МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра космического приборостроения и систем связи

УТВЕРЖДАЮ

Проректор по учебной работе
_______________ О.Г. Локтионова
«09» 09 _______________ 2020 г.

ОСНОВЫ МОДЕЛИРОВАНИЯ ДВИЖЕНИЯ КОСМИЧЕСКИХ АППАРАТОВ И СРЕДСТВА АВТОМАТИЗАЦИИ РАСЧЕТОВ

Методические указания к лабораторным занятиям по дисциплине
«Основы конструкций космических аппаратов»

Курск - 2020

УДК 681.5

Составитель Е. О. Брежнева

Рецензент

Доктор технических наук, профессор кафедры вычислительной техники *И.Е. Чернецкая*

Основы моделирования движения космических аппаратов и средства автоматизации расчетов: методические указания к лабораторным занятиям по дисциплине «Основы конструкций космических аппаратов»/ Юго-Зап. гос. ун-т; сост. Е.О. Брежнева. Курск, 2020. - 58 с.: Ил. 18. Табл. 8. Библиограф.: с.52

Излагаются краткие теоретические сведения об основах моделирования и средствах автоматизации расчетов. Приведены контрольные вопросы и задания на моделирование движения космических объектов.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям автоматики и электроники (УМО АЭ).

Предназначены для бакалавров направления подготовки 11.03.03 «Конструирование и технология электронных средств».

Текст печатается в авторской редакции

Подписано в печать 09.09.2020. Формат 60×84 1/16.
Усл. печ. л. 3,37. Уч.- изд. л. 3,05. Тираж *30* экз. Заказ *251*. Бесплатно.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

На начальном этапе проектирования космических аппаратов, как правило, имеется ограниченный набор исходных данных. Однако, накоплен большой теоретический и практический материал, позволяющий построить математические модели, описывающие состав бортовых систем и конструкцию практически с любой степенью точности. Особенность проектирования космических аппаратов состоит в том, что существует такое разнообразие объектов, что попытки получить обобщенные зависимости встречают большие трудности.

Конкретизация назначения космических аппаратов позволяет при построении математических моделей использовать конкретные статистические данные традиционных схем [1]. Кроме того, на начальном этапе проектирования можно абстрагироваться от конкретной схемы искусственного спутника Земли (ИСЗ), что облегчает исследование и построение математических моделей.

Особенность проектирования космических аппаратов (КА) заключается также в том, что существует мало теоретических работ, связанных с выбором параметров и логикой процесса их проектирования на этапе предэскизных разработок. Кроме того, отсутствуют достаточно простые и универсальные математические модели, устанавливающие связь между исходными данными, проектными параметрами, а также характеристиками КА и параметрами функционирования его систем.

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

1.1 ЦЕЛИ И ЗАДАЧИ МОДЕЛИРОВАНИЯ

Модель - это:

1) некоторое упрощенное подобие реального объекта;
2) воспроизведение предмета в уменьшенном или увеличенном виде (макет);
3) схема, изображение или описание какого - либо явления или процесса в природе и обществе;
4) физический или информационный аналог объекта, функционирование которого по определённым параметрам подобно функционированию реального объекта;
5) некий объект - заменитель, который в определённых условиях может заменять объект - оригинал, воспроизводя интересующие нас его свойства и характеристики, причем имеет существенные преимущества и удобства (наглядность, обозримость, доступность испытаний, лёгкость оперирования с ним и так далее);
6) новый объект, который отражает некоторые стороны изучаемого объекта или явления, существенные с точки зрения целей моделирования;
7) новый объект (реальный, информационный или воображаемый), отличный от исходного, который обладает существенными для целей моделирования свойствами и в рамках этих целей полностью заменяет исходный объект.

Модель представляет собой способ существования знаний. Информационная модель - это описание моделируемого объекта на одном из языков кодирования информации.

Моделирование - это:

1) построение моделей реально существующих объектов (предметов, явлений, процессов);
2) замена реального объекта его подходящей копией;
3) исследование свойств объекта при помощи эксперимента на модели.

Моделирование является неотъемлемым элементом любой целенаправленной деятельности. Моделирование представляет собой один из основных методов познания.

Целью моделирования является разработка и создание адекватной модели объекта или процесса, удовлетворяющей всем предъявляемым к ней требованиям, соотнесенным с поставленными задачами моделирования.

Задачи моделирования:

1) Познание окружающего мира. Зачем человек создает модели? Чтобы ответить на этот вопрос, надо заглянуть в далекое прошлое. Несколько миллионов лет назад, на заре человечества, первобытные люди изучали окружающую природу, чтобы научиться противостоять природным стихиям, пользоваться природными благами, просто выживать. Накопленные знания передавались из поколения в поколение устно, позже письменно, наконец, с помощью предметных моделей. Так родилась, к примеру, модель земного шара — глобус, — позволяющая получить наглядное представление о форме нашей планеты, ее вращении вокруг собственной оси и расположении материков. Такие модели позволяют понять, как устроен конкретный объект, узнать его основные свойства, установить законы его развития и взаимодействия с окружающим миром моделей.

2) Создание объектов с заданными свойствами (задача типа «Как сделать, чтобы...»). Накопив достаточно знаний, человек задал себе вопрос: «Нельзя ли создать объект с заданными свойствами и возможностями, чтобы противодействовать стихиям или ставить себе на службу природные явления?» Человек стал строить модели еще не существующих объектов. Так родились идеи создания ветряных мельниц, различных механизмов, даже обыкновенного зонтика. Многие из этих моделей стали в настоящее время реальностью. Это объекты, созданные руками человека. Определение последствий воздействия на объект и принятие правильного решения (задача типа «Что будет, если...»: что будет, если увеличить плату за проезд в транспорте, или что произойдет, если закопать ядерные отходы в такой-то местности?).

3) Эффективность управления объектом (или процессом).

4) Выявление наиболее существенных факторов, формирующих те или иные свойства объекта;

5) Прогнозирование поведения сложных процессов и явлений.

1.2 ЭТАПЫ МОДЕЛИРОВАНИЯ

Основные этапы построения математической модели:
1) исследование объекта и содержательная постановка задачи;
2) концептуальная постановка задачи;
3) выбор метода моделирования;
4) реализация;
5) проверка адекватности;
6) анализ результатов.

Исследование объекта и содержательная постановка задачи. Математические модели, особенно использующие численные методы, требуют для своего построения значительных интеллектуальных, финансовых и временных затрат. Поэтому решение о разработке новой модели принимается лишь в случае отсутствия иных, более простых путей решения возникших проблем (например, модификации одной из существующих моделей). Необходимость в новой модели может появиться в связи с проведением научных исследований, выполнением проектно-конструкторских работ, созданием систем автоматического управления. Основной целью обследования объекта моделирования является подготовка содержательной постановки задачи моделирования, т.е. списка основных вопросов об объекте моделирования, интересующих заказчика.

На этом этапе важную роль играют специалисты – постановщики задач, которые должны не только хорошо разбираться в предметной области моделирования, знать возможности современной вычислительной техники, но и быть достаточно коммуникабельными, способными «разговорить» практиков, хорошо знающих объект моделирования.

На основании анализа всей собранной информации постановщик задачи должен сформулировать такие требования к будущей модели, которые: с одной стороны, удовлетворяли бы заказчика, а с другой - позволяли бы реализовать модель в заданные сроки и в рамках выделенных материальных средств.

Этап обследования объекта моделирования включает следующие работы:

1. выявление основных факторов, механизмов, влияющих на поведение объекта моделирования, определение параметров, позволяющих описывать моделируемый объект;

2. сбор и проверка имеющихся экспериментальных данных об объектах-аналогах, проведение при необходимости дополнительных экспериментов;

3. аналитический обзор литературных источников, анализ и сравнение между собой построенных ранее моделей данного объекта (или подобных рассматриваемому объекту);

4. анализ и обобщение всего накопленного материала, разработка общего плана создания математической модели.

На основе собранной информации постановщик и заказчик формулируют содержательную или техническую постановку задачи моделирования, которая, как правило, не бывает окончательной и может уточняться в процессе разработки модели. Весь собранный материал об объекте, содержательная постановка задачи, требования к реализации модели и представлению результатов, оформляются в виде технического задания на проектирование и разработку модели.

Концептуальная постановка задачи. Содержательная модель является синтезом когнитивных моделей, каждого из членов рабочей группы. На основании содержательной модели разрабатывается концептуальная, или «естественнонаучная» постановка задачи моделирования. Концептуальная постановка задачи моделирования - это сформулированный в терминах конкретных дисциплин (физики, химии, биологии т.д.) перечень основных вопросов, интересующих заказчика, а также совокупность гипотез относительно свойств и поведения объекта моделирования. Наибольшие трудности при формулировке концептуальной постановки приходится преодолевать в моделях, находящихся на «стыке» различных дисциплин. Различия традиций, понятий и языков, используемых для описания одних и тех же объектов, являются очень серьезными препятствиями, возникающими при создании «междисциплинарных» моделей.

Выбор метода моделирования. Данный этап следует разбить на два взаимосвязанных подэтапа: математическая постановка задачи моделирования и выбор и обоснование метода решения задачи.

Математическая постановка задачи. Концептуальная постановка позволяет сформулировать математическую постановку задачи моделирования, т.е. совокупность математических соотношений, описывающих поведение и свойства объекта моделирования. Как было отмечено ранее, совокупность математических соотношений определяет вид оператора модели. Наиболее простым будет оператор модели в случае, если он представлен системой алгебраических уравнений. Следует отметить, что во многих областях знаний (механике, физике, биологии и т.д.) принято выделять законы, справедливые для всех объектов исследования данной области знаний, и соотношения, описывающие поведение отдельных объектов или их совокупностей. К числу первых в физике и механике относятся, например, уравнения баланса массы, количества движения, энергии и т.д., справедливые при определенных условиях для любых материальных тел, независимо от их конкретного строения, структуры, состояния, химического состава. Уравнения этого класса подтверждены огромным количеством экспериментов, хорошо изучены и в силу этого применяются в соответствующих математических моделях как данность. Соотношения второго класса в физике и механике называют уравнениями состояния. Они устанавливают особенности поведения материальных объектов или их совокупностей (например, жидкостей или газов) при воздействиях различных внешних факторов. Необходимо отметить, что определяющие соотношения - это основной элемент, «сердцевина» любой математической модели физико-механических процессов. Именно ошибки в выборе или установлении определяющих соотношений приводят к количественно неверным результатам моделирования. Совокупность математических соотношений указанных двух классов определяет оператор модели. В большинстве случаев оператор модели включает в себя систему обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных и интегро-дифференциальных уравнений. Для обеспечения корректности постановки задачи к системе уравнений добавляются начальные или граничные условия, которые, в свою очередь, могут быть алгебраическими или дифференциальными соотношениями различного порядка.

Для контроля правильности полученной системы математических соотношений требуется проведение ряда обязательных проверок:

1. контроль размерностей, включающий правило, согласно которому приравниваться и складываться могут только величины одинаковой размерности;

2. контроль порядков, состоящий из грубой оценки сравнительных порядков складываемых величин и исключением малозначимых параметров;

3. контроль характера зависимостей заключается в проверке того, что направление и скорость изменения выходных параметров модели, вытекающие из математических соотношений, такие, как это следует непосредственно из «физического» смысла изучаемой модели;

4. контроль экстремальных ситуаций - проверка того, какой вид принимают математические соотношения, а также результаты моделирования, если параметры модели или их комбинации приближаются к предельно допустимым значениям, чаще всего к нулю или бесконечности. В подобных экстремальных ситуациях модель часто упрощается, математические соотношения приобретают более наглядный смысл, упрощается их проверка;

5. контроль граничных условий, включающий проверку того, что граничные условия действительно наложены, что они использованы в процессе построения искомого решения и что значения выходных параметров модели на самом деле удовлетворяют данным условиям;

6. контроль физического смысла - проверка физического или иного смысла исходных и промежуточных соотношений;

7. контроль математической замкнутости, состоящий в проверке того, что выписанная система математических соотношений дает возможность, притом однозначно, решить поставленную математическую задачу.

