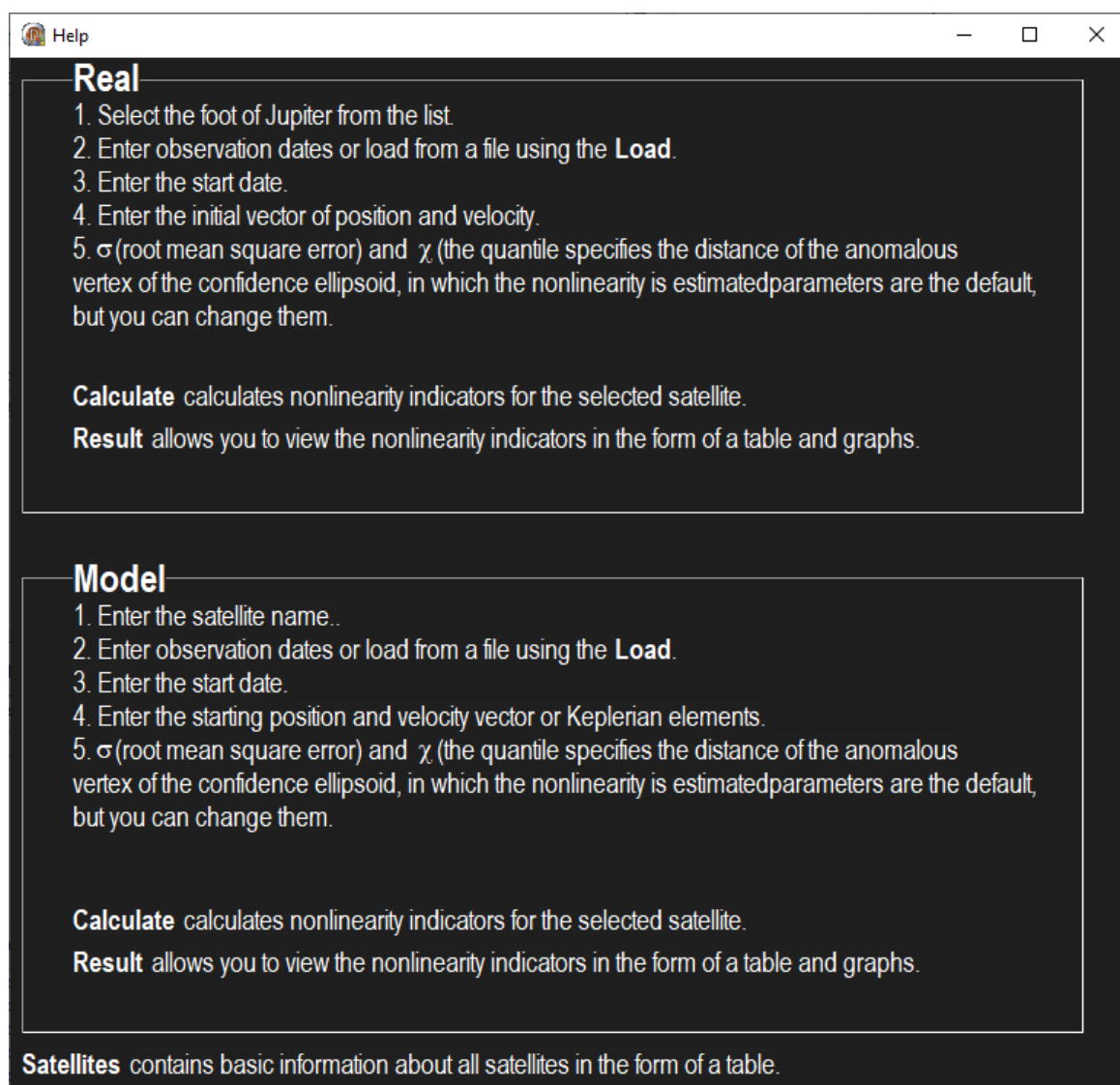


Рисунок 3.9 – Раздел Help

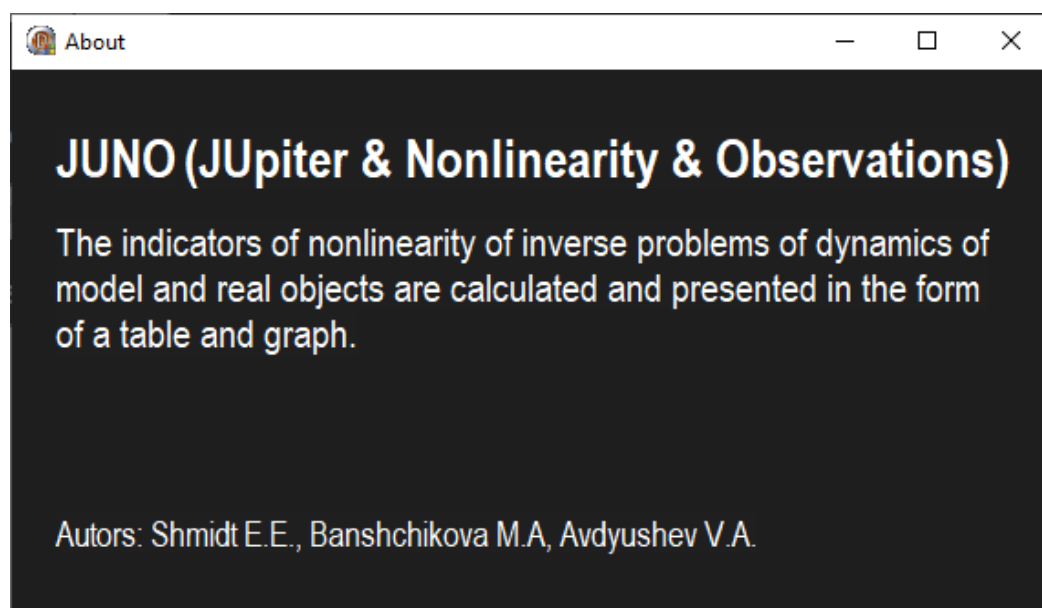


Рисунок 3.10 – Раздел About

Раздел About содержит сведения о программе (Рисунок 3.10). Также создается новая форма, которая вызывается в OnClick, как немодальное окно. На форме располагаются компоненты три Label. Отличием всех надписей являются параметры, установленные в свойстве Font: для названия программы и расшифровки аббревиатуры: стиль шрифта – Arial, цвет – clWhite (белый), размер – 22, начертание – полу утолщенный, полужирный; для краткого описания программы: размер – 16, начертание – полу утолщенный; для имени авторов работы: размер – 14.

### **3.2 Описание интерфейса**

Для вычисления показателей нелинейности реальных объектов необходимо на вкладке Real сначала заполнить блок с данными спутника (Satellite data): выбрать спутник из списка, ввести даты наблюдений или загрузить их из файла с помощью кнопки Load; затем – блок с исходными данными (Initial data): ввести начальную дату и начальный вектор положения и скорости; данные в блоке вспомогательных параметров (Auxiliary parameters) стоят по умолчанию, при необходимости их можно изменить с помощью кнопок вверх/вниз или вводом числа. Когда все данные будут заполнены, нужно нажать на кнопку Calculate для вычисления показателей нелинейности, после выполнения расчетов все поля с данными будут очищены. Увидеть результат можно в отдельном окне интерфейса, нажав на кнопку Result. В окне Result можно сохранить таблицу и график с показателями нелинейности, воспользовавшись кнопками Save table и Save graph. Для очистки всех данных с графика и таблицы используется кнопка Clear.

Аналогичным образом происходит вычисление показателей нелинейности для модельных объектов на вкладке Model, отличительными особенностями являются ввод имени спутника, а не его выбор в блоке с данными спутника (Satellite data), а также если неизвестен начальный вектор положения и скорости, его можно оставить пустым и заполнить блок с

элементами (Elements), в этом случае после нажатия на кнопку Calculate рассчитаются не только показатели нелинейности, но и начальный вектор положения и скорости. Точно также вычисляются элементы орбиты, если введен начальный вектор положения и скорости. Для того, чтобы отчистить все поля нужно нажать на кнопку Clear.

Визуализация орбит спутников располагается на вкладке Orbits. Для построения орбиты необходимо заполнить блок с элементами (Elements), ввести название спутника и нажать на кнопку Orbits. Отчистить поля с данными можно с помощью кнопки Clear data, а картинку – с помощью кнопки Clear.

