



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»**

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

ОТЧЁТ ПРОЕКТНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ

«Введение в авиационную и ракетно-космическую технику»

«Луна-1»
(тема)

Проверил: _____

Подпись: _____

Выполнили:

студенты группы

М8О-115Б-23

Шаталов М.

Агафонов А.

Верменников М.

Алексеева А.

Москва, 2023

СОДЕРЖАНИЕ:

ВВЕДЕНИЕ:	4
1 ОПИСАНИЕ РЕАЛЬНОЙ МИССИИ	6
1.1 Цели и история запуска аппарата	6
1.2 Причины неудачи реализации реальной миссии	7
1.3 Траектория движения аппарата 5-3 января 1959 года	11
1.4 Технические характеристики	15
1.5 План выполнения работы	16
2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ	17
2.1 Гомановский переход	17
3 ФИЗИЧЕСКАЯ МОДЕЛЬ	22
3.1 Основные формулы	22
3.1.1 Взлёт	23
3.1.2 Посадка	25
4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	26
4.1 Используемые технологии	26
4.2 Логирование	26
4.3 Автопилот	27
4.3.1 Взлет	27
4.3.2 Выполнение манёвра перехода на орбиту Луны	28

4.3.3	Посадка на Луну	29
5	ПОСТРОЕНИЕ, АНАЛИЗ И СРАВНЕНИЕ ГРАФИКОВ ПО KSP И МАТЕМАТИЧЕСКОЙ МОДЕЛИ	31
5.1	Программа математического моделирования движения	31
5.1.1	Описание Математической Модели	31
5.1.2	Используемые технологии	32
5.2	Сравнение графиков	32
5.3	Причины расхождений между графиками	33
5.4	Выводы	34
6	ЗАКЛЮЧЕНИЕ	35
6.1	Итоги	35
6.2	Работа участников проекта	36
	ИСТОЧНИКИ	37
	ПРИЛОЖЕНИЕ А	40
	ПРИЛОЖЕНИЕ В	47
	ПРИЛОЖЕНИЕ С	48
	Часть 1	48
	Часть 2	49
	ПРИЛОЖЕНИЕ D	51

ВВЕДЕНИЕ:

Название проекта: «Луна-1»

Название команды: «Lunar Conquest»

Актуальность проекта: мы решили взять за основу своего проекта запуск и полёт советской автоматической межпланетной станции (АМС) Луна-1, реализованные 2 января 1959 года, так как все участники проекта считают данную космическую миссию наиболее интересной к изучению и реализации.

Цель проекта: изучить и смоделировать миссию по запуску, полёту и приземлению советской автоматической межпланетной станции (АМС) Луна-1. Выделить причины, по которым спутник не смог приземлиться на Луну. Учитывая все прошлые ошибки, сконструировать верную математическую и физическую модель полёта. Провести симуляцию в Kerbal Space Program.

Задачи:

1. Спроектировать советскую автоматическую межпланетную станцию (АМС)
2. Протестировать её.
3. Рассчитать летные характеристики.
4. Смоделировать полет.
5. Выработать необходимые для полёта станции алгоритмы.
6. Составить отчет о проделанной работе.
7. Выполнить миссию.

Состав команды и роли (все студенты из группы М8О-115Б-23):

Фамилия Имя	Роль	Задачи
Шаталов М.	Тимлид, KSP, программист	Разделение и планирование задач, а также управление командной работой. Изучение работы дополнения kRPC и написание программы автопилота.
Агафонов А.	Математическая, физическая модель	Главный составитель математической и физической модели, а также помощник программиста по составлению графиков.
Верменников М.	Симуляция взлёта и посадки в графиках	Программист и составитель графиков и математических моделей
Алексеева В.И.	Собирает информацию, пишет отчёт, видеомонтаж, презентация	Занимается изучением истории миссии, созданием презентации, отвечает за создание отчёта и видеоотчёта по проделанной работе.

1 ОПИСАНИЕ РЕАЛЬНОЙ МИССИИ

1.1 Цели и история запуска аппарата

2 января 1959 года стартовала с космодрома Байконур ракета “Луна” (другое название — “Восток-Л”). Она несла космический аппарат, который должен был быть доставлен на Луну.

“Восток-Л” представлял собой межконтинентальную ракету Р-7 с третьей ступенью. В блоке “Е” находился шар диаметром 1 м 40 см очень похожий на первый спутник земли. Шар нес в себе аппаратуру: магнитометр, детектор микрометеоритов, счетчик космических лучей, а также два металлических вымпела с гербом Советского Союза и надписью “СССР, январь, 1959 г.” Кроме этого, в него поместили килограмм натрия и систему подогрева.

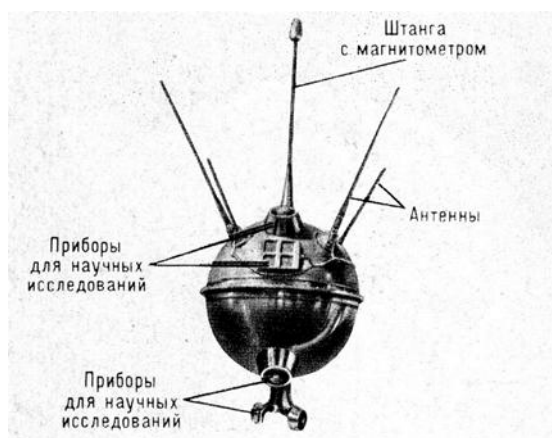


Схема аппарата "Луна-1", рисунок 1, [1]

Межпланетному космическому аппарату не суждено было спуститься на поверхность Луны — он промахнулся на порядок 6000 км. Согласно расчетам советских ученых, “Луна-1” должна была оказаться гораздо ближе к спутнику Земли и под действием его притяжения упасть на лунную поверхность 4 января примерно в 6 утра по московскому времени. Через восемь часов этот факт должен был быть зафиксирован

Лабораторией реактивного движения (г. Пасадена, Калифорния), но американцам удалось лишь поймать слабый сигнал, подтверждающий запуск СССР первой в мире межпланетной станции.

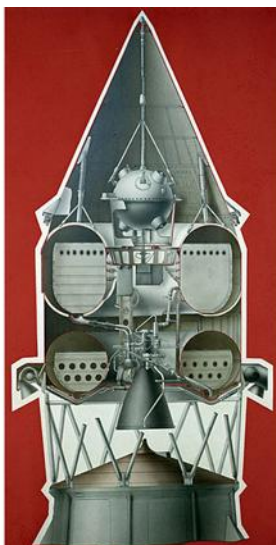


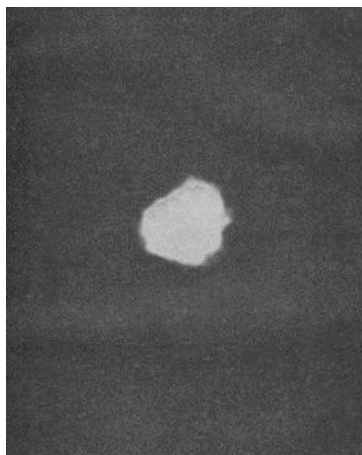
Схема размещения межпланетной станции "Луна-1", рисунок 2, [2]

5 января 1959 года около 10 утра связь с межпланетной станции прервалась. Согласно расчетам 7-8 января “Луна-1” полностью освободилась от сил гравитации Земли и Луны и вышла на околосолнечную орбиту, став новой крохотной планетой, год на которой равен 450 земных суток.

1.2 Причины неудачи реализации реальной миссии

Предположительная причина, почему же ракета так и не приземлилась на Луну: представитель разработчика системы радиуправления, выставя 1 января плоскость антенн РУП-А, ошибся по углу места на 2° , выставив 44° вместо 42° . Из-за ошибки двигатель блока «Е» выключился позже назначенного момента, что и послужило причиной промаха.

Конечная масса блока «Е» вместе с космическим аппаратом «Луна-1» составляла 1472 кг. Полезный груз (361,3 кг) включал научную и измерительную аппаратуру, четыре радиопередатчика и источники электропитания, размещенные в отделяемом аппарате и на блоке «Е».



Эксперимент в космосе: «Искусственная комета», рисунок 3, [2]

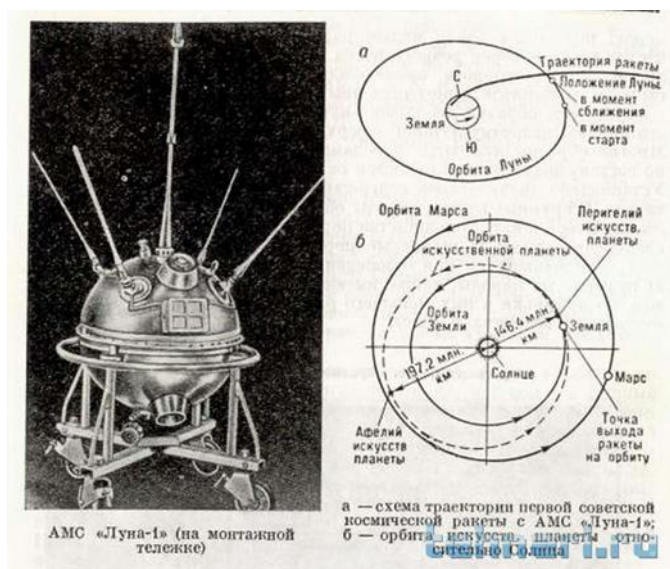
Луна-1	Элементы орбиты
Производитель Союз Советских Социалистических Республик, ОКБ-1	Тип орбиты Гелиоцентрическая
Задачи Изучение Луны, доставка на ее поверхность вымпела СССР	Эксцентриситет 0,14767
Спутник Солнца	Наклонение 0,0010°
Стартовая площадка Союз Советских Социалистических Республик Байконур 1/5	Период обращения 450,0 дней
Ракета-носитель 8К72 №51-6	Апоцентр 1,315 а. е.
Запуск 2 января 1959 16:41:21 UTC	Перицентр 0,9766 а. е.