Выбор и обоснование выбора метода решения задачи. При использовании разработанных математических моделей, как правило, требуется найти зависимость некоторых неизвестных заранее параметров объекта моделирования (например, координат и скорости центра масс тела), удовлетворяющих определенной системе уравнений. Таким образом, поиск решения задачи сводится к отысканию некоторых зависимостей искомых величин от исходных параметров модели. Как было отмечено ранее, все методы решения задач, составляющих «ядро» математических моделей, можно подразделить на аналитические и алгоритмические. Аналитические методы более удобны для последующего анализа результатов, но применимы лишь для относительно простых моделей. В случае, если математическая задача допускает аналитическое решение, оно, без сомнения, предпочтительнее численного. Алгоритмические методы сводятся к некоторому алгоритму, реализующему вычислительный эксперимент с использованием ЭВМ. Точность моделирования в подобном эксперименте существенно зависит от выбранного метода и его параметров (например, шага интегрирования). Алгоритмические методы, как правило, более трудоемки в реализации, требуют обширной библиотеки специального программного обеспечения и мощной вычислительной техники. Общим для всех численных методов является сведение математической задачи к конечномерной. Это чаще всего достигается дискретизацией исходной задачи, т.е. переходом от функции непрерывного аргумента к функциям дискретного аргумента. Полученное решение дискретной задачи принимается за приближенное решение исходной математической задачи. Применение любого численного метода неминуемо приводит к погрешности результатов решения задачи.

Численный, или приближенный, метод реализуется всегда в виде вычислительного алгоритма. Прежде всего, алгоритм должен быть реализуем и обеспечивать решение задачи за допустимое машинное время. Важной характеристикой алгоритма является его погрешность. Для очень малых значений погрешности время вычислений может быть недопустимо большим. Поэтому на практике добиваются некоторого компромисса между точностью и затрачиваемым машинным временем. Если погрешность в процессе вычислений неограниченно возрастает, то такой алгоритм называется неустойчивым, или расходящимся. В противном случае алгоритм называется устойчивым, или сходящимся.

Проверка адекватности. Под адекватностью математической модели понимается степень соответствия результатов моделирования – экспериментальным данным или тестовой задаче.

Проверка адекватности модели преследует две цели:

1. убедиться в справедливости гипотез, принятых на этапах концептуальной и математической постановок;

2. установить, что точность полученных результатов соответствует точности, оговоренной в техническом задании.

Проверка разработанной математической модели выполняется путем сравнения с имеющимися экспериментальными данными о реальном объекте или с результатами других, созданных ранее и хорошо себя зарекомендовавших моделей. В первом случае говорят о проверке путем сравнения с экспериментом, во втором - о сравнении с результатами решения тестовой задачи. Решение вопроса о точности моделирования зависит от требований, предъявляемых к модели, и ее назначения. В моделях, предназначенных для выполнения оценочных расчетов, удовлетворительной считается точность 10-15 %. В моделях, используемых в управляющих системах, требуемая точность может быть 1-2% и даже более. Как правило, различают качественное и количественное совпадение результатов сравнения. При качественном сравнении требуется лишь совпадение некоторых характерных особенностей исследуемых параметров (например, наличие экстремальных точек, возрастание или убывание параметра). При количественном сравнении большое значение следует придавать точности исходных данных для моделирования и соответствующих им значений сравниваемых параметров.

Если математические модели систем и процессов строятся в основном как подобные детерминированные модели, обладающие общим с оригиналом математическим описанием, то проверки адекватности математической модели достаточно убедиться в выполнении двух свойств: точности и непротиворечивости. Под точностью понимается обобщенная характеристика рассогласования соответствующего параметра модели и оригинала, которая должна быть не больше, чем заранее заданное значение приемлемой погрешности.

1.3 ПРОЦЕСС ПРОЕКТИРОВАНИЯ КОСМИЧЕСКИХ АППАРАТОВ

Процесс проектирования космических аппаратов представляет собой многоуровневый итерационный процесс, в течении которого рассчитываются характеристики аппарата и его массовая сводка. Для того, чтобы организовать такой процесс, необходимо разработать логику и технологию его. Учитывая, что для решения такой сложной задачи используется большое число уравнений, исходных данных и численных значений, входящих в уравнение коэффициентов необходимо применение ЭВМ.

Модель процесса проектирования должна содержать следующие основные модели: существования, возможности, движения и масс.

На рисунке 1 представлена блок-схема процедуры автоматизированного проектирования ИСЗ.

1.3.1 МОДЕЛЬ СУЩЕСТВОВАНИЯ КА

Модель существования КА содержит уравнения, связывающие между собой начальную массу аппарата с некоторыми основными проектными параметрами. В состав этих уравнений из обычно задаваемых исходных данных входит масса целевой аппаратуры. Кроме того, в эти уравнения входят статистические коэффициенты, которые характеризуют конструкцию, служебные системы и другие элементы КА. Для общности эти уравнения представляются в безразмерном виде.

1.3.2 МОДЕЛЬ ВОЗМОЖНОСТИ

Модель возможности представляет собой систему уравнений, описывающих связь между характеристиками КА и некоторыми основными проектными параметрами. Фактически эта система уравнений является упрощенным и обобщенным решением задачи выполнения цели полета с учетом только основных действующих факторов.

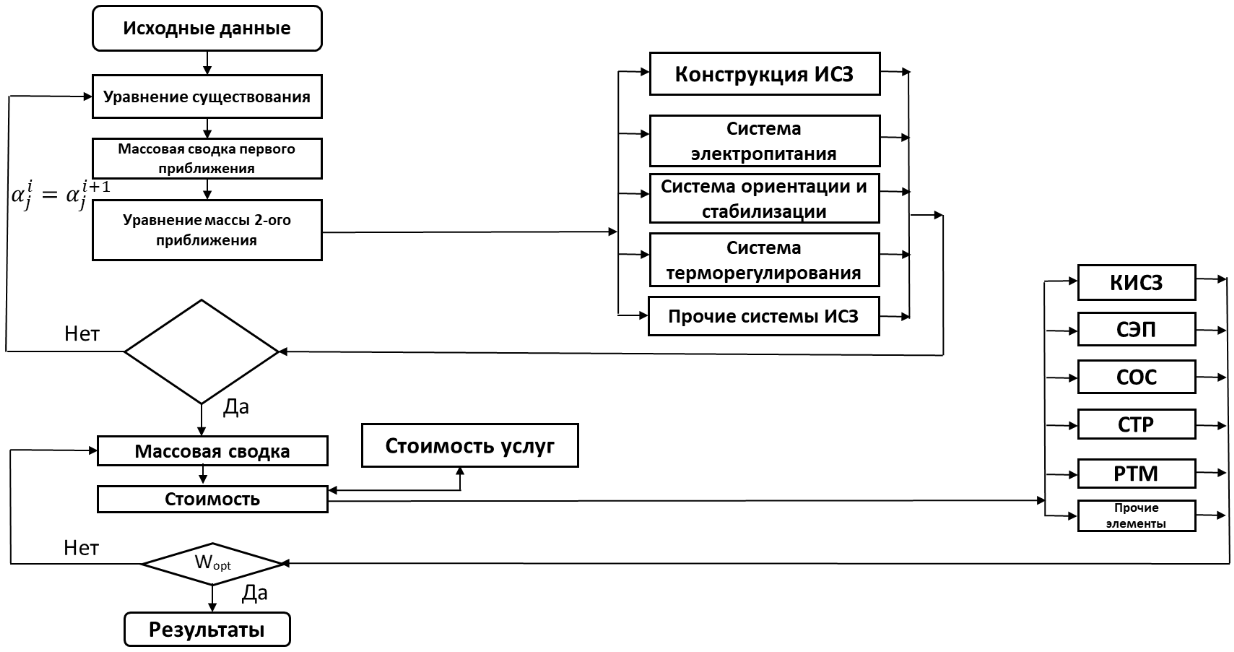

Рисунок 1 - Блок-схема процедуры автоматизированного проектирования ИСЗ: КИСЗ – конструкция ИСЗ, СЭП – система электропитания, СОС – система ориентации и стабилизации, СТР – система терморегулирования, РТМ - радиотелеметрия

1.3.3 МОДЕЛЬ ДВИЖЕНИЯ

Модель движения – это система дифференциальных уравнений, описывающих движение КА. Решение этих уравнений позволяет, во-первых, подтвердить тот факт, что с данными проектными параметрами и массово-геометрическими характеристиками КА выполняет поставленную задачу, и, во-вторых, получить дополнительную информацию для уточнения характеристик служебных систем.

Эти три модели образуют первый (верхний) уровень проектирования. Он является недостаточным для определения масс, размеров и энергетики аппарата, поэтому необходимо иметь модель, описывающую аппарат с учетом его основных составных частей. Эту модель назовем моделью для расчёта массовых характеристик.

1.3.4 МОДЕЛЬ МАСС

Модель масс включает в себя набор уравнений, с помощью которых определяются массы составных частей: корпуса, бортовых систем и др. Эта модель позволяет получить более точное значение массы ИСЗ на основе применения уравнений, включающих статистические коэффициенты, и описывающие элементы конструкции и бортовых систем. Таких моделей может быть несколько в зависимости от степени проработки конструкции и бортовых систем.

1.3.5 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ КОСМИЧЕСКОГО АППАРАТА

В процессе решения задач оптимизации проектных параметров КА возникает необходимость в моделях существования, включающих описание наиболее существенных элементов – служебных систем, масса которых, с одной стороны, зависит от требований со стороны целевой аппаратуры, а с другой, определяется их структурой и собственными параметрами каждой системы.

В [1] принято следующее основное уравнение существования КА:

$$\mu_{\text{пн}} = \frac{1 - \frac{\alpha_{\text{констр}}}{\rho_{\text{ИСЗ}}} - \delta_{\text{проч}}}{1 + \varepsilon_{\text{СЭП}} + \beta_{\text{СТР}} + \gamma_{\text{СОС}} - \delta_{\text{проч}}},$$

показывающее связь между относительной массой целевой аппаратуры и статистическими коэффициентами, описывающими долю каждой служебной системы в составе КА.

В таблице 1 представлены принятые в уравнении обозначения.

Анализ ИСЗ, имеющих мощность 0,5-2,5 кВТ, дает формулу

$$m_{\text{ЦА}}=46{,}7+73{,}3W_{\text{БРК}}.$$

Энергетическая мощность, требуемая для обеспечения работы системы ориентации и стабилизации, системы терморегулирования и других систем зависит от энергетической мощности целевой аппаратуры

$$W_{\text{ИСЗ}}=-0{,}10+1{,}6\,W_{\text{БРК}}.$$

Таблица 1 – Принятые обозначения

$\mu_{\text{ПН}}=\dfrac{m_{\text{ЦА}}}{m_{\text{ИСЗ}}}$	относительная масса полезной нагрузки
$\rho_{\text{ИСЗ}}=\dfrac{m_{\text{ИСЗ}}}{V_{\text{ИСЗ}}}$	плотность размещения аппаратуры и систем
$\alpha_{\text{констр}}=\dfrac{m_{\text{констр}}}{V_{\text{ИСЗ}}}$	удельная масса конструкции
$\varepsilon_{\text{СЭП}}=\dfrac{m_{\text{СЭП}}}{m_{\text{ЦА}}}$	относительная масса системы энергопитания
$\beta_{\text{СТР}}=\dfrac{m_{\text{СТР}}}{m_{\text{ЦА}}}$	относительная масса системы терморегулирования
$\gamma_{\text{СОС}}=\dfrac{m_{\text{СОС}}}{m_{\text{ЦА}}}$	относительная масса системы ориентации и стабилизации
$\delta_{\text{проч}}=\dfrac{m_{\text{проч}}}{(m_{\text{ИСЗ}}-m_{\text{ЦА}})}$	относительная масса прочих элементов КА
$m_{\text{ЦА}}$	масса целевой аппаратуры
$m_{\text{ИСЗ}}$	масса космического аппарата
$V_{\text{ИСЗ}}$	объем космического аппарата
$m_{\text{констр}}$	масса конструкции
$m_{\text{СЭП}}$	масса системы энергопитания
$m_{\text{СТР}}$	масса системы терморегулирования
$m_{\text{СОС}}$	масса СОС
$m_{\text{проч}}$	масса прочих элементов

Массу системы энергопитания на основе солнечных батарей представляют в виде суммы масс основных составляющих: массы панелей солнечных батарей, массы буферных аккумуляторов, массы системы контроля работы.

Масса панелей солнечных батарей:

$$m_{\text{СБ}} = \frac{0,9W}{\varphi}(T_{\text{с.а.с.}} + 1)^{1,15},$$

где W – расчетная средняя мощность, отдаваемая в систему энергопитания, φ – коэффициент полезного действия солнечных батарей, $T_{\text{с.а.с.}}$ – срок активного существования ИСЗ.