При возникновении затруднений в использовании графического интерфейса JUNO нужно перейти на вкладку Help.

### **3.3 Работа в JUNO. Представления результатов**

Числовые данные на вкладках Real и Model можно записывать стандартно, в виде числа с плавающей точкой (экспоненциальная форма представления вещественных (действительных) чисел, в которой число хранится в виде мантиссы и показателя степени), либо с двойной точностью. Начальный вектор положения и скорости на странице Model нельзя задавать через число с двойной точностью.

На вкладке Real был выбран спутник J43 Arche S/2002\_J1, даты наблюдений были загружены из файла с помощью кнопки Load, а также введены начальная дата и начальный вектор положения и скорости (Рисунок 3.11). Для вычисления показателей нелинейности была нажата кнопка Calculate. После нажатия на кнопку Result открывается окно с результатами.

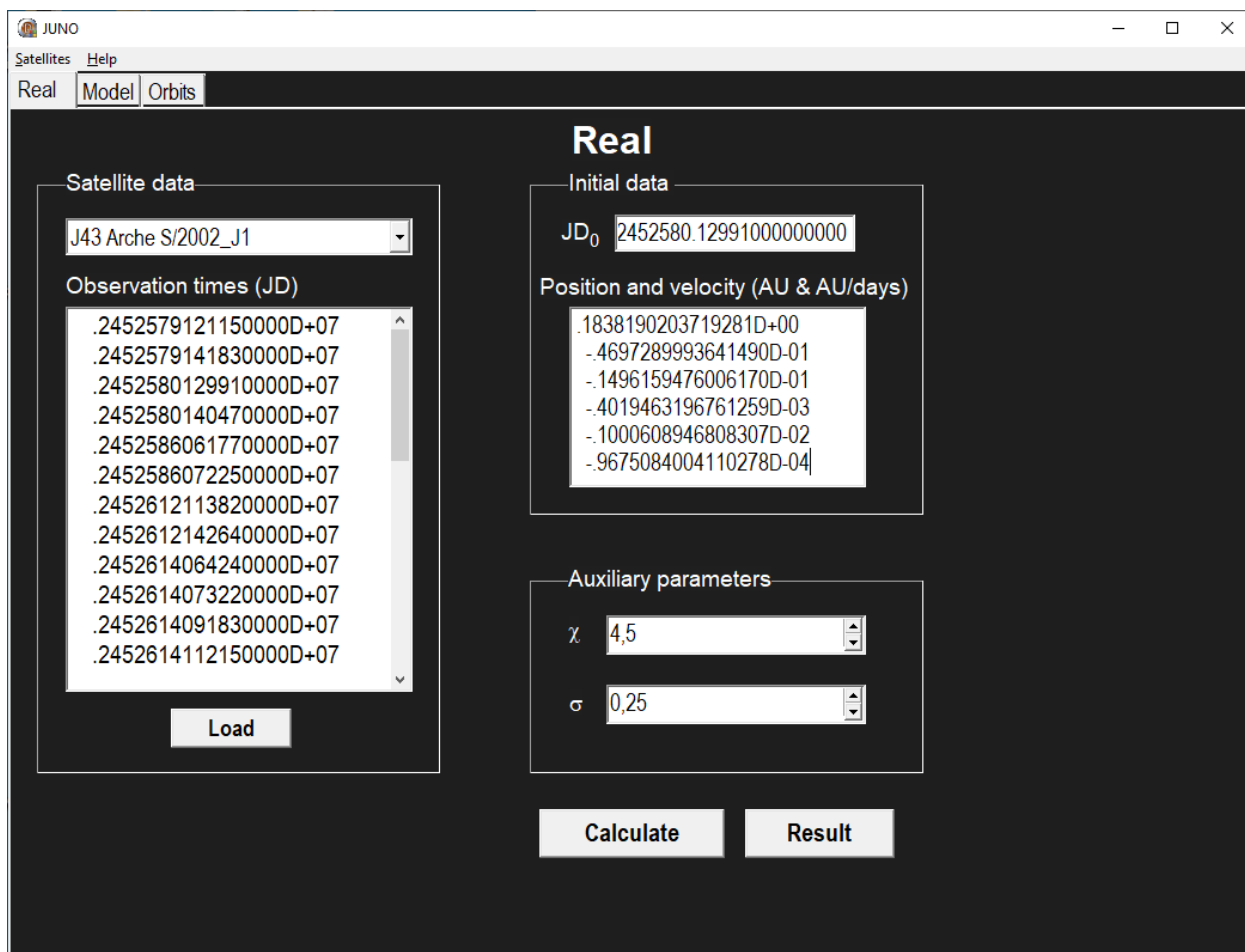


Рисунок 3.11 – Вкладка Real для спутника Архе

В качестве модельного объекта был взят спутник Себетос (далекий спутник Урана). На вкладке Model было записано название спутника на английском, даты наблюдений также, как и у реального объекта были загружены из файла, введены начальная дата и элементы орбиты. После нажатия на кнопку Calculate рассчитываются показатели нелинейности, а также начальный вектор положения и скорости и записывается в соответствующее поле (Рисунок 3.12).

The screenshot shows the JUNO software interface with the 'Model' tab selected. The interface is divided into three main sections: Satellite data, Initial data, and Elements.

**Satellite data:** A text box contains the name 'Setebos'. Below it, a list of observation times (JD) is displayed, ranging from 2.45175006063000E+0006 to 2.45139889988000E+0006. A 'Load' button is at the bottom.

**Initial data:** A text box contains the initial data  $JD_0$  2.45137803981000E+0006. Below it, a list of position and velocity (AU & AU/days) is displayed, ranging from 0.0620131767250165 to -0.000292898490005416. An 'Auxiliary parameters' section contains two spinners:  $\chi$  set to 4.5 and  $\sigma$  set to 0.25.