Характеристики межпланетной станции «Луна-1», рисунок 4, [2]

«Луна-1» несла вымпелы: сферический из стальных пятиугольных элементов с зарядом взрывчатого вещества внутри шара для их разброса и в виде капсулы, заполненной жидкостью, в которой размещались алюминиевые полоски. Оба имели обозначение, указывающее государственную принадлежность аппарата, месяц и год его запуска.

После выключения двигателя произошло отделение контейнера с научной аппаратурой. На высоте 1500 км скорость ракеты относительно центра Земли несколько превышала 10 км/сек, на высоте 100 000 км она равнялась примерно 3,5 км/сек.

4 января в 5:57 по московскому времени «Луна-1» прошла на расстоянии 6000 километров от поверхности Луны и вышла на гелиоцентрическую орбиту. В период наибольшего сближения ракета находилась на небесной сфере выше Луны и несколько правее ее, если смотреть с Северного полушария Земли. В этот момент ракета находилась на небесной сфере между звездами Спика и Весов. После прохождения около Луны космическая ракета продолжала удаляться от Земли, скорость ее относительно центра Земли продолжала убывать, приближаясь к 2,1 км/сек. Гравитационное поле Луны изменило траекторию АМС, но не смогло её притянуть к себе.



Траектория полёта Луна-1, рисунок 5, [3]

7—8 января 1959 года и позже, когда ракета находилась на расстояниях порядка 1 млн. км и более, влияние Земли на ракету стало настолько слабым, что движение ракеты стало в основном определяться

силой тяготения Солнца. Несмотря на то, что станция в Луну не попала, АМС «Луна-1» стала первым в мире космическим аппаратом, достигшим второй космической скорости, преодолевшим притяжение Земли и ставшим искусственным спутником Солнца.

Таким образом, в целом полёт можно охарактеризовать как частично успешный, рекордный для своего времени и весьма плодотворный с научной точки зрения. Наклон орбиты искусственной двойной планеты (блок «Е» и отделившаяся от неё «Луна-1») к плоскости эклиптики составляет около 1° , эксцентриситет 0,148, минимальное и максимальное расстояние от Солнца соответственно 146,4 млн. км и 197,2 млн. км. Период обращения искусственной планеты — примерно 450 суток. Относительные скорости отделения были невелики и поэтому блок «Е», и отделившаяся от неё «Луна-1» находятся на близких орбитах.

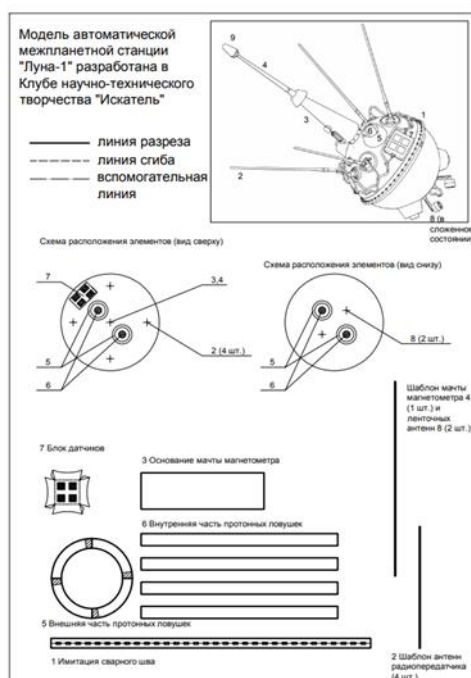


Рис.6 – Устройство Луна-1[4]

1.3 Траектория движения аппарата 5-3 января 1959 года

Для контроля траектории и определения координат использовали автоматизированную систему, разработанную для определения траектории баллистических ракет.

Развертка на поверхность Земли выглядела так:

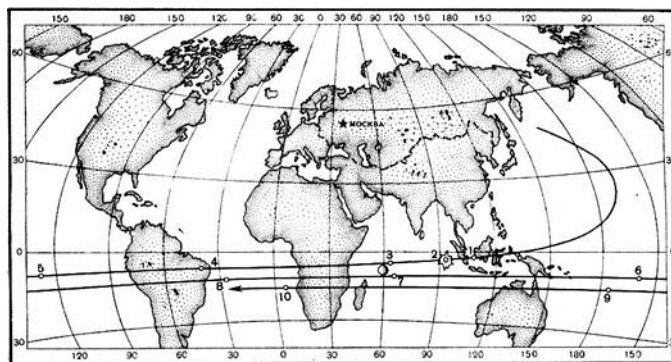


Рис. 7 - Развертка на поверхность Земли, рисунок [5]

Если собрать всю информацию из сообщения ТАСС можно выделить такие точки:

3 января в

3 ч: 3 градуса 12 мин ю. ш. и 108 градусов в. д. 100 000 км от Земли
[1]

6 ч: 4 градуса 30 мин ю. ш. и 63.5 градуса в. д. 137 000 км от Земли
[3]

13 ч: 7 градусов 33 мин ю. ш. и 40 градусов з. д. 209 000 км от Земли
[4]

16 ч: 8 градусов 20 мин ю. ш. и 86(85) градусов з. д. 237 000 км от Земли

19 ч 8 градусов 57 мин ю. ш. и 131(130) градус з. д. 265 000 км от Земли [5]

21 ч 9 градусов 18 мин ю. ш. и 160 градусов з. д. 284 000 км от Земли [6]

4 января в

0 ч: 9 градусов 45 мин ю. ш. и 155 градусов в. д. 311 000 км от Земли

3 ч: 10 градусов 7 мин ю. ш. и 110 градусов в. д. 336 600 км от Земли

В 5 ч 57 мин ракета прошла на минимальном расстоянии от Луны (5-6 тыс. км) и стала спутником Солнца [7]. После чего ТАСС стал публиковать ее координаты в астрономических координатах. Хотя часть координат, как видно на схеме, были пересчитаны и в земные

5 января в 10 ч ее аккумуляторы сели и связь с ней прекратилась.

В квадратных скобках отмечены номера аналогичных точек на карте Земли.

В двух круглых скобках помечены более точные координаты. При анализе данных оказалось, что они заметно выбиваются из общей траектории. Вероятно, в сообщении ТАСС была ошибка в 1 градус западной долготы.

Зная эти данные и скорость вращения Земли можно перестроить их в трехмерные координаты и рассчитать примерную траекторию (рисунок 8).

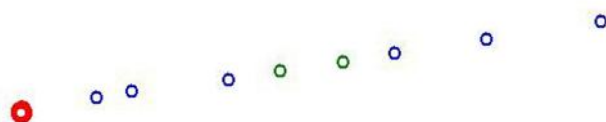


Рис. 8 - Примерная траектория движения [5]

Вторая точка схеме (рисунок 9) заметно выбивается. Тут пытались учесть также и момент запуска искусственной натриевой кометы. К сожалению, из-за облачности над территорией СССР в тот момент (ее сфотографировали только на одной обсерватории), а также из-за того, что расплылось только 10% всего натрия, координаты не очень точны.

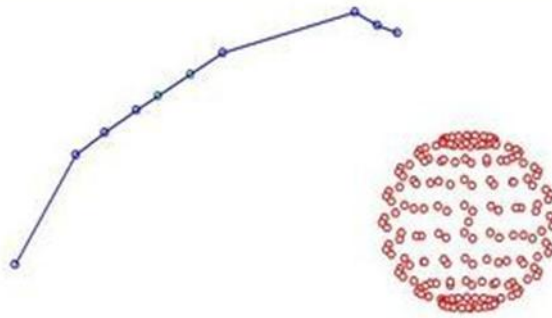


Рис.9 - Схема движения ракеты к Луне [5]

Немного обработав полученный результат, можно получить следующую схему (рисунок 10):

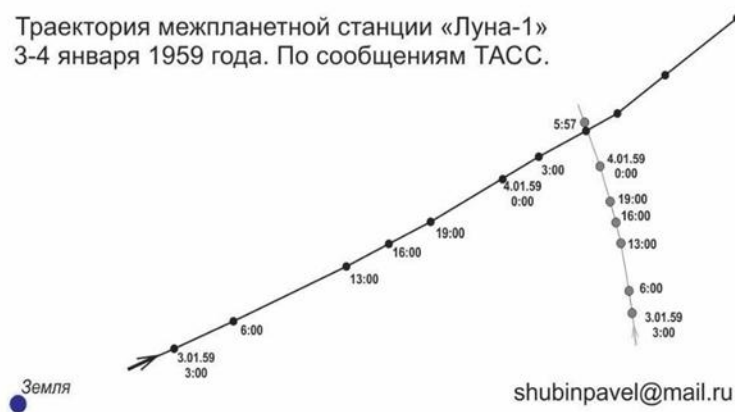


Рис.10 - Траектория межпланетной станции «Луна-1», ТАСС [5]

Хорошо видно, как станция и Луна движутся в место их встречи. А если внимательно изучить траекторию «Луны-1» можно заметить и изгиб после встречи ее с Луной. Гравитационное поле Луны достаточно

изменило ее траекторию, чтобы это можно было заметить даже на такой приблизительной схеме.

<u>Переменные</u>	<u>Значения</u>
Масса первой ступени	3533кг
Масса первой ступени без топлива	2033
Сила тяги двигателей первой ступени	60кН
Масса второй ступени	2789кг
Масса второй ступени без топлива	790кг
Сила тяги второй ступени	60кН
Масса третьей ступени	10690кг
Масса третьей ступени без топлива	2690кг
Сила тяги третьей ступени	215.0кН (в вакууме) 167.969кН (при 1 атм.)
Масса четвёртой ступени	30840кг
Масса четвёртой ступени без топлива	6840кг
Сила тяги четвёртой ступени	3*205кН
Гравитационная постоянная G	$6,67 * 10^{-11}$
Радиус Земли r	6 400 км
Масса Земли M	$6 * 10^{24}$ кг
Молярная масса воздуха μ	29 г/моль
Газовая постоянная R	8,31
Абсолютная температура T	~293 К
Нормальное давление p_0	10^5 Па
Удельная газовая постоянная для воздуха R'	287 Дж/кг*К
Ускорение свободного падения на Луне	$1,622 \text{ м/с}^2$

1.4 Технические характеристики

Табл. 1 - Характеристики и конструкция ракеты-носителя Луна [5.1]

1.5 План выполнения работы

1.5.1 Теоретическая часть:

1. Описание реальной миссии.
2. План реализации миссии с учётом всех ошибок реальной миссии.