Масса буферных аккумуляторов:

$$m_{\text{акк}} = 20,0E(1 + 3,0T_{\text{с.а.с.}}^{0,5}),$$

где E – расчетная емкость буферных аккумуляторов.

Масса системы контроля работы СЭП:

$$m_{\text{контр}} = 30,0(W + 1)^{0,5}.$$

Масса системы терморегулирования зависит от холодопроизводительности Q и срока активного существования ИСЗ:

$$m_{\text{СТР}} = 110Q^{0.8}(T_{\text{с.а.с.}} + 0,2)^{0,25}.$$

Массу системы ориентации и стабилизации можно представить в виде следующей функции массы ИСЗ:

$$m_{\text{СОС}} = -50m_{\text{ИСЗ}}^2 + 215m_{\text{ИСЗ}} - 35.$$

2. СРЕДСТВА АВТОМАТИЗАЦИИ РАСЧЕТОВ

Использование разнообразных средств автоматизации расчетов параметров электрических устройств, узлов и блоков позволяет облегчить задачу проектирования космического аппарата.

Существует множество программ, используемых для автоматизации математических расчетов, но самыми распространенными, доступными и простыми в использовании являются *Excel* и *Matlab*.

Далее рассмотрены интерфейс, библиотеки, функциональные возможности этих программ, проведен сравнительный анализ.

2.1 *MATLAB*

MATLAB является одной из самых распространенных, старейших и мощнейших систем автоматизации математических расчетов, созданных компанией *Math Works* в конце 70-х годов. В процессе развития в систему добавлялись различные дополнения в виде пакетов расширений, благодаря которым росли ее возможности.

В базовый набор слов системы входят спецзнаки, знаки арифметических и логических операций, арифметические, алгебраические, тригонометрические и некоторые специальные функции, векторные и матричные функции, операторы построения гистограмм, графиков в декартовой и полярной системах координат, трехмерных поверхностей и многое другое.

MATLAB широко используется в качестве вычислительного инструмента в науке и технике, охватывающего области физики, химии, математики и всех технических потоков. Чаще всего данная система используется в следующих областях:

- обработка сигналов и связь;
- системы управления;
- тест и измерение;
- вычислительные финансы и биология.

К достоинствам *MATLAB*, кроме сказанного выше, можно также отнести легкость написания программ и открытость ее архитектуры, что позволяет создавать новые пакеты расширений.

К недостаткам системы можно отнести линейную запись выражений и формул, что по сравнению, например, с системой

MATHCAD выглядит несколько неудобным, и некоторую громоздкость системы, что создает сложности в ее изучении. В данном отчете будет представлена версия *MATLAB R*2013*b* (рисунок 2).

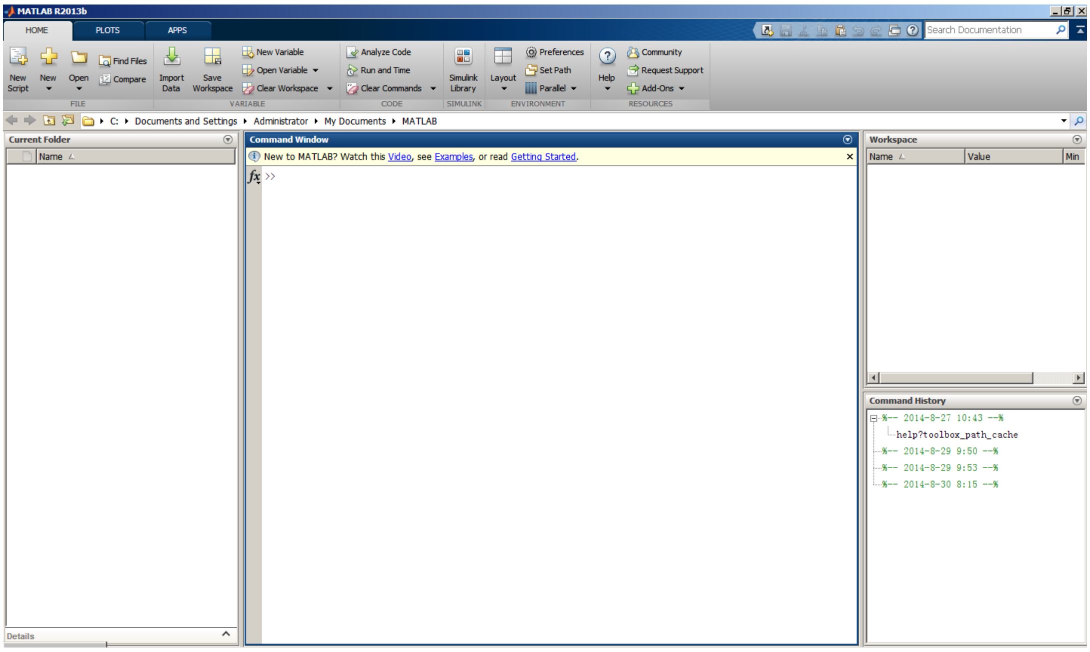

Рисунок 2 – Рабочая среда *MATLAB*

На рисунке 2 представлена рабочая среда, которая содержит следующие элементы:

1. *New script* – новый сценарий (создание нового *m*-файла);

2. *New* (открытие подменю с позициями):

- *New script* – открывает окно редактора/отладчика m-файлов;

- *Function* – открывает окно с шаблоном для создания функции пользователя;

- *Figure* – открытие пустого окна графики;

3. *Open* – выводит подменю с позициями:

- *Open* – открывает существующий файл (m-файл);

- *Find files* – открывает окно для поиска заданного файла;

4. *Import data* – открывает окно импорта файлов данных;

5. *Save workspace* – открывает окно записи рабочей области в виде файла с заданным именем;

6. *New variable* – создание новой переменной (рисунок 3);

7. *Clear workspace* – очистка рабочей области;

8. *Set Path* – открывает окно установки путей доступа файловой системы;

9. *Preference* – открывает окно настройки элементов интерфейса;

10. *Help* – открывает окно справки.

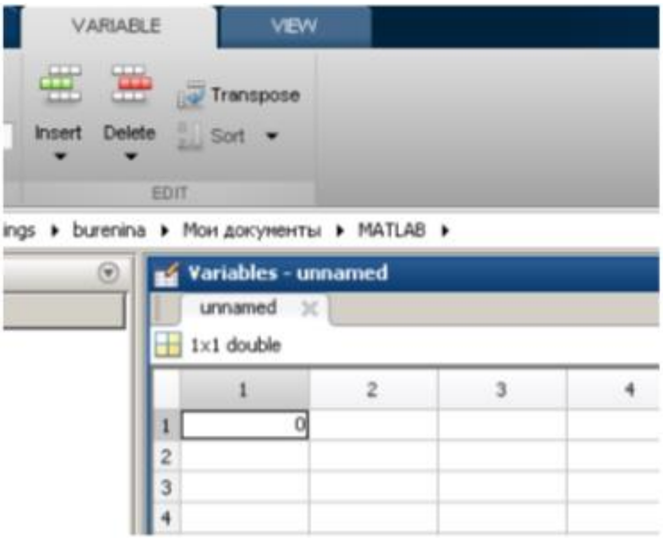

Рисунок 3 – Окно создания новой переменной

Редактора/отладчика *m*-файлов использует следующие цветовые выделения набранного текста:

- синий цвет – ключевые слова языка программирования;
- зеленый цвет – комментарий;
- черный цвет – переменные, операторы;
- красный цвет – синтаксические ошибки.

В данной системе существует два режима работы:

1) Режим прямых вычислений, в котором команда вводится после приглашающего символа «>>». Ввод команды завершается клавишей «*Enter*»;

2) Режим программы, в котором пользователь оформляет последовательность вычислений в виде программы.

Рассмотрим основные функциональные возможности данной системы автоматизации математических расчетов (таблица 2).

В арифметических выражениях можно использовать следующие знаки арифметических операций:

1) + (сложение);
2) – (вычитание);
3) * (умножение);
4) / (деление);

5) .^ (возведение в степень).

Таблица 2 – Математические функции

Математическая функция	Описание
sum(a)	Вычисление суммы
sqrt(a)	Вычисление квадратного корня
round(a)	Округление до целого
exp(a)	Вычисление экспоненты
log(a)	Вычисление натурального логарифма
log10(a)	Вычисление десятичного логарифма
sin(a)	Вычисление синуса
cos(a)	Вычисление косинуса

2.2 *EXCEL*

Excel – приложение *Microsoft Office*, предназначенное для работы с электронными таблицами. Данное приложение позволяет хранить, организовывать и анализировать информацию, работать с ней.

Первая версия *Excel* была выпущена в 1985 году.

В первую очередь данное приложение было создано для обработки данных. К этому понятию относят:

- проведение различных вычислений с помощью формул и функций, встроенных в редактор;
- построение диаграмм;
- обработка данных в списках;
- решение задач оптимизации;
- статистическая обработка данных, анализ и прогнозирование.

К преимуществам можно отнести:

- доступность программы и ее понятность для пользователя, в отличие от *MATLAB*, который на первых этапах изучения работы вызывает трудности;
- дешевизна. Данная программа дешевле *MATLAB*, кроме того существует онлайн-версия, которая не уступает по функционалу обычной программе;
- обработка очень больших массивов данных;

- удобное и наглядное представление данных в виде таблицы, в то время как в *MATLAB* все представлено строкой, что не всегда удобно.

Но *MATLAB*, как уже упоминалось выше, обладает большим, более мощным функционалом, который отсутствует у *Excel*.

В данном отчете будет представлена версия *Excel* 2013 (рисунок 4).

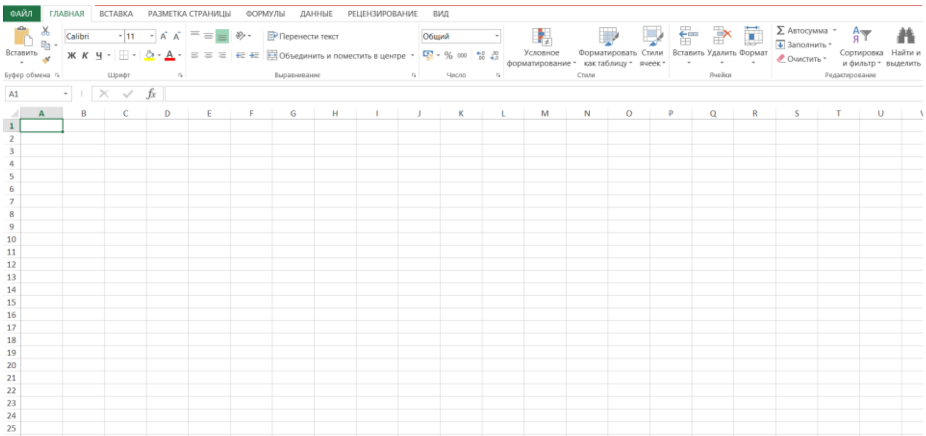

Рисунок 4 – Рабочее пространство *Excel* 2013

Лента, являющаяся основным рабочим элементом, содержит все команды, необходимы для выполнения распространенных задач.

Панель быстрого доступа позволяет получить доступ к основным командам (рисунок 5).

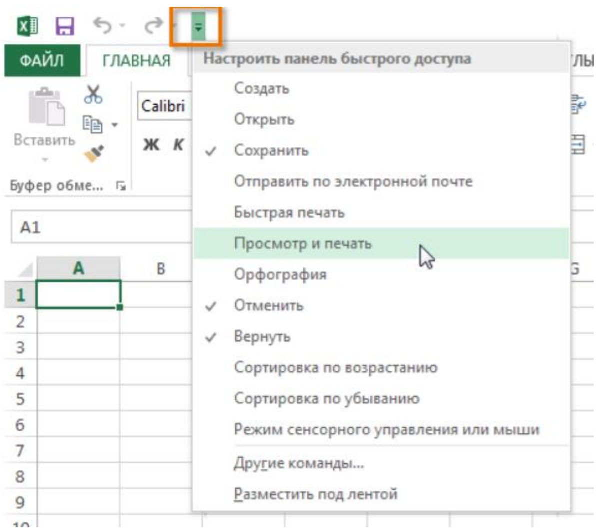

Рисунок 5 – Панель быстрого доступа

Многие группы содержат блок различных команд (рисунок 6).