**Elements:** A list of orbital elements is displayed, ranging from  $a$  0.1164481815 AU to  $M$  179.449. A 'Calculate' button is at the bottom.

**Buttons:** 'Calculate', 'Result', and 'Clear' buttons are located at the bottom right of the interface.

Рисунок 3.12 – Вкладка Model для спутника Сетебос с заполненным окном Elements

Если был введен начальный вектор положения и скорости, то после нажатия на кнопку Calculate, помимо показателей нелинейности рассчитываются элементы орбиты и записываются в блок Elements (Рисунок 3.13).

The screenshot shows the JUNO software interface with the 'Model' tab selected. The interface is divided into several sections:

- Satellite data:** A text box contains the name 'Sebetos'. Below it, a list of observation times (JD) is displayed:
  - 2.45187658677000E+0006
  - 2.45187661488000E+0006
  - 2.45181076362000E+0006
  - 2.45181081253000E+0006
  - 2.45181086067000E+0006
  - 2.45181177180000E+0006
  - 2.45181182190000E+0006
  - 2.45181188171000E+0006
  - 2.45175001820000E+0006
  - 2.45175006063000E+0006
  - 2.45175010303000E+0006
  - 2.45244391498000E+0006
 A 'Load' button is at the bottom.
- Initial data:**
  - JD<sub>0</sub>: 2.45137803981000E+0006
  - Position and velocity (AU & AU/days):
    - 0,061782483586495
    - 0,174556074507823
    - 0,000266946393967003
    - 0,000692600316514179
    - 0,000241895933832096
    - 0,000292898490005416
- Auxiliary parameters:**
  - $\chi$ : 4,5
  - $\sigma$ : 0,25
- Elements:**
  - a: 0,116 AU
  - e: 0,591
  - i: 158,235 °
  - $\omega$ : 359,953 °
  - $\Omega$ : 250,235 °
  - M: 179,449 °

At the bottom right, there are three buttons: 'Calculate', 'Result', and 'Clear'.

Рисунок 3.13 – Вкладка Model для спутника Сетебос с заполненным окном начальных данных

Как видно из Рисунка 3.12 и Рисунка 3.13 переход от элементов к координатам и обратно совпадает в пределах округления чисел.