1.5.2 Моделирование

1. На основе формул изменения массы ракеты, её высоты и скорости моделируем взлёт
2. На основе формул изменения координат траектории полёта моделируем перелёт ракеты с орбиты Земли до Луны
3. На основе формул изменения массы ракеты, её высоты и скорости моделируем посадку ракеты на Луну.

1.5.3 Работа с KSP

1. Проектирование ракеты
2. Моделирование миссии

1.5.4 Выступление

1. Презентация
2. Отчет команды

1.5.5 Видео технической части.

2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Наша задача разделилась на три составляющие: взлёт ракеты, полёт до Луны и посадка.

2.1 Гомановский переход

Для реализации нашей миссии и построения математической модели мы решили использовать уравнения для орбитальных переходов Гомана (рисунок 11).

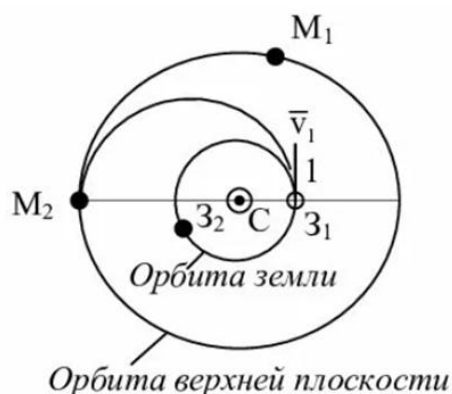


Рис. 11 – Гомановский переход [6]

Гомановская траектория в небесной механике — эллиптическая орбита, используемая для перехода между двумя другими орбитами, обычно находящимися в одной плоскости. В простейшем случае она пересекает эти две орбиты в апоцентре и перигентре [7]. Орбитальный манёвр для перехода включает в себя два импульса работы двигателя на разгон — для входа на гомановскую траекторию и для схода с неё.

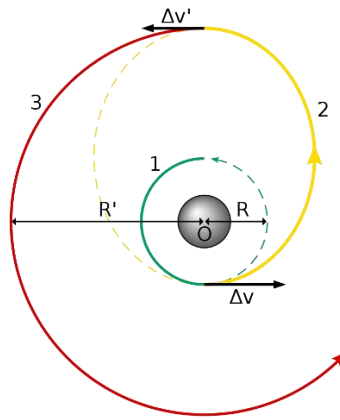


Рис. 12 - Гомановская траектория перехода (жёлтый) с низкой круговой орбиты (зелёный) на более высокую круговую орбиту (красный).

Δv и $\Delta v'$ — первое и второе включения двигателя на разгон [7].

Для расчета необходимых параметров на понадобились следующие формулы:

$$1. V = \sqrt{GM \left(\frac{2}{r} - \frac{1}{a} \right)},$$

где r – радиус орбиты Луны и a – расстояние от нашей ракеты до центра Луны, а множитель под корнем равен стандартному гравитационному параметру Луны (GM). Но так как наш полёт подразумевает полёт по орбите Луны, то в нашем случае $a = r$. Для Луны мы были взяты следующие данные: $a = r = 100.000$ и $GM = 6.5138398 \cdot (10^{**}10)$.

Исходя из расчётов, нами были найдены скорость Гомановского перехода для выхода с Земли и подлёта до Луны: скорость примерно равна 807 м/с.

Для успешного полёта нам также необходимо было знать угол между вектором ракеты, проходящем через центр Земли, и прямой, на которой лежит диаметр Земли, проходящей через Луну (рисунок 12).

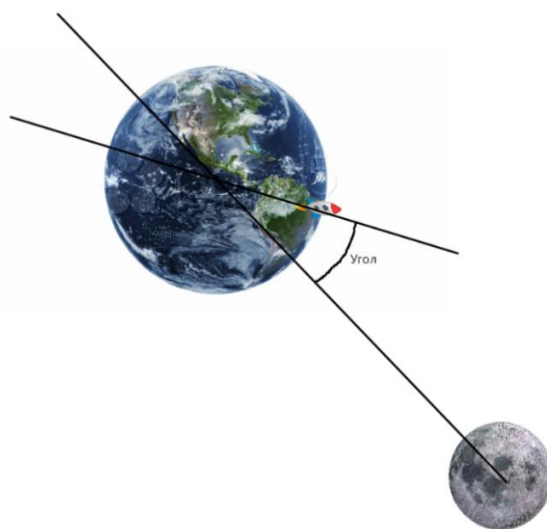


Рис. 13 - угол между вектором ракеты, проходящем через центр Земли, и прямой, на которой лежит диаметр Земли, проходящей через Луну [7.1].

Данный угол мы находили по следующей формуле:

$$2. c = \frac{1}{2} \arcsin \left(\frac{Lg}{v_0^2} \right),$$

где L – расстояние между Землёй и Луной, g – сила притяжения Земли,

v_0 – скорость взлёта нашей ракеты, то есть скорость, пока мы не начали лететь под углом. Итак, мы взяли следующие данные: $L = 11.400.000$, $v_0 = 939$, $g = 9.81$. По нашим расчётам угол вышел равен примерно 41 градусу.

Рисунок 13, описывающий смысл нахождения угла для Гомановского перехода:

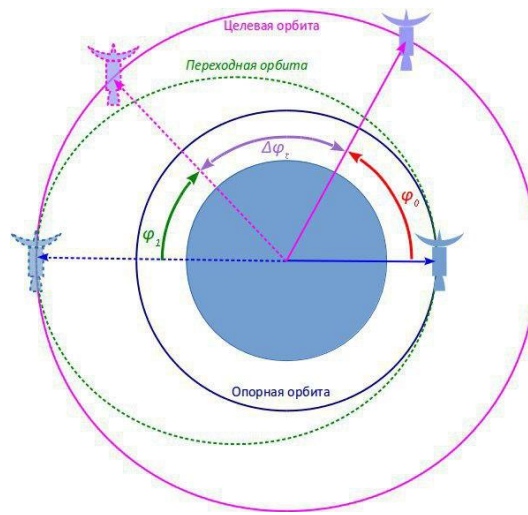


Рис. 14 - смысл нахождения угла для Гомановского перехода [6]

Где опорная орбита - орбита Земли, переходная орбита - сам Гомановский переход, целевая орбита - орбита Луны.

Код вычисления угла для Гомановского перехода на внешнюю орбиту (то есть на лунную) находится в приложении А п. 1.

В формулах приведено то, как мы вычисляли угол, но наши знания в физике не сильны, а потому это формула изначально использовалась для вычисления угла, при котором тело должно попасть на определённое расстояние при определённой начальной скорости.

Найденная нами формула изначально создана для земных тел: пушек, ружей и так далее. Начальная скорость принимается здесь за скорость, с которой снаряд вылетает из дула так сказать. И в зависимости от того, как далеко нам нужно снаряд закинуть мы настраиваем угол между нашей пушкой и целью.

Понятно, что здесь всё по-другому. Мы угол настраиваем тоже для расстояния, но сила притяжения тут уже перестаёт работать, так как в момент начальной скорости мы вышли с атмосферы, поэтому тут g мы убираем и теперь наш угол равен примерно 34 градусам. В нашем случае

мы принимаем начальную скорость за ту скорость, с которой мы выходим с орбиты, то есть это и есть скорость вылета так сказать.

Как оно всё ещё работает? Мы имеем предположение, что гравитационные силы Земли и Луны всё же действуют на тело, а потому она летит не по прямой, а по Эллипсу, а значит её полёт близок к полёту снаряда пушки на Земле с некоторыми погрешностями (пару градусов).

3 ФИЗИЧЕСКАЯ МОДЕЛЬ

3.1 Основные формулы

Мы создали физическую модель, чтобы посчитать, удастся ли нам реализовать миссию по полёту до Луны, с помощью тех данных, которые мы вывели в математической модели. Основные, используемые нами формулы приведены ниже.

Закон всемирного тяготения Ньютона:

$$3. F_{\text{притяжения}} = G \frac{m_1 m_2}{r^2} [8],$$

Где $F_{\text{притяжения}}$ — сила гравитации, действующая между двумя объектами, m_1, m_2 — массы объектов, r — расстояние между центрами их масс, а G — гравитационная постоянная.

Зависимость массы от времени при трате топлива:

$$4. m = m_0 - kt,$$

где, k — коэффициент траты топлива, t — время, m — начальная масса с топливом, m_0 — масса в зависимости от потраченного топлива.

Ускорение свободного падения:

$$5. g = G \frac{M}{R^2},$$

где $G \approx 6,67 \cdot 10^{-11} \frac{M^2}{\text{с}^2}$, M — масса планеты, R — радиус планеты.

Подставляя в эту формулу массу и радиус планет, вычислим ускорение у них.

3.1. Взлёт

Главным условием взлёта будем считать, в первую очередь, - характеристическую скорость орбитального манёвра - в астродинамике и ракетодинамике изменение скорости космического аппарата, которое необходимо для выполнения орбитального манёвра.

Для описания процесса взлёта будем использовать второй закон Ньютона:

$$6. F_{тяги} - F_{притяжения} - F_{сопр} = ma$$

В игре сила атмосферного сопротивления ($F_{сопр}$) моделируется следующим образом:

$$7. F_{сопр} = \rho V^2 d \Lambda [9],$$

где ρ - плотность атмосферы, V^2 - скорость корабля ($\frac{M}{c}$), d - безразмерный коэффициент сопротивления, Λ - площадь поперечного сечения (M^2).