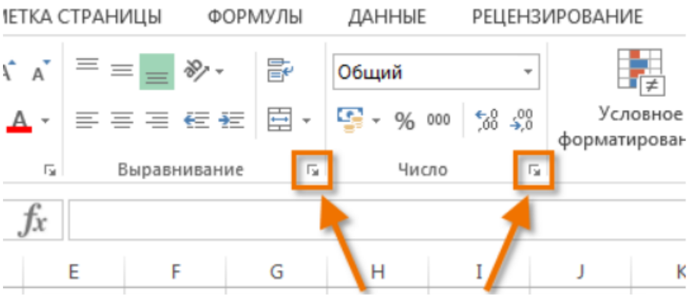

Рисунок 6 – Блок команд

В поле «Имя» отображает адрес или имя выбранной ячейки (рисунок 7).

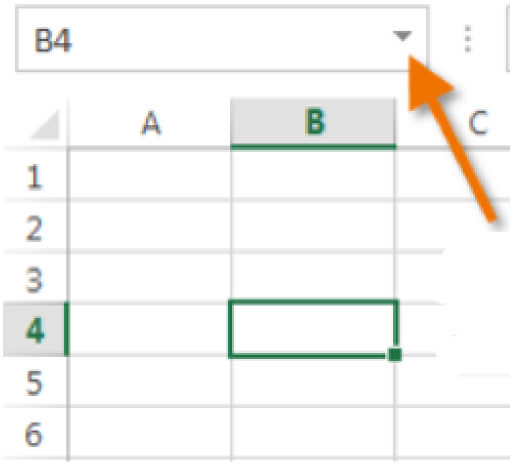

Рисунок 7 – Поле отображения адреса ячейки

В строку формул можно вводить данные, формулы и функции, которые также появятся в выбранной ячейке. (рисунок 8).

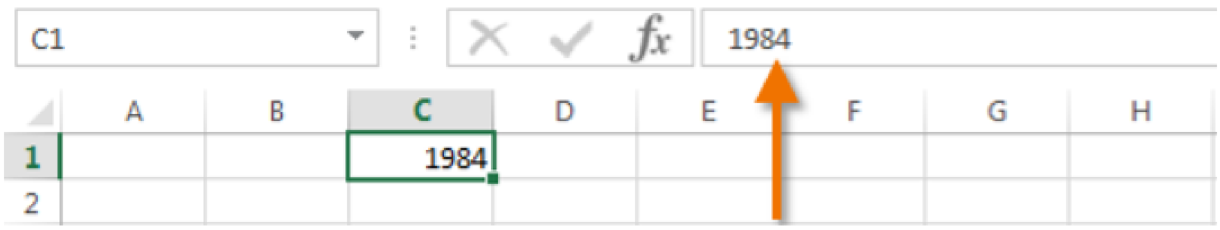

Рисунок 8 – Строка формул

Рассмотрим основные функциональные возможности данной системы автоматизации математических расчетов в таблице 3.

Таблица 3 – Математические функции и формулы

Значение	Оператор	Пример написания в строке	Формула
Сложение	+	=A1+A2	=СУММ(A1;A2)
Вычитание	-	=A1-A2	-
Деление	/	=A1/A2	-
Умножение	*	=A1*A2	-
Возведение в степень	^	=A2^3	=СТЕПЕНЬ(A2;3)
		=A2^(A1+3)	=СТЕПЕНЬ(A2;A1+3)
Логарифм вида $\log_n m$	-	-	*=LOG(m;n)*
Десятичный логарифм	-	-	*=LOG10(m)*
Корень *n*-ой степени	-	=A2^(1/3)	=СТЕПЕНЬ(A2;1/3)
Косинус числа *x*	-	-	*=COS(A2)*
Синус числа *x*	-	-	*=SIN(A2)*

Таким образом, представленные выше программы могут использоваться в качестве средств автоматизации расчетов параметров электронных устройств, узлов и блоков. *MATLAB* и *Excel* имеют как преимущества, так и недостатки, поэтому проектировщик сам решает, какую программу необходимо использовать для решения различных задач.

3. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В СИСТЕМЕ *MATLAB*

Для решения систем дифференциальных уравнений существует несколько встроенных процедур. Рассмотрим применение процедуры *ode45*. Вызов процедуры:

[t,r]=ode45(@DiffEquationFunction,[Tstart,Tfinish], StartVector).
Отметим следующее, что процедура *ode45* может решить систему уравнений следующего вида: $\frac{d}{dt}F(t) = G(t, x_1, x_2,..., x_n)$,

где *F(t)* - есть векторная функция $(\dot{x}_1, \dot{x}_2 ... \dot{x}_n)$.

Рассмотрим процедуру решения системы дифференциального уравнения, представленного в виде функции *threepoint(t,x).*

```
function f=threepoint(t,x)
M1=50; M2=0; C1x=5; C1y=0; C2x=0; C2y=10;
f=[x(3);x(4);...
 -M1*(x(1)-C1x)/(sqrt((x(1)-C1x)^2+(x(2)-C1y)^2))^3-...
 M2*(x(1)-C2x)/(sqrt((x(1)-C2x)^2+(x(2)-C2y)^2))^3;...
 -M1*(x(2)-C1y)/(sqrt((x(1)-C1x)^2+(x(2)-C1y)^2))^3-...
 M2*(x(2)-C2y)/(sqrt((x(1)-C2x)^2+(x(2)-C2y)^2))^3];
```

Решим систему дифференциальных уравнений, вызвав процедуру *ode45* из файла-функции *dynpoint.m.*

```
function dynpoint()
[t,h]=ode45(@threepoint,[0,1000],[0,0,0,4.3]);
x=h(:,1);
y=h(:,2);
x1=5; y1=0; x2=0; y2=100;
plot(x,y,'b-',x1,y1,'r+',x2,y2,'r*').
```

Основой для решения обыкновенных дифференциальных уравнений первого порядка является задача Коши:

$$\frac{dy}{dx} = f(x, y)$$

с одной зависимой переменной *y(x).*

Пример. Дано дифференциальное уравнение

x'(t) + 2x(t) = sin(t),
x(0) = 0.

После запуска системы *MATLAB* нажмем кнопку *Simulink*, а затем в открывшемся окне кнопку *Create a new Model*. В открывшемся файле создадим схему решения уравнения, перетаскивая при нажатой левой кнопки мыши необходимые блоки из окна *Simulink Library Brouser*.

Для построения схемы решения уравнения в *Simulink* используется блок *Integrator* (класс *Continuos*) (рисунок 9). На его вход подается производная, а на выходе получают величину *x*. Блоки *Sum* (Сумматор) и *Gain* (Усилитель) (класс *Math*) необходимы для формирования значения *x*' в соотствии с ОДУ. Для получения сигнала *sin(t)* используется блок *Sine Wave* (класс *Sources*), в котором необходимо провести установки, соответствующие задаче, открыв блок двойным щелчком мыши или выбрав опцию *Block Parameters* при нажатой правой кнопке мыши. Полученное значение *x(t)* подается на вход блока *Scope*. При открытии данного блока появляется график решения. Установить масштабы осей, соответствующие полученному решению можно, нажав кнопку *Autoscale*.

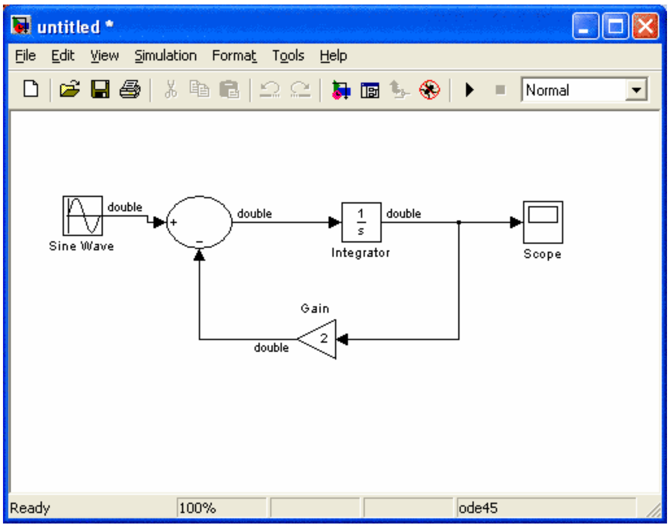

Рисунок 9 – Схема *Simulink* для решения дифференциального уравнения (2.1)

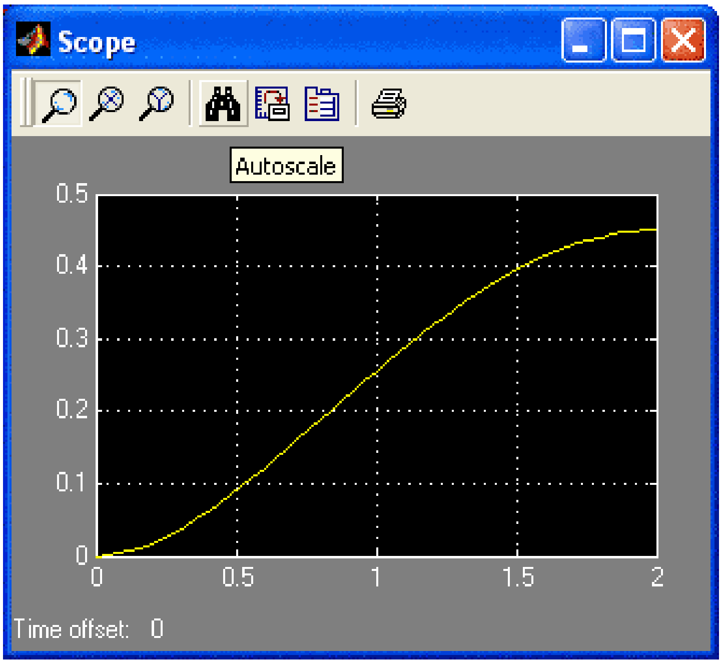

Рисунок 10 – График решения дифференциального уравнения

Для проверки найденного решения в окне *Command Window* создадим *M* - файл для решения задачи (*File* ==> *New* ==> *M-file*). В открывшемся окне (рисунок 11) создадим функцию решения задачи, которую сохраним в текущей директории под именем *f.m* (указанное имя система предлагает по умолчанию).

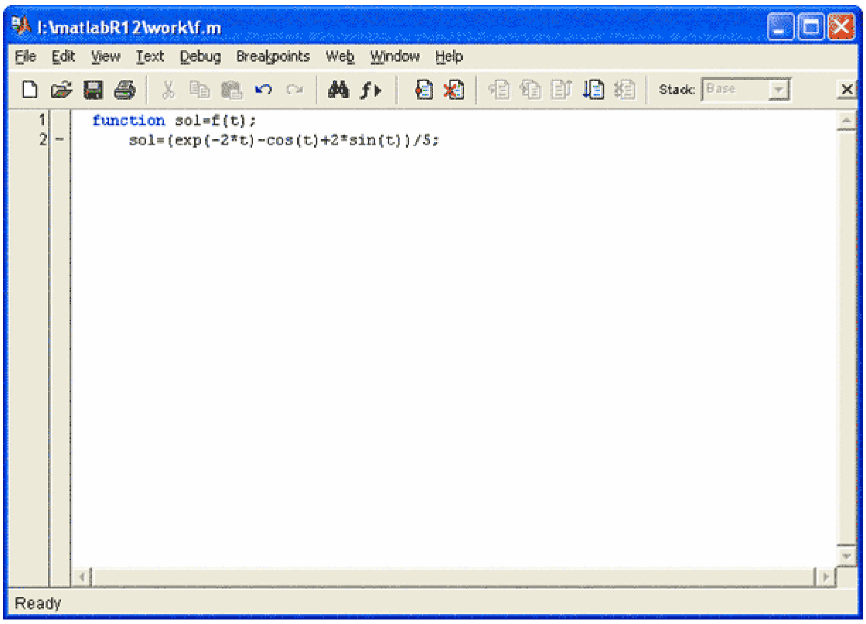

Рисунок 11 – *M* - файл для проверки решения дифференциального уравнения

После этого в командном окне наберем текст:

>> t=(0:0.1:2);
>> y=f(t);
>> plot(t,y)
>> grid on

После выполнения команд открывается окно с графиком функции (рисунок 12). Очевидно, что два полученных графика идентичны.