С добавлением нового объекта график в окне Result перестраивается, меняется ширина гистограмм, т.е. чем больше объектов на графике, тем, соответственно, меньше ширина прямоугольников. Всего на графике может быть помещено 23 объекта, это связано с тем, что высота графика ограничена, а также ширина гистограмм достигнет минимального размера и график будет сложно анализировать. Гистограмма нового объекта строится выше предыдущего, а в таблицу новые данные добавляются в последнюю строку.

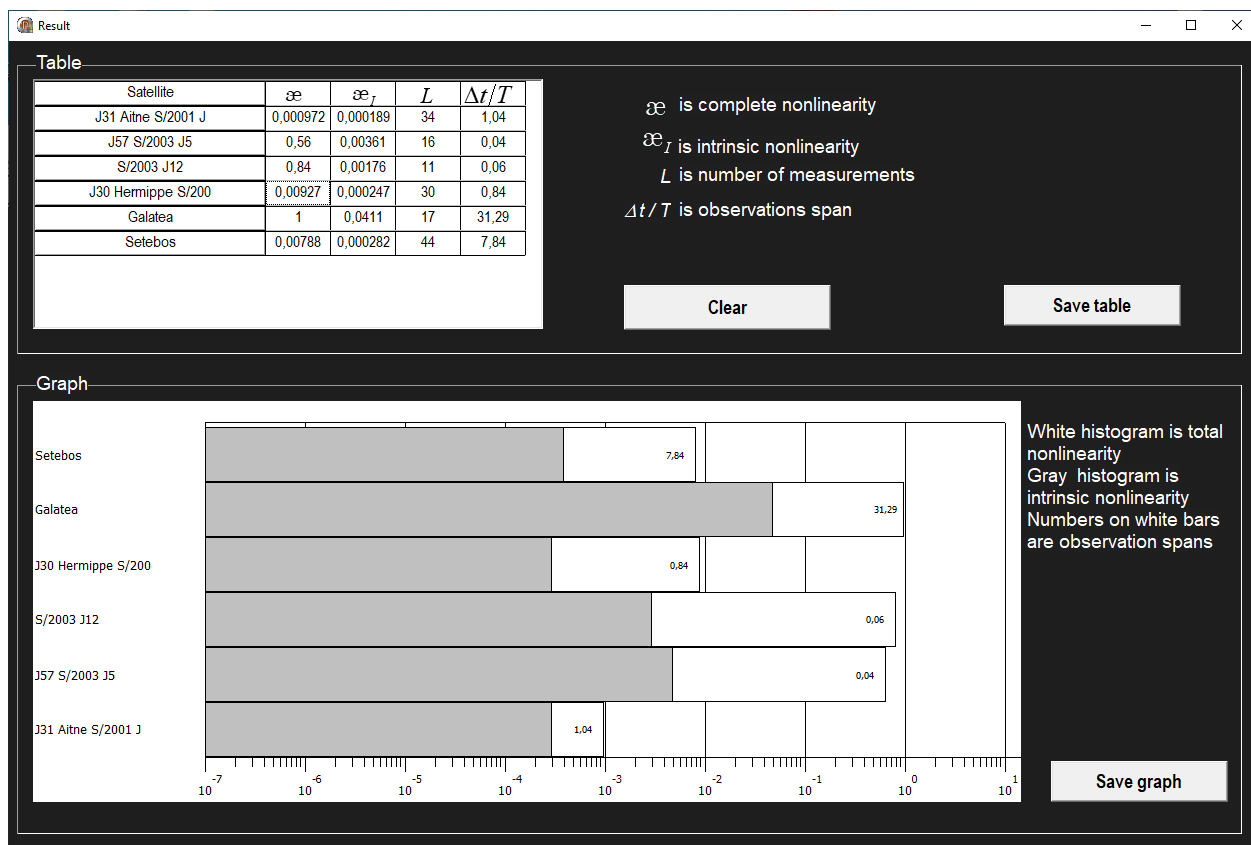


Рисунок 3.14 – Окно Result

На Рисунке 3.14 изображено окно Result после выполнения расчетов нелинейности для выбранных спутников. На этом рисунке представлены данные для четырех далеких спутников Юпитера: Этне, S/2003\_J5, S/2003\_J13, Гермиппе и двух модельных спутников: Галатея (близкий спутник Нептуна) и Себетос (далекий спутник Урана).