Коэффициент сопротивления (d) рассчитывается как усреднённый по массе коэффициент сопротивления всех деталей корабля. Для большинства аппаратов без открытых парашютов d очень близок к 0.2, так как это - коэффициент сопротивления подавляющего большинства деталей.

Плотность атмосферы ρ прямо пропорционально атмосферному давлению p на данной высоте, являющему собой функцию высоты (a), атмосферного давления на поверхности (p_0), и характеристической высоты (H):

$$8. p = p_0 * e^{-\frac{a}{H}} [9],$$

$$9. \rho = 1.2230948554874 \frac{kg}{m^3 atm} p[9],$$

где p измеряется в атмосферах и ρ в $\frac{кг}{м^3}$. Множитель $1.2230948554874 \frac{kg}{m^3 atm}$, данный в `FlightGlobals.getAtmDensity(1.0)` [9], означает плотность воздуха при одной атмосфере (на уровне моря Кербина).

Ускорение, которое является векторной величиной, показывающей, на сколько изменяется вектор скорости \vec{v} тела при его движении за единицу времени. Тут мы вычисляем чему равна \vec{a} при t :

$$10. \vec{a} = \frac{d\vec{v}}{dt}[10]$$

Второй закон Ньютона — дифференциальный закон движения, описывающий взаимосвязь между приложенной к материальной точке силой и получающимся от этого ускорением этой точки:

$$11. \vec{a} = \frac{\vec{F}}{m}[10],$$

где \vec{a} — ускорение материальной точки, \vec{F} — равнодействующая всех сил, приложенных к материальной точке, m — масса материальной точки. Отсюда получаем:

$$12. \frac{dv}{dt} = \frac{F_{тяги} - F_{притяжения} - F_{сопр}}{m}$$

Для нахождения скорости на ось ОУ распишем формулу [13] на ось ОУ. Для этого введём угол α между нормалью к Кербину и направлением ракеты. С помощью умножения сил, действующих параллельно направлению ракеты, на $\cos \alpha$ получим проекцию этих сил на ось ОУ:

$$13. \frac{dV_y}{dt} = \frac{F_{тяги} * \cos \alpha}{m} - \frac{F_{сопр} * \cos \alpha}{m} - F_{притяжения},$$

Скорости изменения высоты на t вычисляем по следующей формуле:

$$14. v_x = \frac{dx}{dt}; v_y = \frac{dy}{dt}; v_z = \frac{dz}{dt} [11],$$

Нахождение скорости через малый промежуток времени:

$$15. V_{тек} = V_0 + dV,$$

Нахождение высоты через малый промежуток времени:

$$16. H_{тек} = H_0 + dh$$

3.2. Посадка

Далее рассчитаем необходимые параметры ракеты при посадке на Луну.

Аналогично используем формулу второго закона Ньютона (формула [6]) и формулу [14] скорости изменения высоты на t .

Вычисляем по формуле [10] скорость на ось ОУ:

$$17. \frac{dVy}{dt} = \frac{F_{тяги} * \cos\alpha}{m} - F_{притяжения},$$

где α - угол между нормалью и направлением полёта ракеты.

Аналогично формулам [15] и [16] вычисляем $V_{тек}$ и $H_{тек}$.

4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

4.1 Используемые технологии

Для того чтобы осуществлять взаимодействие с игрой существует большое число различных методов. Так как игра не предполагает управление кораблём иным способом нежели через прямое управление, то приходится использовать сторонние методы, которые предлагают различные моды. Основными являются kos, kRPC, MechJeb. Для реализации нашего проекта был выбран мод kRPC так как он даёт большие возможности для управления кораблём и получения с него данных в том числе в реальном времени. Важным преимуществом данного дополнения является возможность написания собственных скриптов, благодаря которым и появляется возможность написать полный автопилот. По данному моду существует подробная документация с примерами на официальном сайте дополнения.

4.2 Логирование

Для того чтобы выполнить логирование нами была использована возможность получения текущих полётных данных из KSP, которую предоставляет мод kRPC. Основной проблемой при выполнении этапа логирования стали глобальные ускорения времени, и фризы игры. Для того, чтобы побороть данные проблемы, мы приняли решение использовать время из самой игры, которое тоже можно получить из дополнения kRPC. Так же мы решили проблему неточностей, неоднократным запуском программы и последующее усреднение результатов полёта.

4.3 Автопилот

Благодаря тому, что для мода kRPC существует подробная документация, удалось изучить основные инструменты общения с сервером игры, а также основные инструменты для управления кораблем. Так как было принято решение о написании полного автопилота, то писать целую программу сразу было бы очень нерационально. Именно поэтому, мы разделили написание автопилота на 3 основных этапа.

1. Взлёт и выход на орбиту
2. Выполнение манёвра перехода на орбиту Луны
3. Посадка на поверхность Луну

Код программ можно найти в приложении А.

4.3.1 Взлет

Основные фазы взлетной траектории:

1. Взлет: начальная фаза, в ходе которой включаются разгонные двигатели первой ступени, и ракета разгоняется вертикально до достижения высоты, на которой начинается изменение угла поворота.
2. Ускорение и наклоны: фаза включает в себя постепенный наклон траектории полета, в результате которого ракета направляется к горизонту.
3. Отделение ступеней и включение новых двигателей: фаза включает в себя отслеживание оставшегося топлива в ступенях и своевременное их отделение для продолжения подъема.
4. Выход на орбиту: Фаза, в ходе которой ракета достигает и выравнивает заданную орбиту.

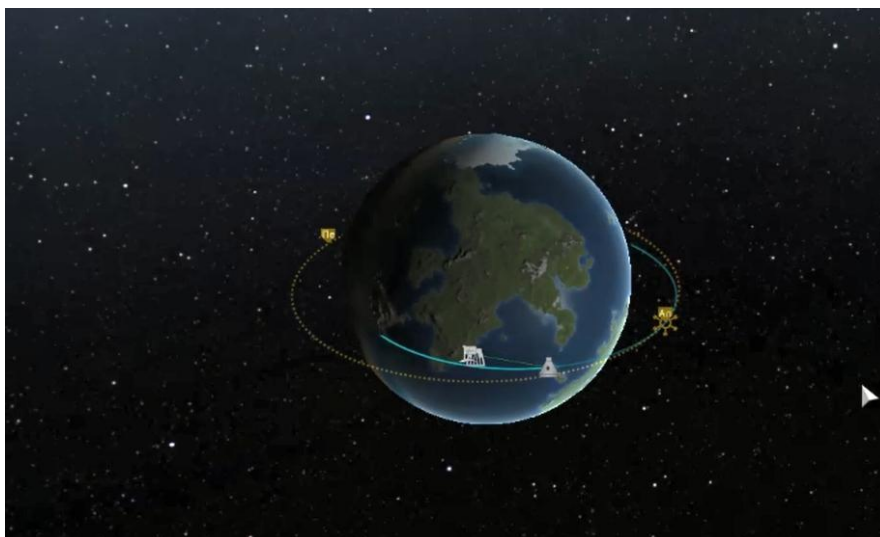


Рис.14 – Орбита в Kerbal Space Program (KSP) [11.1]

Для того чтобы выполнить логирования нами была использована возможность получения текущих полётных данных из KSP, которую предоставляет мод kRPC. Основной проблемой при выполнении этапа логирования стали глобальные ускорения времени, и фриззы игры. Для того, чтобы побороть данные проблемы, мы приняли решение использовать время из самой игры, которое тоже можно получить из дополнения kRPC. Так же мы решили проблему неточностей, неоднократным запуском программы и последующее усреднение результатов полёта.

4.3.2 Выполнение манёвра перехода на орбиту Луны

Для того, чтобы наш полет не зависел от времени старта миссии, мы используем полет через промежуточную орбиту на Земле. Для этого мы будем ждать необходимой позиции запуска двигателя уже на орбите. Когда угол становится равным углу посчитанному по формуле [2] мы запускаем двигатели, которые разгоняют ракету до скорости, необходимой для успешного выполнения Гомановского перехода.

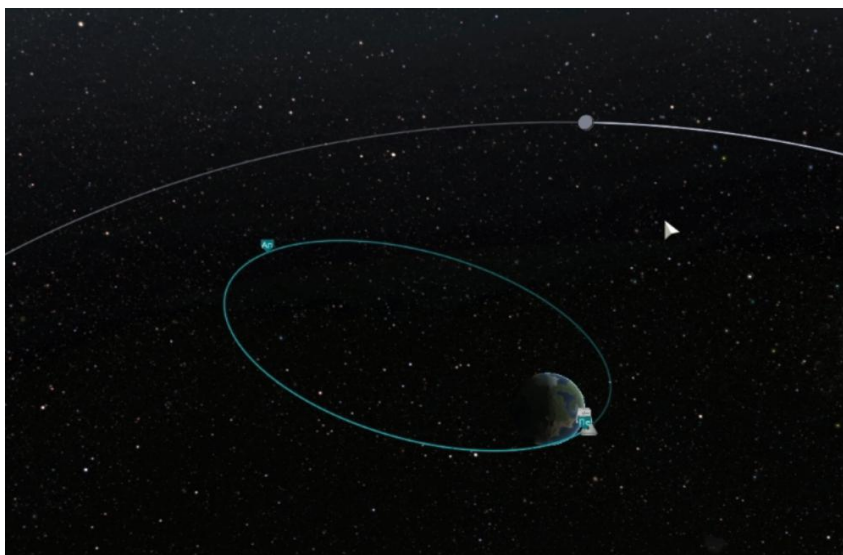


Рис.15 – Момент перехода на орбиту Луны в KSP [11.2]

4.3.3 Посадка на Луну

По достижении минимального расстояния между Луной и кораблём, мы начинаем этап посадки. Который включает в себя расчёт высоты, на которой необходимо включить двигатели, а также своевременное их включение для выполнения мягкой посадки. Данный этап являлся самым проблемным среди всех, так как было перебрано большое количество различных вариантов расчета необходимой высоты, используя различные формулы, каждая из которых давала неприемлемую погрешность. Только спустя большое количество времени нам удалось вывести математическую модель, обеспечивающую необходимую точность.