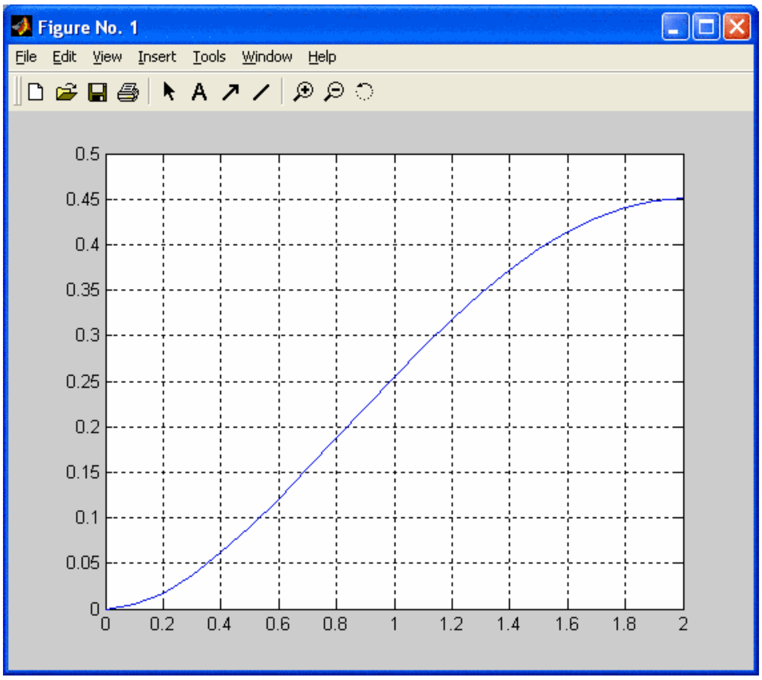

Рисунок 12 – График функции проверки решения дифференциального уравнения

Задания.

1. Решить систему дифференциальных уравнений, используя функцию *ode45:*

$$\begin{cases} y' = \dfrac{z}{x}, \\ z' = \dfrac{2z^2}{x(y-1)} + \dfrac{z}{x}, \qquad \text{на } [1,2]. \\ y(1) = 0, \quad z(1) = \dfrac{1}{3} \end{cases}$$

2. Решите системы дифференциальных уравнений в системе *Simulink.*

2.1 Для решения системы дифференциальных уравнений

$$\begin{cases} \dfrac{dx}{dt} = x(\alpha - \beta y - \gamma x), \\ \dfrac{dy}{dt} = -y(\delta - \varepsilon x) \end{cases}$$

при заданных значениях параметров (α = 0.1; β = 0.05; γ = 0.03; δ = 0.2; ε = 0.15) соберите блок-схему задачи в системе *Simulink* в соответствии с рисунком 13.

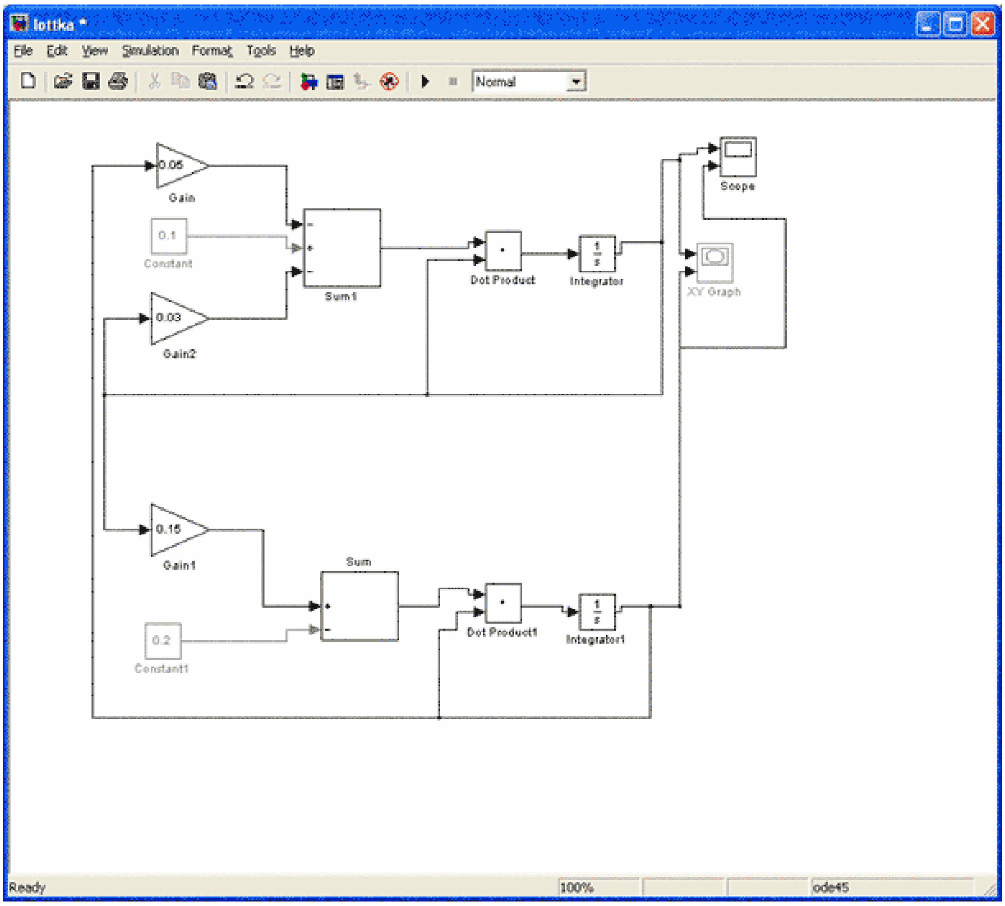

Рисунок 13 – Блок-схема решения системы дифференциальных уравнений

2.2. Раскрыв блок интегратора, задайте начальные значения: y_1=2, y_2=0.01.

2.3 После окончания моделирования, раскрыв блоки *Scope* и *XY-Graph*, изучите полученные графики.

2.4 Создайте в *MATLAB M* - файл для задания правой части системы ОДУ. Для конкретного набора начальных значений систему можно решить, используя метод Рунге - Кутты 4-го порядка (встроенная функция *ode45*):

> [T, Y] = ode45('vlm',[0 164],[2 0.01]).

Параметры функции *ode45*: имя *M* - файла, диапазон изменения независимой переменной, начальные значения.

2.5 Постройте фазовый портрет, использовать функцию *plot*:

> plot(Y(:,1),Y(:,2));

>axis[0 4 0 4].

2.6 Постройте схему решения в *Simulink* и получите график решения задачи в соответствии с выбранным вариантом (таблица 4). Сравнить с решением задач в *MATLAB* с помощью функции ode45 и в *Simulink.*

Таблица 4 – Варианты заданий

№ варианта	Задача
1	**2**
1	$\begin{cases} y'=\dfrac{z}{x}, \\ z'=\dfrac{2z^2}{x(y-1)}+\dfrac{z}{x}, \\ y(1)=0, \quad z(1)=\dfrac{1}{3} \end{cases}$ на $[1,2]$.
2	$\begin{cases} y'=(z-y)x, \\ z'=(z+y)x, \\ y(0)=1, \quad z(0)=1 \end{cases}$ на $[0,1]$.
3	$\begin{cases} y'=\cos(y+2z)+2, \\ z'=\dfrac{2}{x+2y^2}+x+1, \\ y(0)=1, \quad z(0)=0.05 \end{cases}$ на $[0, 0.3]$.
4	$\begin{cases} y'=e^{-(x^2+z^2)}+2x, \\ z'=2y^2+z, \\ y(0)=0.5, \quad z(0)=1 \end{cases}$ на $[0, 0.3]$.

1	**2**
5	$y'' = -\frac{y'}{x} + \frac{y}{x^2} + 1$, $y(3) = 6$, $y'(3) = 3$.
6	$y'' - y = e^x$, $y(0) = 0$, $y'(0) = 0.5$ на [0,1].
7	$y'' - 2y' = x^2 - 1$, $y(1) = -1/6$, $y'(1) = -3/4$ на [1,2].
8	$y'' - 2y' = 3e^x$, $y(0.3) = 1.415$, $y'(0) = 5.83$ на [0.3, 0.6].

4. РАСЧЕТ МАССЫ ЦЕЛЕВОЙ АППАРАТУРЫ И ЭНЕРГЕТИЧЕСКОЙ МОЩНОСТИ

Цель работы: получить математические модели для расчета массы целевой аппаратуры КА и энергетической мощности ИСЗ.

Исходные данные: исходные данные представлены в таблицах 2 и 3.

Задания:

1. используя данные статистических исследований разработать математическую модель зависимости массы целевой аппаратуры от энергетической мощности бортового радиокомплекса;
2. используя данные статистических исследований разработать математическую модель зависимости энергетической мощности СОС, СТР и других служебных систем от энергетической мощности целевой аппаратуры;
3. результаты представить в аналитической и графических формах, используя программные средства;
4. сформулировать выводы.

Таблица 2 – Зависимость массы целевой аппаратуры от энергетической мощности бортового радиокомплекса

Характеристики	Название спутника			
	Инмарсат -2	Инмарсат -3	*TV-SAT*	Астра -1А
Масса ИСЗ, кг	860	1100	1025	1820
Масса целевой аппаратуры, кг	130	190	220	190
Энергетическая мощность целевой аппаратуры, Вт	1200	2300	2500	2000
Срок активного существования, лет	10	13	9	10

Таблица 3 – Зависимость энергетической мощности СОС, СТР и других служебных систем от энергетической мощности целевой аппаратуры

Характеристики	Название спутника				
	Интелсат-5а	Интелсат-6	Интелсат-*7*	*TV-SAT*	Астра -1А
1	**2**	**3**	**4**	**5**	**6**
Масса ИСЗ, кг	1142	4170	3610	1025	1820
Масса целевой аппаратуры, кг	90	662	446	220	190

1	2	3	4	5	6
Энергетическая мощность ИСЗ, Вт	1453	2252	3968	4500	3600
Энергетическая мощность целевой аппаратуры, Вт	978	1531	2580	2500	2000
Срок активного существования, лет	7	13	15	9	10

5. РЕАКТИВНОЕ ДВИЖЕНИЕ. ВЕРТИКАЛЬНЫЙ ВЗЛЕТ ОДНОСТУПЕНЧАТОЙ РАКЕТЫ

Реактивное движение. Ракета - летательный аппарат, двигающийся вследствие выброса части собственной массы. Такой вид движения является реактивным.

Для характеристики реактивного движения необходимо рассмотреть его закономерности. Пусть в замкнутой системе, в отсутствии силовых полей от неподвижного тела массой m, которое будет рассматриваться как материальная точка со скоростью u отделяется частица массой dm. В силу закона сохранения импульса тело также приходит в движение:

$$\vec{u}dm = (m - dm)d\vec{V},$$

где V – скорость тела.

Пренебрегая бесконечно малыми второго порядка, уравнение изменения импульса будет выглядеть так:

$$d\vec{P} \approx md\vec{V} - \vec{u}dm = 0.$$

Согласно 2-му закону Ньютона:

$$\frac{d\vec{P}}{dt} = \vec{F}.$$

Тогда уравнение движения, в том числе и в неподвижной системе координат примет вид:

$$m\frac{d\vec{V}}{dt} = \vec{F} + \vec{u}\frac{dm}{dt}.$$

Принцип относительности Галилея и закон сложения скоростей позволяют под V понимать скорость тела относительно неподвижной системы координат. Движение материальной точки осуществляется таким образом, как будто на нее действует не только внешнее силовое поле, но и добавочная реактивная сила T (тяговая сила).

$$\vec{T} = \vec{u}\frac{dm}{dt}.$$

Так как $\frac{dm}{dt} < 0$, то тяговая сила направлена против скорости отделяющейся частицы.

Тогда уравнение динамики переменной массы примет окончательны вид:

$$m\frac{d\vec{V}}{dt} = \vec{F} + \vec{T}.$$

На ракету вблизи поверхности Земли действуют различные силы. В данном случае используются сила всемирного тяготения и сила лобового сопротивления. Если расположить начало координат на поверхности Земли, направив ось у вертикально вверх, то сила всемирного тяготения, направленная к центру Земли будет иметь вид:

$$F_g = -G\frac{Mm}{(R+y)^2},$$

где M и R – масса и радиус Земли, G – постоянная всемирного тяготения, а m – масса ракеты.

На поверхности сила всемирного тяготения проявляется как сила тяжести, тогда

$$mg = G\frac{Mm}{R^2}, F_g = -\frac{gR^2m}{(R+y)^2},$$

где g – ускорение свободного падения на поверхности Земли.

Силу лобового сопротивления со стороны атмосферы Земли будем полагать квадратичной относительно скорости и направленной против скорости движения. Тогда:

$$\overrightarrow{F_D} = -\beta|\vec{V}|\vec{V}, \beta = \frac{CS\rho}{2},$$

где C – коэффициент лобового сопротивления, определяемый формой тела, S – характерная площадь тела, ρ – плотности среды.