## ЗАКЛЮЧЕНИЕ

В данной магистерской диссертационной работе представлены результаты исследования полной и внутренней нелинейностей в обратных задачах для 63 новых далеких спутников Юпитера. Выявлена взаимосвязь между нелинейностью и условиями спутниковых наблюдений. В частности, показано, что полная нелинейность очень сильная, когда период наблюдений меньше 0.1 от орбитального периода и когда окна между наблюдательными данными составляют порядка несколько лет, в то время как внутренняя нелинейность достаточно слабая почти для всех спутников, что указывает на возможность применения нелинейных методов для адекватного моделирования их орбитальной неопределенности. А также в работе подробно описана разработка графического интерфейса JUNO для расчета показателей нелинейности обратных задач динамики реальных и модельных объектов и представления результатов в виде таблицы и графика в интерфейсе программы и с сохранением их в файл.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Jacobson R. A. // *Astron. J.* – 2000. – V. 120. – I. 5. – P. 2679–2686.
2. Sheppard S. S., Jewitt D. C., Kleyna J., Marsden B. G., Jacobson R. A. // *IAU Circ.*, 7900. – V. 1. – 2002.
3. Авдюшев В. А., Баныщикова М. А. // *Астрон. вестник.* – 2007. – Т. 41. – № 5. – С. 446–452.
4. Авдюшев В. А., Баныщикова М. А. // *Известия вузов «Физика».* Изд. ТГУ. Том 53. Вып. 10. 2010. С. 27–30.
5. Desmars J., Arlot S., Arlot J.-E., Lainey V., Vienne A. // *Astron. Astrophys.* – 2009. – V. 62. – P. 321–330.
6. Emel'yanov N. // *PSS.* – 2010. – V. 58. – I. 3. – P. 411–420.
7. Avdyushev V. A. // *Celest. Mech.* – 2011. – V. 110 (4). – P. 369–388.
8. Авдюшев В. А., Сюсина О. М., Тамаров В. А. Нелинейность в обратных задачах спутниковой динамики // *Астрон. вестн.* – 2021. – Т. 55. – №1. – С. 1–13.
9. Avdyushev V. A // *Celest. Mech.* – 2017. – V. 129 (4). – P. 537–552.
10. Bates D. M., Watts D. G. John Wiley & Sons Inc. – 1988. 365 p.
11. Draper N. R., Smith H. John Wiley & Sons, Inc. – 1998. 706 p.
12. Орбитальные параметры спутников Юпитера. – URL: <http://lnfm.sai.msu.ru/neb/rw/natsat/jupord.htm> (дата обращения: 15.04.2021).
13. Natural Satellites Data Center. – URL: <http://www.sai.msu.ru/neb/nss/obspos/bjupooour.htm> (дата обращения: 02.02.2021).
14. Баныщикова М.А. Численное моделирование движения внутренних и внешних спутников Юпитера: дис. ... канд. физ.-мат. наук / М.А. Баныщикова. – Томск, 2009. – С. 11-14.
15. Аксенов Е.П. Теория движения искусственных спутников Земли. – М.: Наука. – 1977. – С. 360.