Рис.16 – Момент посадки на Луну в KSP [11.3]

Игровая модель ракеты представлена в приложении В.

5 ПОСТРОЕНИЕ, АНАЛИЗ И СРАВНЕНИЕ ГРАФИКОВ ПО KSP И МАТЕМАТИЧЕСКОЙ МОДЕЛИ

5.1 Программа математического моделирования движения

Для построения графиков на основе математической модели необходимо было написать программу на языке программирования Python, моделирующую взлет и посадку ракеты на основе физической модели. При программировании мы использовали такие библиотеки как: SciPy, NumPy (для вычисления дифференциальных уравнений), Matplotlib (дает возможность численно решить уравнения движения и визуализировать результаты для более наглядного понимания), а также встроенный в Python модуль math. Для физического описания процесса полета мы использовали второй закон Ньютона.

5.1.1 Описание Математической Модели

Математическая модель включает в себя систему дифференциальных уравнений. С помощью них описывается изменение координат, скоростей и массы ракеты в процессе ее взлета на орбиту. Также при составлении программы важно было уделить внимание точному отражению динамики изменяемых параметров, так как такой подход позволит построить наиболее приближенные к реальности графики.

5.1.2 Используемые технологии

Для написания программы математического моделирования мы применяли следующие технологии:

1. Python: Язык программирования, выбранный за его удобство и широкие возможности.
2. matplotlib: библиотека Python для визуализации данных, позволяющая строить необходимые графики.
3. SciPy [13]: библиотека для языка Python, основанная на расширении NumPy, но для более глубоких и сложных научных вычислений, анализа данных и построения графиков.
4. NumPy [14]: для работы с числовыми массивами, векторами и матрицами, позволяет строить графики и гистограммы.
5. Модуль math: предоставляет обширный функционал для проведения вычислений с вещественными числами (числами с плавающей точкой).

5.2 Сравнение графиков

Графики, выводимые программой, включают в себя две основные линии данных: первую, отражающую значения, полученные из математической модели, и вторую, основанные на данных из игры Kerbal Space Program. Это позволяет нам провести сравнительный анализ между тем, как наша модель предсказывает движение объекта, и данными из симулятора космического полета KSP.

Кривые, представленные на графиках, показывают визуальное отображение скорости объекта от времени, высоты объекта от времени и массы объекта от времени при взлёте (см. приложение С п. 1) и при посадке (см. приложение С п. 2). Сравнение поведения этих кривых

поможет нам оценить точность нашей математической модели, а также ее соответствие данным из игрового мира Kerbal Space Program.

5.3 Причины расхождений между графиками

Расхождения между графиками в математической модели и данных из Kerbal Space Program могут быть обусловлены несколькими причинами:

1. Упрощенность Модели: при построении математической модели зачастую упрощаются определенные аспекты реального движения объекта. Например, в нашей математической модели было упрощено вычисление силы сопротивления. Как правило, определение коэффициентов необходимых для расчета сопротивления — это сложный процесс по ряду причин. Во-первых, их величины не постоянны. Более того, зачастую их определение происходит экспериментальным путем в аэродинамической трубе или на испытательном полигоне, оснащенном точным измерительным оборудованием.
2. Вариации в Физических Параметрах: иногда даже незначительное изменение физических характеристик объекта способны привести к изменениям в его движении. Такие параметры, как масса объекта, воздействующие на него силы и другие, могут быть сложны для точного определения.
3. Численные Округления и Погрешности: при численном решении дифференциальных уравнений возникают округления, которые, как правило, накапливаются и оказывают непосредственное влияние на конечный результат. Подобные численные погрешности вносят свой вклад в расхождения.

4. Особенности Данных из KSP: Данные из KSP могут содержать некоторые особенности, которые не были учтены в математической модели. Например, особенности физического движка игры или влияние других параметров.

5.4 Выводы

Анализируя расхождения между графиками и учитывая возможные причины их возникновения, можно прийти к выводу о том, что, не смотря на недостаток исходных данных для физической модели и других преград, составленная математическая модель достаточно точно отражает изменение параметров ракеты на этапе взлета и нам удалось наиболее приближенно к реальности рассчитать зависимости.

6 ЗАКЛЮЧЕНИЕ

6.1 Итоги

В процессе создания нашего проекта нам удалось самостоятельно смоделировать полёт ракеты-носителя «Луна» для совершения лунной миссии по доставке аппарата Луна-1 на Луну и реализовать его, используя программу Kerbal Space Program.

Медиа к работе можно найти в приложении D.

Учебные итоги:

1. Освоение математического моделирования физических процессов.
2. Совершенствование навыков программирования.
3. Ознакомление с научным симулятором KSP.
4. Совершенствование навыков командной работы
5. Получение прикладных навыков выполнения работы в срок.

Проектные итоги:

1. Получены математические модели, описывающие нашу миссию.
Произведено моделирование миссии в научном симуляторе KSP.
Созданы программные симуляции.
2. Составлен отчет о реализации миссии по запуску спутника Луна-1.
3. Удалось провести верные расчёты для посадки аппарата Луна-1 на Луну.

6.2 Работа участников проекта

Работу каждого участника проекта мы оценили следующим образом:

Шаталов Максим: разделение и планирование задач, а также управление командной работой. Изучение работы дополнения kRPC и написание программы автопилота.

Агафонов Андрей: главный составитель математической и физической модели, а также помощник программиста по составлению графиков.

Верменников Михаил: программист графиков и математических моделей. Были вычислены угол для выполнения Гомановского перехода с Земли на Луну, а также значение, которое необходимо было прибавить к текущим развитым скоростям на орбите Земли для успешного выполнения Гомановского перехода на внешнюю орбиту Луны. Михаилом были реализованы графики по логам, предоставленными из KSP Максимом Шаталовым, а также реализован изначальный программный код для математической модели взлёта и посадки, который впоследствии дорабатывался Андреем Агафоновым.

Алексеева Виктория: изучила историческую справку о миссии; собрала и структурировала информацию. Отвечала за подготовку и написание текстового отчета, презентации. Продумывала и писала сценарий к видеоотчету.

ИСТОЧНИКИ

Электронные источники:

1. https://znaniya-sila.narod.ru/solarsis/zemlya/moon_02.htm
2. Как межпланетная станция "Луна-1" стала первой искусственной кометой. [web-ресурс] – Режим доступа: <https://dzen.ru/a/YdmcB2hlcS91Llg6>
3. Луна-1 - Википедия. [web-ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9B%D1%83%D0%BD%D0%B0-1>
4. Модель автоматической 9 межпланетной станции "Луна-1" разработана в Клубе научно-технического творчества "Искатель". [pgf - file] – Режим доступа: <http://ntm.pro/images/DKB/Rockets/moon-1/moon-1.pdf>
5. Путь «Луны-1»/Хабр [web-ресурс] – Режим доступа: <https://habr.com/ru/articles/228763/>
- 5.1. Таблица характеристики и конструкция ракеты-носителя Луна. Составлена при помощи источников [1], [2], [3].
6. Гомановская траектория в небесной механике - Википедия. [web-ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D0%BC%D0%B0%D0%BD%D0%BE%D0%B2%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D1%80%D0%B0%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D1%8F
7. Апоцентр и перицентр – Википедия. [web-ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BE%D1%86%D0%B5%D0%BD%D1%82%D1%80_%D0%B8_%D0%BF%D0%B5%D1%80%D0%B8%D1%86%D0%B5%D0%BD%D1%82%D1%80
- 7.1. Рисунок, изображающий угол между вектором ракеты, проходящем через центр Земли, и прямой, на которой лежит диаметр Земли, проходящей через Луну. Изображали сами.

8. Закон всемирного тяготения Ньютона – Википедия. [web-ресурс] – Режим доступа:
https://en.wikipedia.org/wiki/Newton%27s_law_of_universal_gravitation
9. Atmosphere/ru – Википедия. [web-ресурс] – Режим доступа:
<https://wiki.kerbalspaceprogram.com/wiki/Atmosphere/ru>
10. Ускорение – Википедия. [web-ресурс] – Режим доступа:
<https://ru.wikipedia.org/wiki/%D0%A3%D1%81%D0%BA%D0%BE%D1%80%D0%B5%D0%BD%D0%B8%D0%B5>
11. Скорость – Википедия. [web-ресурс] – Режим доступа:
<https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%BE%D1%80%D0%BE%D1%81%D1%82%D1%8C>
- 11.1. Орбита в Kerbal Space Program (KSP). Наш скриншот экрана во время запуска игры.
- 11.2. Момент перехода на орбиту Луны в KSP. Наш скриншот экрана во время запуска игры.
- 11.3. Момент посадки на Луну в KSP. Наш скриншот экрана во время запуска игры.
12. API: FlightGlobals – Википедия. [web-ресурс] – Режим доступа:
<https://wiki.kerbalspaceprogram.com/wiki/API:FlightGlobals>
13. SciPy — это библиотека для языка Python - blog.skillfactory. [web-ресурс] – Режим доступа:
<https://blog.skillfactory.ru/glossary/scipy/#:~:text=SciPy%20%E2%80%94%D1%8D%D1%82%D0%BE%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0%20%D0%B4%D0%BB%D1%8F,%D0%B2%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D0%B9%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C%D1%8E%20%D0%B8%20%D1%81%D0%BA%D0%BE%D1%80%D0%BE%D1%81%D1>

https://gb.ru/posts/python_numpy

14. Библиотека Numpy - gb. [web-ресурс] – Режим доступа:
https://gb.ru/posts/python_numpy

Книжные источники:

12. Астрономия. 10 — 11 классы: учеб. для общеобразоват. организаций: базовый уровень / В. М. Чаругин. — Информационно-библиографическая культура: учеб. пособие / В. В. Чаругин. — М.: Просвещение, 2018. — ил. — (Сферы 1-11). - ISBN 978-5-09-053903-6 — стр. 45

13. Астрономия. 10 — 11 классы: учеб. для общеобразоват. организаций: базовый уровень / В. М. Чаругин. — Информационно-библиографическая культура: учеб. пособие / В. В. Чаругин. — М.: Просвещение, 2018. — 144 с. : ил. — (Сферы 1-11). - ISBN 978-5-09-053903-6 — стр. 48

14. Практикум по компьютерному математическому моделированию. Часть II: Компьютерное моделирование физических процессов: учебно-методическое пособие / О. А. Широкова – Казань: КФУ, 2015. – стр. 20

15. Курс общей физики. Механика / Алешкевич В. А., Деденко Л. Г., Караев В. А. Под ред. В. А. Алешкевича. — М.: ФИЗМАТЛИТ, 2011. — —ISBN 978-5-9221-1271-0. – стр. 204

ПРИЛОЖЕНИЕ А

Часть 1

**Код вычисления угла для Гомановского перехода на внешнюю орбиту
(то есть на лунную)**

```
import math
L = 11_400_000
v0 = 939
g = 1

inner = (L * g) / (v0 ** 2)
print(inner / (1 + abs(inner)))
angle = 1/2 * math.asin(inner / (1 + abs(inner)))

print(angle)
```

Часть 2

Взлёт

```
# импорт используемых библиотек
import math
import time
import krpc
import os
import json

def log():
    global data
    current_ut = space_center.ut
    v = vessel.flight(vessel.orbit.body.reference_frame).vertical_speed
    vector_v_all =
vessel.flight(vessel.orbit.body.reference_frame).velocity
    v_all = (vector_v_all[0] ** 2 + vector_v_all[1] ** 2 + vector_v_all[2]
** 2) ** 0.5
    h = vessel.flight(vessel.orbit.body.reference_frame).surface_altitude
    data.append([vessel.mass, h, v, v_all, current_ut])
    time.sleep(0.3)

# выставаем необходимые параметры орбиты
turn_start_altitude = 250
turn_end_altitude = 45000
target_altitude = 150000
```



```

# настраиваем соединение с сервером игры
conn = krpc.connect(name='Launch into orbit')

# получаем данные о корабле
vessel = conn.space_center.active_vessel
space_center = conn.space_center
tech_stage = 7
ut = conn.add_stream(getattr, conn.space_center, 'ut')
altitude = conn.add_stream(getattr, vessel.flight(), 'mean_altitude')
apoapsis = conn.add_stream(getattr, vessel.orbit, 'apoapsis_altitude')
stage_resources = vessel.resources_in_decouple_stage(stage=tech_stage,
cumulative=False)
stage_fuel = conn.add_stream(stage_resources.amount, 'LiquidFuel')

# настраиваем корабль перед взлетом
vessel.control.sas = False
vessel.control.rcs = False
vessel.control.throttle = 1.0
turn_angle = 0
data = []

# выводим обратный отсчет перед взлетом
print('3...')
time.sleep(1)
print('2...')
time.sleep(1)
print('1...')
time.sleep(1)
print('Взлет!')

# запуск двигателя
vessel.control.activate_next_stage()
vessel.auto_pilot.engage()
vessel.auto_pilot.target_pitch_and_heading(90, 90)

while True:

    # выполняем полное логирование полета
    log()

    # изменение угла поворота корабля при взлете
    if altitude() > turn_start_altitude and altitude() < turn_end_altitude:
        frac = ((altitude() - turn_start_altitude) /
                (turn_end_altitude - turn_start_altitude))
        new_turn_angle = frac * 90
        if abs(new_turn_angle - turn_angle) > 0.5:
            turn_angle = new_turn_angle
            vessel.auto_pilot.target_pitch_and_heading(90 - turn_angle, 90)

    # управление системой отсоединения ступеней
    if stage_fuel() < 0.1:
        vessel.control.activate_next_stage()
        tech_stage -= 1
        time.sleep(1)
        stage_resources =
vessel.resources_in_decouple_stage(stage=tech_stage, cumulative=False)
        stage_fuel = conn.add_stream(stage_resources.amount, 'LiquidFuel')
        time.sleep(1)
        print('stage separated', tech_stage, stage_fuel())

# ВЫХОД ИЗ ЦИКЛА В МОМЕНТ ДОСТИЖЕНИЯ ВЫСОТЫ КОНЦА МАНЕВРА

```

```

    if apoapsis() > target_altitude * 0.9:
        print('Approaching target apoapsis')
        break

vessel.control.throttle = 0.25
while apoapsis() < target_altitude:
    log()

vessel.control.throttle = 0.0

# ждем выхода из атмосферы
while altitude() < 70500:
    log()

# сохранение в файл логов взлета
with open("Data_vzlet", 'w', encoding="UTF-8") as file:
    json.dump(data, file, ensure_ascii=False, indent=2)

# проверка на то, что запись логов велась
if len(data) != 0:
    print("Лог взлета записан!")

# планирование маневра закругления орбиты
print('Planning circularization burn')
mu = vessel.orbit.body.gravitational_parameter
r = vessel.orbit.apoapsis
a1 = vessel.orbit.semi_major_axis
a2 = r
v1 = math.sqrt(mu * ((2. / r) - (1. / a1)))
v2 = math.sqrt(mu * ((2. / r) - (1. / a2)))
delta_v = v2 - v1
node = vessel.control.add_node(ut() + vessel.orbit.time_to_apoapsis,
prograde=delta_v)

# рассчитывание времени работы двигателя для выполнения маневра
F = vessel.available_thrust
Isp = vessel.specific_impulse * 9.82
m0 = vessel.mass
m1 = m0 / math.exp(delta_v / Isp)
flow_rate = F / Isp
burn_time = (m0 - m1) / flow_rate

# ориентация корабля для маневра
vessel.auto_pilot.reference_frame = node.reference_frame
vessel.auto_pilot.target_direction = (0, 1, 0)
vessel.auto_pilot.wait()

# запуск ускорения времени до старта манёвра
burn_ut = ut() + vessel.orbit.time_to_apoapsis - (burn_time / 2.)
lead_time = 5
conn.space_center.warp_to(burn_ut - lead_time)
time_to_apoapsis = conn.add_stream(getattr, vessel.orbit,
'time_to_apoapsis')
while time_to_apoapsis() - (burn_time / 2.) > 0:
    pass

# выполнение манёвра
vessel.control.throttle = 1.0
time.sleep(burn_time - 0.1)
vessel.control.throttle = 0.05
remaining_burn = conn.add_stream(node.remaining_burn_vector,
node.reference_frame)

```

```

fl = True
while remaining_burn()[1] > 0.5:
    if stage_fuel() < 0.1:
        vessel.control.activate_next_stage()
        tech_stage -= 1
        time.sleep(1)
        stage_resources =
vessel.resources_in_decouple_stage(stage=tech_stage, cumulative=False)
        stage_fuel = conn.add_stream(stage_resources.amount, 'LiquidFuel')
        time.sleep(1)
        fl = False
vessel.control.throttle = 0.0
node.remove()

# проверяем оставшееся топливо
if fl == True:
    vessel.control.activate_next_stage()
    vessel.control.activate_next_stage()

# закрываем соединение с сервером
conn.close()

# взел выполнен, запуск файла перелета на Муну
print("Взлет завершен!")
file = "Orbit_change.py"
os.system(f'python {file}')

```

Перелёт и посадка

```

import time
import json
import math
import krpc

# настраиваем соединение с сервером игры
conn = krpc.connect()

# получаем данные о корабле
space_center = conn.space_center
vessel = conn.space_center.active_vessel
data = []

# получение радиус-векторов позиции космического корабля и позиции Луны
distance_to_moon = 100000000000
space_center.rails_warp_factor = 5
min_dist = 1000000000000

# ожидания окна посадки
while distance_to_moon >= 100000:
    min_dist = min(min_dist, distance_to_moon)
    ship_position =
space_center.active_vessel.position(space_center.active_vessel.orbit.body.
reference_frame)

```

```

    moon_position =
space_center.bodies["Mun"].position(space_center.active_vessel.orbit.body.
reference_frame)
    distance_to_moon = ((moon_position[0] - ship_position[0]) ** 2 +
(moon_position[1] - ship_position[1]) ** 2 + (
        moon_position[2] - ship_position[2]) ** 2) ** 0.5
    print(distance_to_moon, min_dist, "метров")
    if (distance_to_moon > min_dist):
        break
    time.sleep(2)
space_center.physics_warp_factor = 0

# настройка корабля для манёвра выравнивания орбиты Муну
vessel.control.sas_mode = vessel.control.sas_mode.retrograde
vessel.control.throttle = 1
time.sleep(2)
apoapsis_altitude = 10 ** 14
periapsis_altitude = 0

# выравнивание орбиты
while abs(periapsis_altitude - apoapsis_altitude) > 10 ** 6:
    vessel = space_center.active_vessel
    orbit = vessel.orbit
    apoapsis_altitude = orbit.apoapsis_altitude
    periapsis_altitude = orbit.periapsis_altitude
    time.sleep(1)
space_center.physics_warp_factor = 0

# отсоединяем ступень перед посадкой
vessel.control.activate_next_stage()
vessel.control.activate_next_stage()

print("Орбита луны успешно выровнена")
time.sleep(3)

#####
#####
# переход к посадке

# настройка корабля перед посадкой
vessel.control.sas_mode = vessel.control.sas_mode.retrograde

# получение орбиты посадки
pere_altitude = conn.get_call(getattr, vessel.orbit, 'periapsis_altitude')
expr = conn.krpc.Expression.less_than(
    conn.krpc.Expression.call(pere_altitude),
    conn.krpc.Expression.constant_double(-198000))
event = conn.krpc.add_event(expr)
vessel.control.throttle = 1

# ожидание выхода на орбиту посадки
with event.condition:
    event.wait()
vessel.control.throttle = 0
time.sleep(1)

# ускорение времени до высоты 25км
space_center.rails_warp_factor = 5
while vessel.flight(vessel.orbit.body.reference_frame).surface_altitude >
25000:
    time.sleep(0.1)
    continue