При заданных ограничениях уравнение динамики тела переменной массы имеет вид:

$$m(t)\frac{d\vec{V}}{dt} = \overrightarrow{F_g} + \overrightarrow{F_D} + \vec{T}.$$

Вертикальный взлет одноступенчатой ракеты. Если разделить уравнение движения на массу, получится:

$$\frac{dV}{dt} = -\frac{u}{m}\frac{dm}{dt} - \frac{\beta}{m}V^2 - \frac{gR^2}{(R+y)^2}, \frac{dy}{dt} = V.$$

Если предположить, что ракета, имеющая стартовую массу m_0, в процессе полета сжигает топливо равномерно. Если время работы двигателя t_0, а m_1 – масса ракеты без топлива (сухая масса), то функция изменения массы ракеты от времени имеет вид:

$$m(t)\begin{cases} m_0 - \alpha t\,, t \le t_0 \\ m_1\,, t > t_0 \end{cases} t_0 = \frac{m_0 - m_1}{\alpha},$$

где α – расход топлива в единицу времени.

Тогда, тяга в единицу времени – удельная тяга,

$$\tau(t) = \frac{T}{m} = -\frac{m'}{m}u = \begin{cases} \dfrac{\alpha u}{m_0 - \alpha t}\,, t \le t_0 \\ 0\,, t > t_0 \end{cases}.$$

Уравнение движения ракеты было сведено к дифференциальному уравнению второго порядка:

$$y'' = \tau(t) - g\left(1 + \frac{y}{R}\right)^{-2} - \gamma(t,y)(y')^2,$$

где коэффициент $\gamma(t, y) = \frac{\beta}{m} = CS\rho(y)/2m(t)$ является функцией времени и высоты. Зависимость плотности воздуха от высоты определяется задаваемой моделью атмосферы Земли.

Цель работы: познакомиться с методикой моделирования вертикального взлета одноступенчатой ракеты с использованием программных средств.

Исходные данные. Первая в мире баллистическая ракета «Фау-2» разработана немецким конструктором Вернером фон Брауном (рисунок 14). Во время второй мировой войны использовалась вермахтом для обстрела территории Великобритании. Пуски начались в 1942 году. В 1944 году «Фау-2» во время первого в истории суборбитального полета достигла высоты 188 километров. Ракета стартовала вертикально, была одноступенчатой и имела жидкостный реактивный двигатель. В качестве двухкомпонентного топлива (горючее – окислитель) применялся этиловый спирт и жидкий кислород.

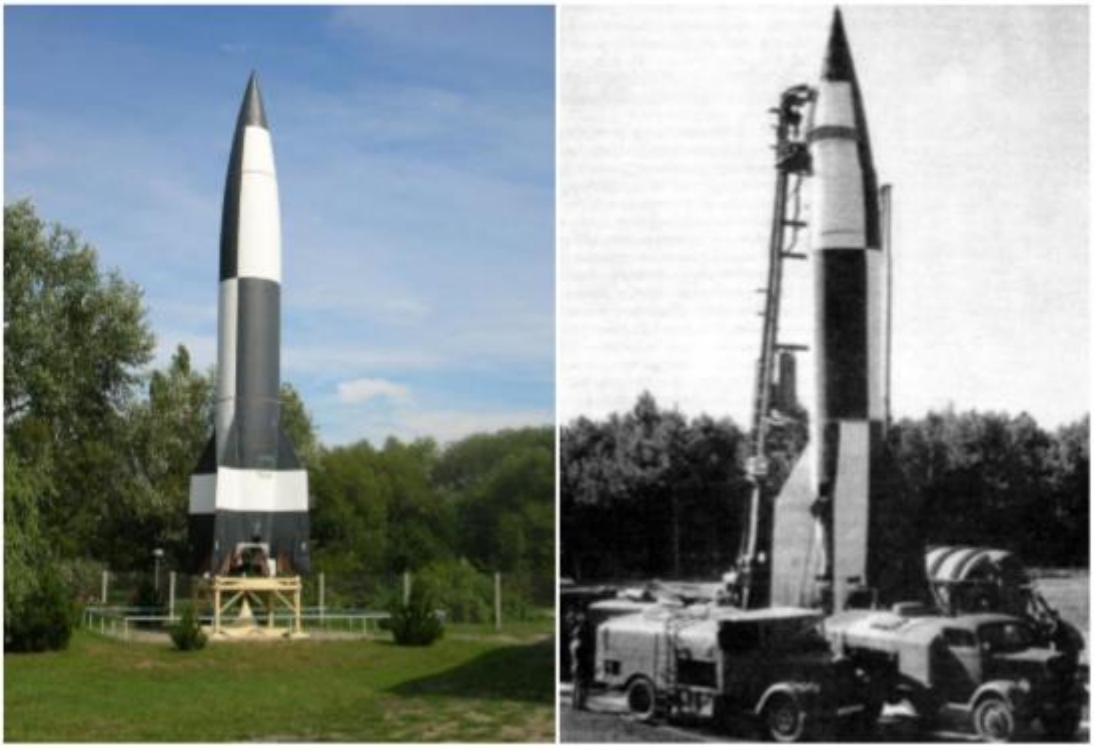

Рисунок 14 - Баллистическая ракета «Фау-2»

Основные характеристики ракету «Фау-2» приведены в таблице 4.

Таблица 4 – Основные параметры ракеты «Фау-2»

Масса стартовая, кг	12500
Масса незаправленной ракеты с боевой частью, кг	4000
Время работы двигателя, с	65
Скорость истечения топлива, м/с	2050
Скорость ракеты в конце работы двигателя, м/с	1450
Диаметр корпуса, мм	1650
Общая длина ракеты, мм	14030

Были рассчитаны средний расход топлива в единицу времени и доля топлива в стартовой массе ракеты:

$$\alpha=131 \text{ кг/с}, \; k\approx 0{,}68.$$

Уравнение движения ракеты «Фау-2» с учетом замены переменной и сведения дифференциального уравнения 2-го порядка к обыкновенному дифференциальному уравнению первого порядка, разрешённому относительно производной. Пусть

$$z=\begin{pmatrix} y \\ V_y \end{pmatrix}, \text{тогда } \frac{dz}{dy}=f(y), \text{ а } \begin{cases} z_1(0)=0 \\ z_2(0)=0 \end{cases}.$$

Численное решение данной системы построим, используя солвер *MATLAB*. Листинг программы *fau2.m*, реализующий это решение, приводится в приложении А. Сделаем несколько замечаний по тексту программы.

Плотность атмосферы Земли рассчитывается с использованием барометрического уравнения. Для наших целей точность такой модели вполне достаточна.

Направление силы лобового сопротивления, противоположное скорости, определяется посредством функции *sign()* из библиотеки математических функций *MATLAB*.

Солверы *MATLAB* позволяют отслеживать события и обрабатывать их в процессе решения. Функция *odeset* определяет структуру *options*. Опция *Events*, установленная в структуре *options*, позволяет зафиксировать момент, когда какая-либо из указанных в функции событий переменных принимает значение равное нулю. Там же определяется ссылка на функцию событий – *@eventsM*.

В этом случае набор из трех переменных - [*value,isterminal,direction*], передается солверу функцией события:

value – содержит вектор величин, зависящих от времени и состояния, равенство нулю которых регистрируется как событие;

isterminal – вектор той же длины, состоящий из значений 0 и 1, компоненты которого отвечают за остановку процесса численного интегрирования. Численное интегрирование прекращается, если это значение равно 1, а для соответствующей функции наступило событие. В противном случае, когда значение равно 0, численное интегрирование продолжается;

direction – вектор той же длины, состоящий из значений 0, 1 и -1, компоненты которого отвечают за направление изменения величин вектора *value*. При значении 1 событие фиксируется, только если соответствующая функция возрастает, при -1 событие фиксируется, только если функция убывает, при 0 событие фиксируется при любом поведении функции.

Солвер в этом случае наряду со стандартными выходными параметрами (вектором значений моментов времени и матрицей значений переменных в эти моменты времени) имеет дополнительные выходные параметры.

te – вектор моментов времени, в которые зафиксированы изменения знака величин *value* в направлении, заданном вектором *direction*; *ye* – матрица значений переменных в эти моменты времени; *ie* – вектор, компоненты которого содержат номера компонент вектора *value*, которые меняют знак в моменты времени *te*.

Приведенный в тексте обработчик событий настроен на регистрацию событий $y=0$ и $V=0$. Событием, условие $y=0$, считается лишь при убывающей функции $y(t)$, а событие $V=0$ – в любом случае. При наступлении события $V=0$ интегрирование продолжается, а при наступлении события $y=0$ интегрирование прекращается.

Задания:

1. изучить листинг программы и создать аналогичный *m*-файл *MATLAB*;

2. запустить файл и сопоставить команды в листинге с полученными результатами;

3. построить графики зависимостей скорости ракеты и ее высоты над поверхностью Земли от времени, проанализировать результаты и описать процесс полета;

4. определить дополнительные параметры, влияющие на скорость ракеты и ее высоту над поверхностью Земли;

5. варьируя дополнительные параметры построить многопараметрические графики, пояснить характер влияния параметров;

6. Заполнить протокол работы программы в таблице 5.

7. Нарисовать алгоритм программы.

Таблица 5 – Протокол работы программы

Максимальная скорость				км/с	
Максимальная высота				км	
Время полета				с	
Событие n=0	t=0.000000 с	y=	км	V=	км/с
Событие n=1	t=232.250480 с	y=	км	V=	км/с
Событие n=2	t=424.462902 с	y=	км	V=	км/с

6. ПОЛЕТ МНОГОСТУПЕНЧАТОЙ БАЛЛИСТИЧЕСКОЙ РАКЕТЫ

Пусть ракета движется в пустом пространстве в отсутствии внешних сил. Тогда уравнение движения при постоянной скорости истечения газа имеет вид

$$m\frac{dV}{dt} = u\frac{dm}{dt}.$$

Интегрируя, получаем, что предельная скорость ракеты определяется скоростью истечения и соотношением масс, стартовой и сухой

$$V_{max} = u * \ln\frac{m_0}{m_1}.$$

Эта формула получена Циолковским и носит его имя. Поскольку для «Фау-2» $\frac{m_0}{m_1} = 3$ то формула Циолковского дает значение $V_{max} \approx 2336$ м/с.

Формула Константина Эдуардовича Циолковского выражает максимальную скорость летательного аппарата, которой он достигает во время полета при реактивном движении.

Эта формула выражает скорость ракеты, переданную газами от сожженного топлива. Уравнение Мещерского и формула Циолковского неразрывно связаны - уравнение Мещерского описывает массу материальной точки, которая изменяется со временем, в то время как при реактивном движении ракеты постоянно идет уменьшение ее массы из-за сгорания топлива. Изменение скорости при изменяющейся массе (уменьшающейся в нашем случае) движущегося тела - вот что подразумевает под собой реактивное движение. Формула Циолковского основывается именно на нем.

Для решения ряда задач теоретической механики в области реактивного движения используют уравнение Мещерского (основное уравнение материальной точки переменной массы) и формулу Циолковского (формула конечной скорости летательного аппарата), которые называются основными соотношениям теории реактивного движения.

Основой при проектировании и планировании в области космических полетов является именно формула Циолковского, вывод которой стал настоящим прорывом для освоения космоса.

Для достижения космических скоростей Циолковский предложил делать ракеты многоступенчатыми. В простейшей реализации многоступенчатая ракета представляет собой несколько размещенных друг на друге ракет (ступеней) (рисунок 15). Первая ступень, полностью использовав горючее, отделяется и падает на Землю. Затем начинают работать двигатели второй ступени, затем – третьей, и т.д. На орбиту выводится лишь малая часть стартовой массы – полезная нагрузка.

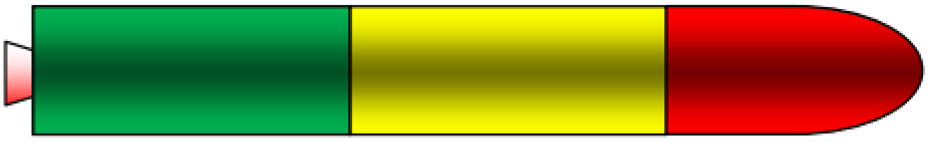

Рисунок 15 – Многоступенчатая баллистическая ракета

Запишем уравнение движения многоступенчатой ракеты. Для i-ой ступени оно совпадает с уравнением движения одноступенчатой ракеты

$$m_i \frac{d\overrightarrow{V_i}}{dt} = \overrightarrow{T_i} + \overrightarrow{F_{gi}} + \overrightarrow{F_{Di}}.$$

Цель работы: познакомиться с методикой моделирования многоступенчатой баллистической ракеты с использованием программных средств.