16. Seidelmann P.K., Abalakin V.K., Bursa M. et al. Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements of the Planet and Satellites: 2000 // *Celest. Mech.* – 2002. – V. 82, N. 1. – P. 83-110.
17. Lieske J.H. Galilean Satellites Ephemerides E5 // *Astron. Astrophys.* – 1998. – V. 129. – P. 205-217.
18. Lainey V., Arlot J.E., Vienne A. New Accurate Ephemerides for the Galilean Satellites of Jupiter. II. Fitting the Observations // *Astron. Astrophys.* – 2004b. – V. 427. – P. 371-376.
19. Stadish E.M. JPL Planetary and Lunar Ephemerides, DE405/LE405 // *Interoffice Memorandum.* – 1998. – V. 312. – P. 1-18.
20. Брумберг В.А. Релятивистская небесная механика. – М. Наука, 1972. – С. 382.
21. Sampson R.A. Jupiter, Satellites I., II., III., IV. // *Mon. Notie. Roy. Astron. Soc.*, 1910. – V.71. – P.124.
22. Lainey V., Duriez L., Vienne A, New Accurate Ephemerides for the Galilean Satellites of Jupiter. I. Numerical Integration of Elaborated Equations of Motion // *Astron. Astrophys.* – 2004 a. – V. 420. – P. 1171-1183.
23. Бордовицына Т.В., Быкова Л.Е. Теория движения и эфемериды VI и VII спутников Юпитера на 1979-2000 годы // *Изд-во ТГУ.* – 1998. – С. 120.
24. Rocher P., Chapront J. Observations and ephemerides of the faint satellites of Jupiter // *Astron. Astrophys.* – 1996. – V. 311. – P. 710-714.
25. Emelyanov N.V. Ephemerides of the outer Jovian satellites // *Astron. Astrophys.*, 2005. – V. 435. – I. 3. – P. 1173-1179.
26. Дубошин Г.Н. Теория притяжения. М.: Изд-во физ.-мат. лит. – 1961. – С. 208.
27. Архангельский А. Я. Программирование в Delphi для Windows. Версии 2006, 2007, Turbo Delphi. – М.: ООО «Бином-Пресс», 2007. – С. 1239.
28. Языки программирования. Delphi. Уроки Delphi. – Компонент Delphi Bevel // Тюрин И. В. [Б. м.], 2011. – URL: <http://www.delphi-prg.ru/komponent-delphi-bevel> (дата обращения: 27.05.2021).

29. Delphi FAQ – Часто задаваемые вопросы. – FloatSpinEdit. Компонент для ввода целых и дробных чисел // «Delphi Sources» by BrokenByte Software. Delphi World FAQ [Б. м.], 2004. – URL: <https://delphisources.ru/pages/faq/base/floatspinedit.html> (дата обращения: 27.05.2021).

30. Википедия – свободная энциклопедия [Электронный ресурс] // Логарифмический масштаб – URL: [https://ru.wikipedia.org/wiki/Логарифмический\\_масштаб](https://ru.wikipedia.org/wiki/Логарифмический_масштаб) (дата обращения: 27.05.2021).

31. Spacegid.com – Ваш гид в мире космоса // Спутники Юпитера – URL: <https://spacegid.com/sputnik-yupitera.html> (дата обращения: 15.04.2021).

# Отчет о проверке на заимствования №1



Автор: Шмидт Екатерина Евгеньевна

Проверяющий: (schmidt-katerina@mail.ru / ID: 6698445)

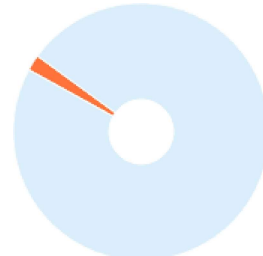
Отчет предоставлен сервисом «Антиплагиат» - [users.antiplagiat.ru](https://users.antiplagiat.ru)

## ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 13  
Начало загрузки: 14.06.2021 15:56:39  
Длительность загрузки: 00:00:02  
Имя исходного файла: Магистерская диссертация.pdf  
Название документа: Магистерская диссертация  
Размер текста: 61 КБ  
Символов в тексте: 62334  
Слов в тексте: 7364  
Число предложений: 524

## ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 14.06.2021 15:56:41  
Длительность проверки: 00:00:04  
Корректировка от 15.06.2021 10:36:43  
Комментарии: не указано  
Модули поиска: Интернет



### ЗАИМСТВОВАНИЯ

2,31%

### САМОЦИТИРОВАНИЯ

0%

### ЦИТИРОВАНИЯ

0%

### ОРИГИНАЛЬНОСТЬ

97,69%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.  
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.

Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	1,24%	<a href="http://vital.lib.tsu.ru/vital/access/services/Download/vtls:000521275/SOURCE1">http://vital.lib.tsu.ru/vital/access/services/Download/vtls:000521275/SOURCE1</a> (2/2)	27 Янв 2020	Интернет
[02]	0,7%	Удаление старых файлов. - Форум программистов <a href="http://programmersforum.ru">http://programmersforum.ru</a>	12 Авг 2017	Интернет
[03]	0,37%	Численное моделирование движения внутренних и внешних спутников Юпитера   dissertation.com.ua <a href="http://dissertation.com.ua">http://dissertation.com.ua</a>	13 Янв 2018	Интернет