```

```

space_center.physics_warp_factor = 0

# ожидание высоты 21км
while vessel.flight(vessel.orbit.body.reference_frame).surface_altitude >
21000:
    time.sleep(0.1)
    continue

h = 1
V = 1

# запуск алгоритма получения высоты посадки и ожидания необходимой высоты
while h > 0 or V > 2:
    F = 60000
    cos_f = 1
    #k = 3.54
    k = 13
    m0 = vessel.mass
    h = vessel.flight(vessel.orbit.body.reference_frame).surface_altitude
    GM = 6.5138398 * 10 ** 10
    r = 200_000
    # dt = 0.1
    V = vessel.flight(vessel.orbit.body.reference_frame).vertical_speed
    dt = 0.01
    t = 0
    current_ut = space_center.ut

    # запись логов посадки
    data.append([m0, h, V, current_ut])

    time.sleep(0.1)
    while V < 0 and h > 0:
        t += 0.01
        dVy = ((F * cos_f) / (m0 - k * t) - GM / ((r + h) ** 2)) * dt
        V = V + dVy
        dh = V * dt
        h = h + dh

    print(h, V)

# запуск двигателя посадки
vessel.control.throttle = 1
vessel.control.legs = True

# финальная корректировка мягкой посадки
while vessel.flight(vessel.orbit.body.reference_frame).surface_altitude >
20:

    # запись логов посадки
    current_ut = space_center.ut
    v = vessel.flight(vessel.orbit.body.reference_frame).vertical_speed
    h = vessel.flight(vessel.orbit.body.reference_frame).surface_altitude
    data.append([vessel.mass, h, v, current_ut])
    time.sleep(0.1)

    if v < -12:
        vessel.control.throttle = 1
    elif v > -8:
        vessel.control.throttle = 0
    time.sleep(1)

```

```

        time.sleep(0.1)

# финальная корректировка мягкой посадки на высоте ниже 20 метров
while vessel.flight(vessel.orbit.body.reference_frame).surface_altitude >
5:

    # запись логов посадки
    current_ut = space_center.ut
    v = vessel.flight(vessel.orbit.body.reference_frame).vertical_speed
    h = vessel.flight(vessel.orbit.body.reference_frame).surface_altitude
    data.append([vessel.mass, h, v, current_ut])
    time.sleep(0.1)

    if v < -2:
        vessel.control.throttle = 0.25
    elif v > -2:
        vessel.control.throttle = 0
        time.sleep(1)
    time.sleep(0.1)
vessel.control.throttle = 0

# успешная посадка
print("Успешная посадка!")

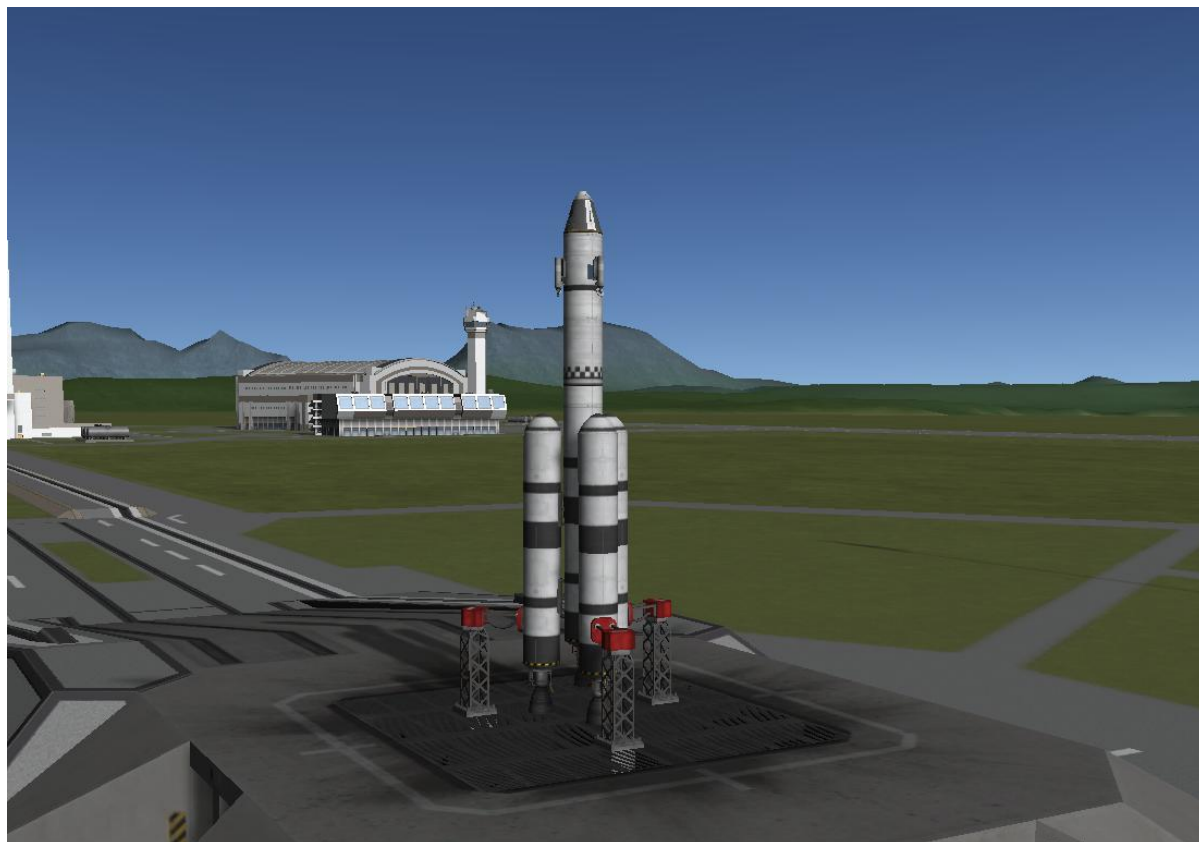
# запись файлов логов
with open("Data_landing", 'w', encoding="UTF-8") as file:
    json.dump(data, file, ensure_ascii=False, indent=2)

# закрываем соединение с сервером
conn.close()

```

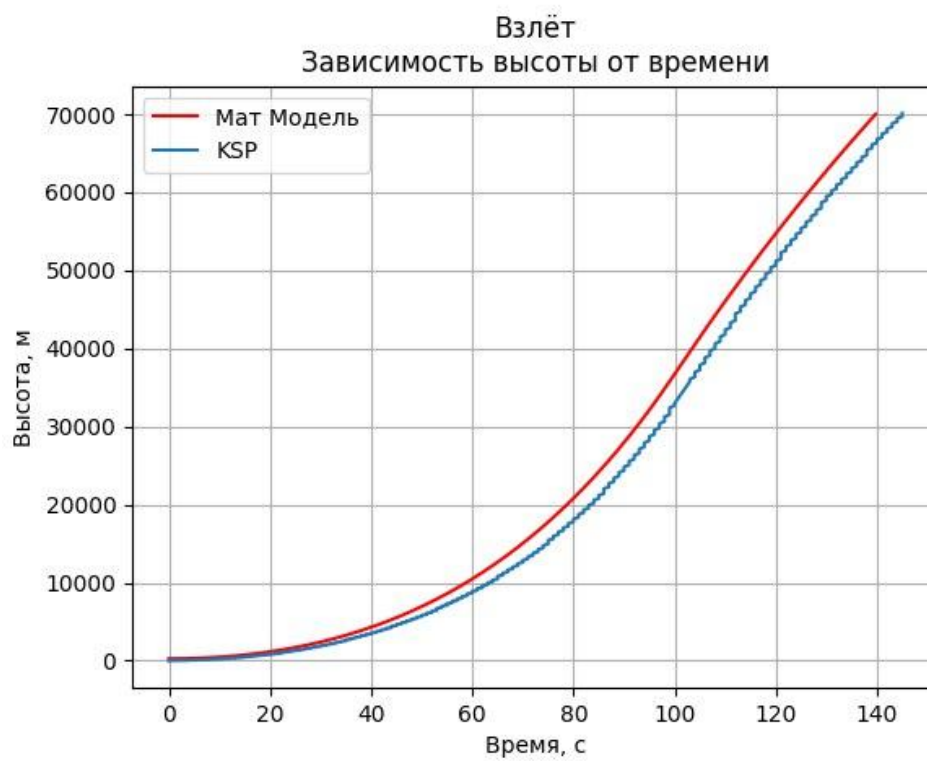
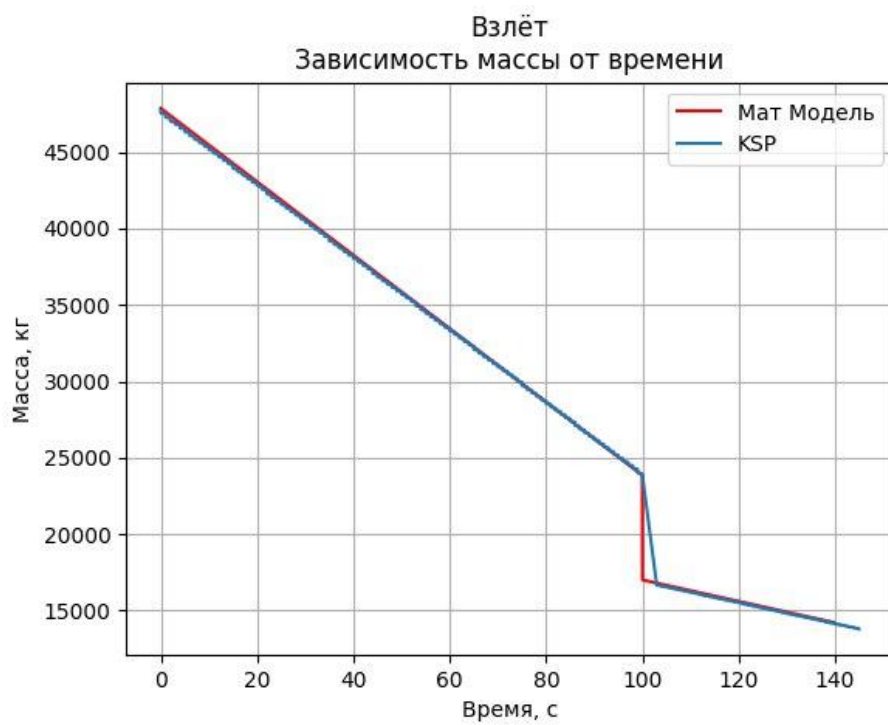
ПРИЛОЖЕНИЕ В