Исходные данные. Рассмотрим трехступенчатую ракету, основные параметры которой соответствуют «Фау-2». Стартовый вес – 12500 кг, расход топлива каждой ступени – $\alpha \approx 131$ кг/с, доля топлива каждой ступени - $\kappa \approx 0{,}68$, полезная нагрузка равна половине сухого веса третьей ступени.

Масса ракеты есть сумма масс ступеней, запишем $m_0 = m_1 + m_2 + m_3$. Тогда время выгорания топлива i-ой ступени

$$t_i = \frac{\kappa m_i}{\alpha},$$

а зависимость массы ракеты от времени имеет вид

$$m(t) = \begin{cases} m_1 + m_2 + m_3 - \alpha t & , t \le t_1 \\ m_2 + m_3 - \alpha t \ , & t_1 < t \le t_1 + t_2 \\ m_3 - \alpha t & , t_1 + t_2 < t \le t_1 + t_2 + t_3 \\ m_3 \dfrac{(1-\kappa)}{2}, & t > t_1 + t_2 + t_3 \end{cases}.$$

Распределим массу ракеты по ступеням следующим образом: 8000 кг, 4000 кг и 500 кг. Тогда масса полезной нагрузки – 160 кг. Такая нагрузка, выведенная на орбиту, относится к классу малых космических аппаратов. На рисунке 16 приведена зависимость массы такой ракеты от времени.

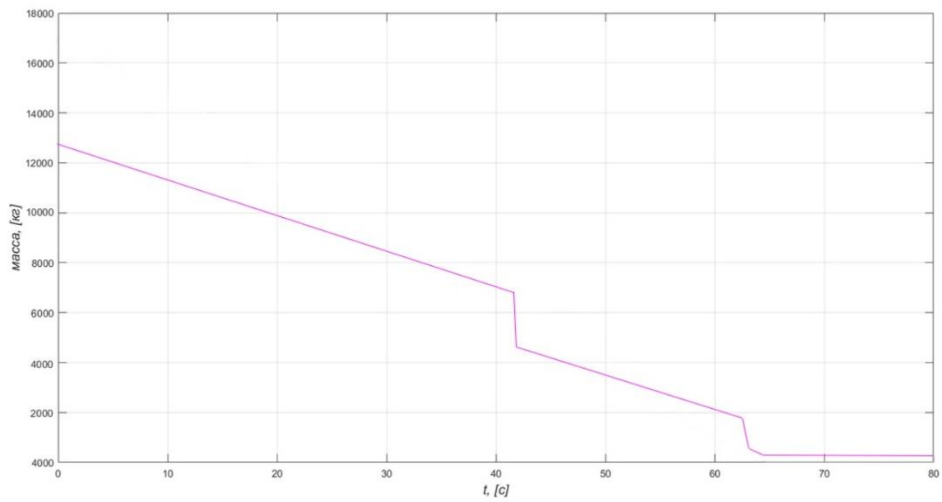

Рисунок 16 – Зависимость массы ракеты от времени

Модульный принцип построения программы расчета полета одноступенчатой ракеты позволяет фактически изменить только программу расчета массы и определить глобальные переменные (Приложение Б).

Задания:

1. модернизировать листинг программы из приложения А с целью моделирования полета многоступенчатой баллистической ракеты;

2. запустить файл и сопоставить команды в листинге с полученными результатами, описать назначение каждой команды;

3. построить график зависимости массы ракеты от времени, проанализировать результаты;

4. определить дополнительные параметры, влияющие на скорость ракеты;

5. получить протоколы работы программы для двух видов топлива (этиловый спирт – кислород, гептил – кислород);

6. Заполнить протоколы работы программы в соответствии с таблицей 5 раздела 5.

7. Нарисовать алгоритм программы.

7. МОДЕЛИРОВАНИЕ СОЛНЕЧНОЙ СИСТЕМЫ

Гелиоцентрическая модель солнечной системы, основанная на представлении о том, что Земля и планеты вращаются вокруг Солнца, окончательно сформулирована польским астрономом Николаем Коперником в книге «О вращении небесных сфер», 1543 г. До него идеи гелиоцентризма встречались в работах целого ряда греческих, арабских и индийских ученых. Иоганн Кеплер в работе «Новая астрономия», 1609 г., сформулировал законы планетных движений, определив форму орбит и установив математическую связь между их геометрическими параметрами и периодами планетных движений. Его законы были получены как итог большого количества точных астрономических наблюдений. Приведем три закона Кеплера:

1. Планеты движутся по эллипсам, в одном из фокусов которого находится Солнце.
2. Радиус-вектор планеты за равные промежутки времени описывает одинаковые площади.
3. Квадраты периодов обращения планет относятся как кубы больших полуосей их орбит.

Параметры орбит планет солнечной системы приведены в таблице. Заметим, что современная классификация относит Плутон к карликовым планетам.

Планеты делятся на два вида: внутренние и внешние. Различие связано с тем, что орбиты внутренних планет (Меркурия и Венеры) находятся внутри орбиты Земли, в то время как орбиты внешних находятся вне орбиты Земли. При движении по эллипсу расстояние от планеты до Солнца меняется, наиболее удалена точка орбиты называется афелий, ближайшая к Солнцу – перигелий (рисунок 17).

Предположим, что все планеты движутся в плоскости эклиптики. Тогда в декартовой системе координат параметрические уравнения движения планет имеют вид:

$$x(\xi) = a(cos\xi - e), y(\xi) = a\sqrt{1-e^2}sin\xi, t(\xi) = \frac{T}{2\pi}(\xi - esin\xi), 0 \le \xi \le 2\pi,$$

где a – большая полуось, e – эксцентриситет, T – период.

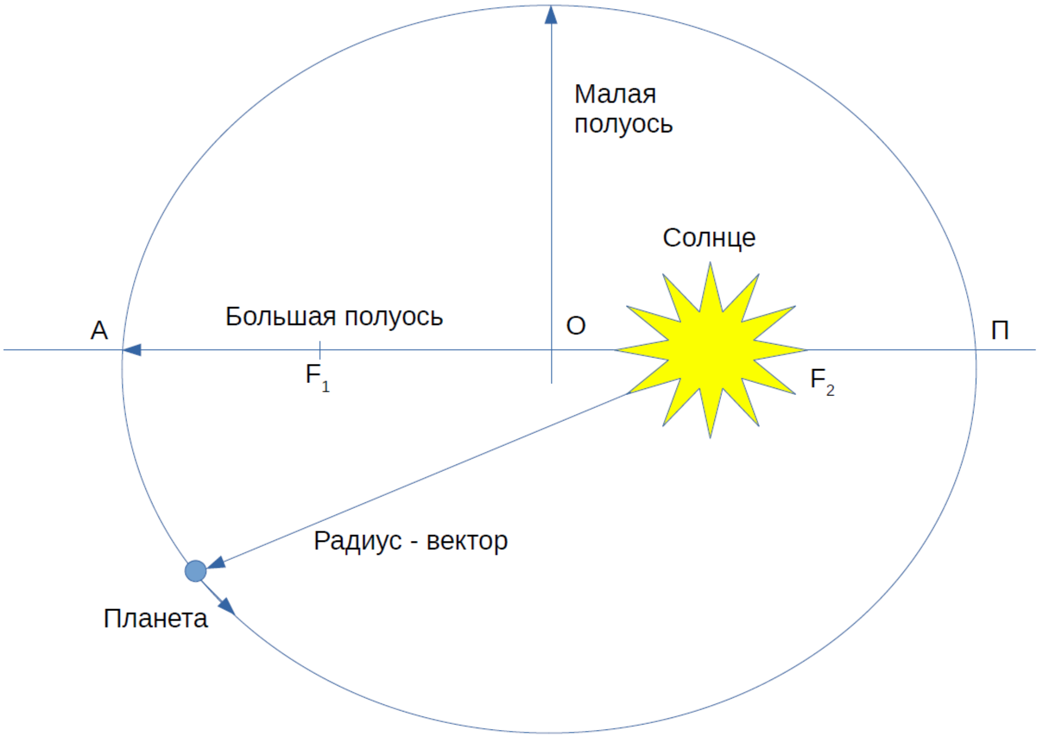

Рисунок 17 – Моделирование Солнечной системы

Цель работы: познакомиться с методикой моделирования солнечной системы с использованием информационных технологий.

Исходные данные. В таблице 6 приведены орбитальные параметры планет. В приложении В представлен листинг программы, позволяющей начертить орбиты планет.

На рисунке 18 приведены результаты расчетов и модель солнечной системы. Как видно из рисунка, любая модель такого рода всегда выполняется с нарушением масштаба.

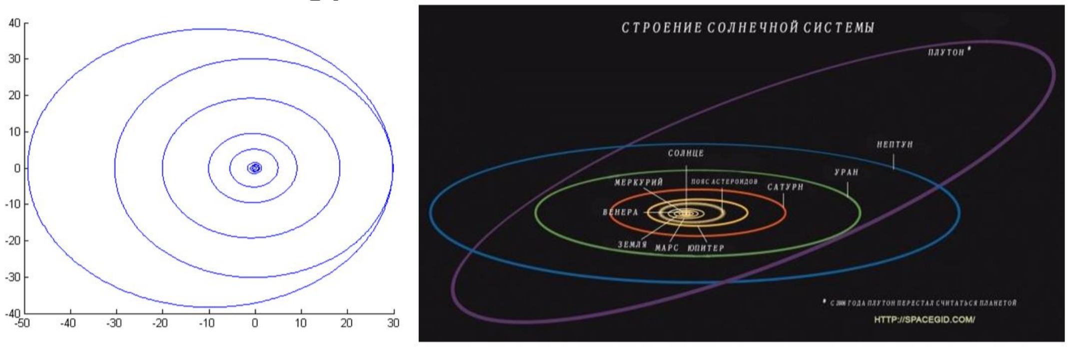

Рисунок 18 - Результаты расчетов и модель солнечной системы

Таблица 6 – Орбитальные параметры планет

Орбитальные параметры планет						
Название	Большая полуось, [а.е .]	Эксцентриситет	Наклон к эклиптике, [град]	Период обращения, [сут]	Наклон оси, [град]	Орбитальная скорость, [км/с]
Меркурий	0.38709831	0.205631752	7.004986389	87.96843362	0.00	47.87
Венера	0.72332982	0.006771882	3.394661944	224.6954354	177.36	35.02
Земля	1.000001018	0.016708617	0	365.24218985	23.45	29.79
Марс	1.523679342	0.09340062	1.849726389	686.92970957	25.19	24.13
Юпитер	5.202603191	0.048494851	1.303269722	4330.595765	3.13	13.06
Сатурн	9.554909596	0.055508622	2.488878056	10746.94044	25.33	9.66
Уран	19.21844606	0.046295899	0.77319611	30588.74035	97.86	6.8
Нептун	30.11038687	0.008988095	1.7699522	59799.90046	28.31	5.44
Плутон	39.5181762	0.245938782	17.12259917	90738.995	122.52	4.74

Задания:

1. изучить листинг программы из приложения В;
2. скопировать и запустить программу для черчения орбит планет, изучить и сопоставить команды в листинге с полученными результатами, описать назначение каждой команды;
3. построить графики орбит планет, присвоить имя планеты каждой орбите;
4. нарисовать алгоритм программы.

8. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Структура системы *MATLAB*.
2. Что такое m-файлы?
3. Как создать, сохранить и вызвать m-файл?
4. Создание векторов в системе *MATLAB*.
5. Создание матриц в системе *MATLAB*. Специальные функции для создания матриц.
6. Перечислите известные вам операторы цикла и ветвления.
7. Графические возможности *MATLAB*.
8. Способ решения линейных уравнений в *MATLAB*.
9. Перечислите функции *MATLAB*, использующиеся для решения систем дифференциальных уравнений.
10. Перечислите типы дифференциальных уравнений и методы их решения?
11. Опишите концепцию решения дифференциальных уравнений в *Simulink*.
12. Назовите назначение элементов схемы на рисунке 9.
13. Какая функция *Matlab* реализует метод Рунге – Кутты?
14. Перечислите основные библиотеки и блоки *Simulink*, используемые при решении математических задач.
15. Назовите цели и задачи моделирования, дайте определение понятиям «модель» и «моделирование».
16. Перечислите этапы моделирования и охарактеризуйте каждый из них.
17. Назовите основные модели процесса проектирования.
18. Охарактеризуйте модель существования КА.
19. Охарактеризуйте модель возможности.
20. Охарактеризуйте модель движения.
21. Охарактеризуйте модель масс.
22. Запишите основное уравнение существования КА и дайте наименование коэффициентам модели.
23. Перечислите основные составляющие конструкции системы энергопитания, массы которых используются при расчете суммарной массы системы энергопитания на основе солнечных батарей.