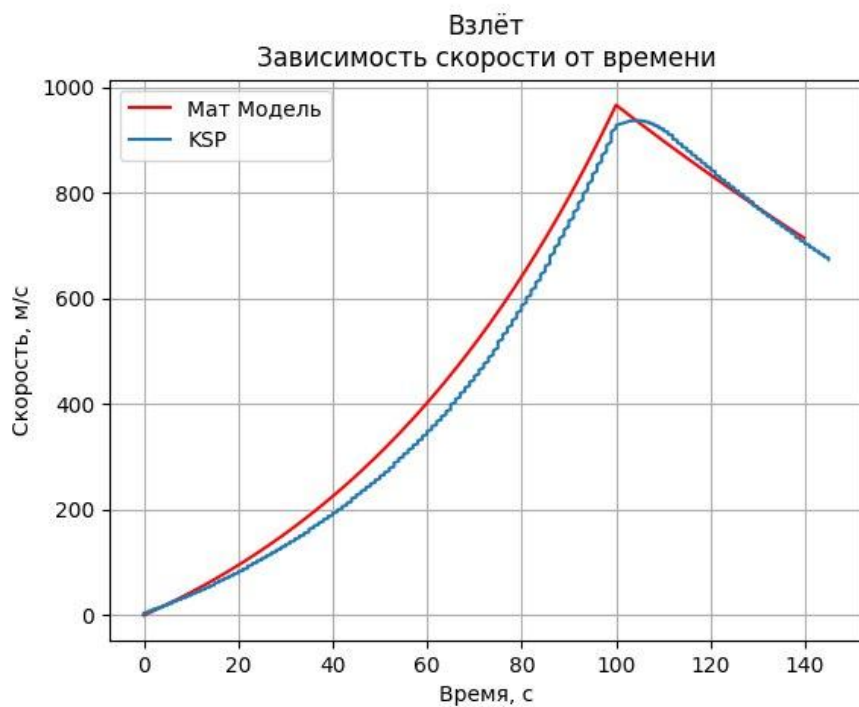
Игровая модель ракеты KSP



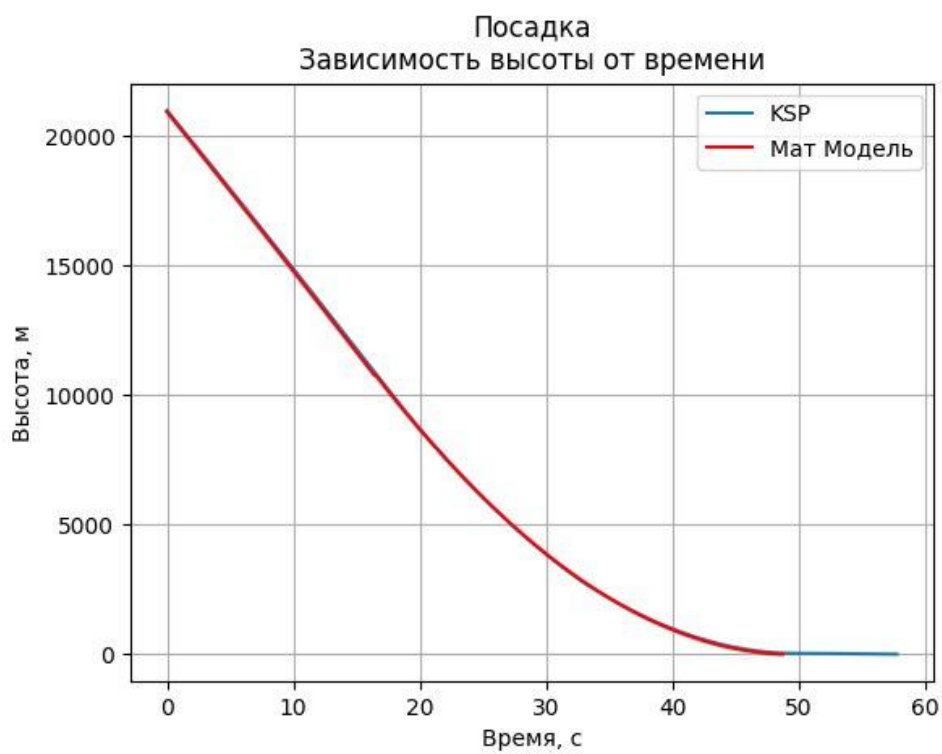
ПРИЛОЖЕНИЕ С

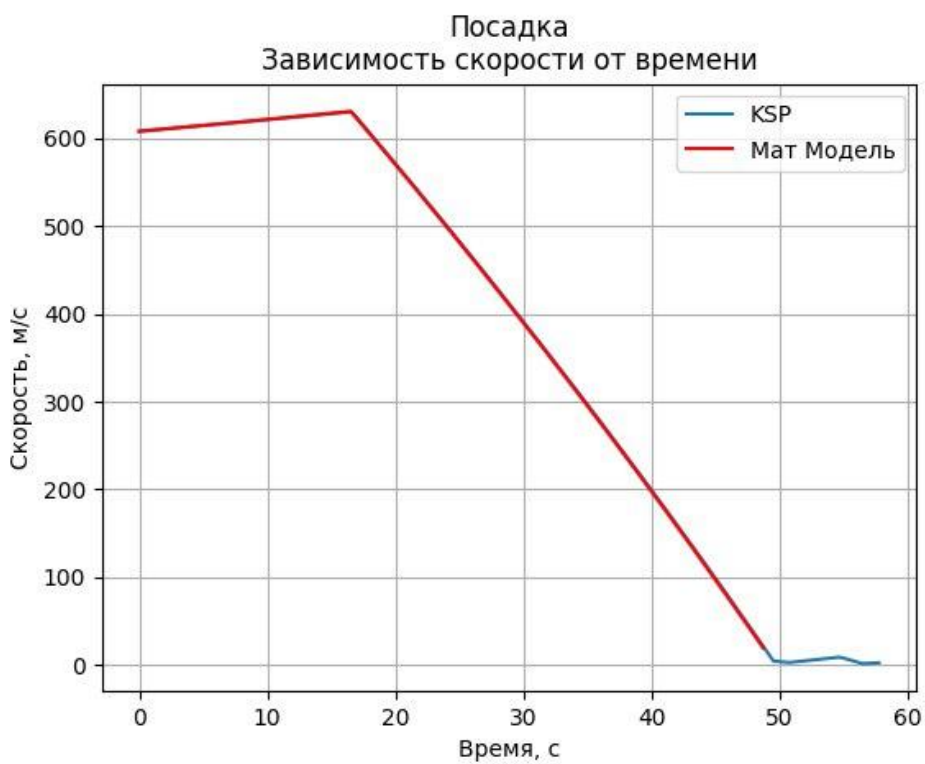
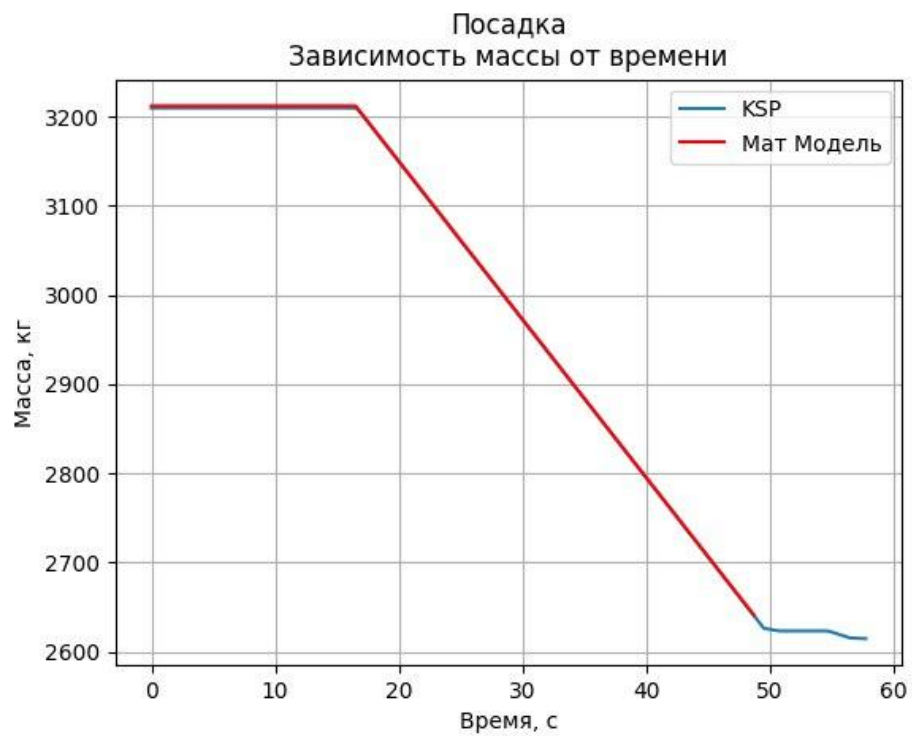
Часть 1





Часть 2





ПРИЛОЖЕНИЕ D
Ссылка на GitHub



Ссылка на отчетное видео