24. Какие параметры используются при расчете массы панелей солнечных батарей?
25. Какие параметры используются при расчете массы буферных аккумуляторов?
26. Какие параметры используются при расчете массы системы контроля работы СЭП?
27. Какие параметры используются при расчете массы системы терморегулирования?
28. Какие параметры используются при расчете массы системы ориентации и стабилизации?
29. Перечислите основные законы, лежащие в основе уравнения динамики тела переменной массы.
30. Охарактеризуйте математически процесс вертикального взлета одноступенчатой ракеты. Какой порядок дифференциального уравнения движения ракеты?
31. Какие основные характеристики ракеты необходимы для моделирования вертикального взлета одноступенчатой ракеты?
32. От каких параметров зависит скорость ракеты и ее высоты над поверхностью Земли?
33. Нарисуйте и поясните зависимости скорости ракеты и ее высоты над поверхностью Земли от времени.
34. Охарактеризуйте процесс полета многоступенчатой баллистической ракеты. Какие законы и уравнения используются при создании модели полета.
35. Уравнения Мещерского и Циолковского. Идея и конструкция многоступенчатой баллистической ракеты. Первая космическая скорость.
36. Нарисуйте график зависимости массы трехступенчатой ракеты от времени, поясните его.
37. Виды топлива, влияние на основные параметры полета ракеты.
38. Охарактеризуйте модель солнечной системы.
39. Перечислите основные параметры орбит планет солнечной системы.

9. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен быть оформлен в соответствии с требованиями ГОСТ и содержать:

1. Титульный лист;
2. Оглавление;
3. Наименование работы, цель исследований;
4. Листинг программы с пояснениями;
5. Графики функции;
6. Необходимы расчеты и протоколы;
7. Анализ результатов и выводы;
8. Ответы на контрольные вопросы;
9. Перечень литературы, использованной при подготовке и выполнении работы.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

***Основная*:**

1. Никольский В.В. Основы проектирования автоматических космических аппаратов. Учебник. С-Пб.: БГТУ "Военмех", 2007. 230 с.
2. Гущин В. Н. Основы устройства космических аппаратов. Учебник для вузов. М.: Машиностроение, 2003. 272 с.
3. Никольский В.В. Проектирование космических аппаратов. Учебное пособие. С-Пб.: БГТУ "Военмех", 2003. 121 с

***Дополнительная*:**

4. Авдуевский В.С., Успенский Г.Р. Космическая индустрия. - М.: Машиностроение, 1989. 568 с.
5. Алемасов В.Е. и др. Теория ракетных двигателей. М.: Машиностроение, 1986. 533 с.
6. ГОСТ 2.105-95 Единая система конструкторской документации. Общие требования к текстовым документам.
7. Космонавтика. Энциклопедия. Гл. Ред. В.П.Глушко. -М.: Сов. Энциклопедия, 1985.
8. Механика космического полета. Под редакцией В.П.Мишина. –М.: Машиностроение, 1989. 408 с.
9. Никольский В.В. Системное проектирование транспортных космических аппаратов. Учебное пособие. С-Пб.: БГТУ "Военмех", 2001. 101 с.
10. Основы проектирования летательных аппаратов (транспортные системы). Под ред. Мишина В.П. М.: Машиностроение, 1985. 360с.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гущин В.Н. Основы устройства космических аппаратов: Учебник для вузов. – М.: Машиностроение, 2003. – 272 с.

2. Волоцуев, В. В. Введение в проектирование космических аппаратов: учеб. пособие / В.В. Волоцуев, И.С. Ткаченко. – Самара: Изд-во Самарского университета, 2018. – 144 с.: ил.

3. Куренков, В. И. Основы устройства и моделирования целевого функционирования космических аппаратов наблюдения [Текст]: учеб. пособие / В. И. Куренков, В. В. Салмин, Б. А. Абрамов. – Самара: СГАУ, 2006. – 296 с.

4. Туманов А.В. Основы компоновки бортового оборудования космических аппаратов: учебное пособие / А.В Туманов, В.В. Зеленцов, Г.А. Щеглаков // М.: Изд-во МГТУ им. Н.Э. Баумана, 2010. – 344 с.

ПРИЛОЖЕНИЕ А

Скрипт программы для моделирования движения космических объектов в системе

```
function fau2

global g0 StartWeight DryWeight OperatingTime FuelFlowRate Alfa Kappa; g0=9.80665; % ускорение свободного падения, м/с2

RadiusEarth=6371000; % радиус Земли, м

StartWeight=12500; % стартовая масса ракеты, кг

DryWeight=4000; % масса не заправленной топливом ракеты, кг OperatingTime=65; % время работы двигателя, с

FuelFlowRate=2050; % скорость истечения топлива, м/с

DiameterCase=1650; % диаметр корпуса, мм

Cx=0.15; % коэффициент лобового сопротивления
Ss=pi*(DiameterCase/1000)^2/4; % характерная площадь ракеты Alfa=(StartWeight-DryWeight)/OperatingTime; % расход топлива, кг/с

Kappa=1-DryWeight/StartWeight; % доля топлива в стартовой массе ракеты Y0=0; % начальная координата YP0=0; % вертикальная начальная скорость T0=0; % время начала движения TN=10000; % время окончания движения dT=0.5; % шаг по времени

options=odeset('Events',@eventsM); % обработчик события

Gamma=@(t,y) Deceleration(t,y,Ss,Cx);

Shot3=@(t,z) [z(2); PullUnit(t)-g0./(1+z(1)/RadiusEarth).^2-Gamma(t,z(1)).*sign(z(2)).*z(2).^2];

[T,Z,te,ye,ie]=ode23(Shot3,[T0:dT:TN],[Y0 YP0],options); % решение ОДУ

disp (sprintf('Максимальная скорость %f [км/с]',max(Z(:,2)/1000)));

disp (sprintf('Максимальная высота %f [км]',max(Z(:,1)/1000)));

disp (sprintf('Время полета %f [с]',max(T)));

for iCur=1:size(ie,1)

disp (sprintf('Событие n=%d, t=%f с, y=%f км V=%f км/с',ie(iCur),te(iCur),ye(iCur,1)/1000,ye(iCur,2)/1000));

end

figure('Color',[1 1 1]);

iCur=1;
```

```
for jCur=T0:0.1:80
z(iCur)=MassRocket(jCur);
iCur=iCur+1;
end
hL1=plot([T0:0.1:80],z); grid;
ylabel('\itмасса, [кг]','fontsize',14);
xlabel('\itt, [c]','fontsize',14);
set(hL1(1), 'LineWidth', 2, 'Color', 'm');
figure('Color',[1 1 1]); hL1=plot(T,Z(:,1)./1000); grid; % зависимость координаты от времени
ylabel('\ity, [км]','fontsize',14);
xlabel('\itt, [c]','fontsize',14);
set(hL1(1), 'LineWidth', 2, 'Color', 'b');
figure('Color',[1 1 1]);
hL2=plot(T,Z(:,2)/1000); grid; % зависимость скорости от времени
ylabel('\itV, [км/c]','fontsize',14);
xlabel('\itt, [c]','fontsize',14);
set(hL2(1), 'LineWidth', 2, 'Color', 'g');
 end
%-----------------------------------------------------
% расчет удельной тяги ракеты
function pu=PullUnit(t)
global g0 StartWeight DryWeight OperatingTime FuelFlowRate Alfa;
        if (t<=OperatingTime)
         pu=Alfa*FuelFlowRate./MassRocket(t);
        else
                pu=0;
        end
end
%-----------------------------------------------------
% расчет массы одноступенчатой ракеты
```

```
function mCur=MassRocket(t)
global g0 StartWeight DryWeight OperatingTime FuelFlowRate Alfa Kappa;
        if (t<=OperatingTime)
                mCur=(StartWeight-Alfa.*t);
        else
                mCur=DryWeight;
end;
end
%---------------------------------------------
function da=Deceleration(t,y,S,C)
global g0 StartWeight DryWeight OperatingTime FuelFlowRate Alfa; Mm=0.02896; % молярная масса воздуха, кг/м
Rg=8.3144598; % газовая постоянная
T0=288.15; % температура на уровне моря, К
ro_air_0=1.225; % плотность воздуха на уровне моря, кг/м ro_air_y=ro_air_0.*exp(-y.*(Mm*g0/(Rg*T0))); % барометрическая формула da=C*S*ro_air_y./(2*MassRocket(t));
end
%-----------------------------------------------------
function [value,isterminal,direction] = eventsM(t,z) % Обработка события для солвера
value = [z(1), z(2)]; % определяем событие: y = 0 или V = 0
 isterminal = [1, 0]; % при обнаружении события y = 0 остановить счет, при обнаружении события V = 0 продолжить счет
direction = [-1, 0]; % учитывать событие y = 0 при убывании, учитывать событие V = 0 в любом направлении
end
```

ПРИЛОЖЕНИЕ Б

Листинг программы для моделирования полета баллистической ракеты ФАУ-2

```
global NumberStages StartWeightStage1 StartWeightStage2
StartWeightStage3;

StartWeightStage1=8000; % стартовая масса первой ступени,
кг

StartWeightStage2=4000; % стартовая масса второй ступени,
кг

StartWeightStage3=500; % стартовая масса третей ступени, кг

NumberStages=3; % количество ступеней

%-------------------------------------------------------

% расчет массы одноступенчатой и трехступенчатой ракеты

function mCur=MassRocket(t)

global g0 StartWeight DryWeight OperatingTime FuelFlowRate
Alfa Kappa;

global NumberStages StartWeightStage1 StartWeightStage2
StartWeightStage3;

switch NumberStages

case 1

if (t<=OperatingTime)

mCur=(StartWeight-Alfa.*t);

else

mCur=DryWeight;

end;

case 3

t1=Kappa*StartWeightStage1/Alfa;

t2=Kappa*StartWeightStage2/Alfa;

t3=Kappa*StartWeightStage3/Alfa;

if (t<=t1) % работа первой ступени
```

```
mCur=StartWeightStage1+StartWeightStage2+StartWeightStage3-
Alfa.*t;
end;
if (t>t1) % работа второй ступени
mCur=StartWeightStage2+StartWeightStage3-Alfa.*(t-t1);
end;
if (t>t1+t2) % работа третей ступени
mCur=StartWeightStage3-Alfa.*(t-t1-t2);
end;
if (t>=t1+t2+t3) % полезная нагрузка
mCur=StartWeightStage3*(1-Kappa);
end;
end
end
```

ПРИЛОЖЕНИЕ В

Листинг программы, позволяющей начертить орбиты планет

```
Planet= zeros(9,3);
        % Большая полуось (а.е.);   Эксцентриситет; Период обращения (сут)
Planet(1,:)=[0.38709831;   0.205631752;     87.96843362];   %   Меркурий
Planet(2,:)=[0.72332982;   0.006771882;    224.6954354];    %   Венера
Planet(3,:)=[1.000001018;  0.016708617;    365.24218985];   %   Земля
Planet(4,:)=[1.523679342;  0.09340062;     686.92970957];   %   Марс
Planet(5,:)=[5.202603191;  0.048494851;   4330.595765];     %   Юпитер
Planet(6,:)=[9.554909596;  0.055508622;  10746.94044];      %   Сатурн
Planet(7,:)=[19.21844606;  0.046295899;  30588.74035];      %   Уран
Planet(8,:)=[30.11038687;  0.008988095;  59799.90046];      %   Нептун
Planet(9,:)=[39.5181762;   0.245938782;  90738.995];        %   Плутон

Cg= {'m', 'b', 'g', 'r','k'};
% рисуем орбиты планет
figure('Color',[1 1 1]);
ksi=linspace(0,2*pi,500);
hold on;
plot(0,0,'ok');
for iCurP=1:9
    a=Planet(iCurP,1);
    e=Planet(iCurP,2);
    T=Planet(iCurP,3);
    t=(0.5*T/pi)*(ksi-e.*sin(ksi));
    x=a.*(cos(ksi)-e);
    y=a*sqrt(1-e^2).*sin(ksi);
     plot(x,y);
    %plot(x,y,Cg{iCurP-4});
    %pause(2)
end
```