

Министерство образования и науки Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Физический факультет
Кафедра астрономии и космической геодезии

Лабораторная работа №1
Численное моделирование задачи улучшения орбиты ИСЗ

Автор работы
студент группы № 052472
Виджая С.

Проверил
ст. преподаватель
_____ А. Г. Александрова

Оглавление

1.	Процедура перезаписи входного файла.....	3
2.	Процедура для обращения к интегратору	4
3.	Вычисления начальных координат и скоростей	4
4.	Подбор вариации	5
5.	Улучшение орбиты	9
6.	Вывод	12

1. Процедура перезаписи входного файла

Для начала требуется создать процедуру перезаписи входного файла. В качестве параметра эта функция принимает вектор положения и скорости, дата, месяц, год, час, минут, секунд конечного момента интегрирования и шаг интегрирования.

```
def configure_file(self, coordinate, velocity, date, month, year, hour, minute, second, t):
    f = open(PATH_IN, "w", encoding="cp866")
    f.write(INPUT_TEMPLATE.format(
        self.format_scientific(coordinate[0]),
        self.format_scientific(coordinate[1]),
        self.format_scientific(coordinate[2]),
        self.format_scientific(velocity[0]),
        self.format_scientific(velocity[1]),
        self.format_scientific(velocity[2]),
        date,
        month,
        year,
        hour,
        minute,
        second,
        t
    ))
```

INPUT_TEMPLATE – это шаблон входного файла.

```
INPUT_TEMPLATE = """1    РЕЖИМ (1 - прогноз; 2 - улучшение орбиты)
2018  9  8  0  0  0.000  Начальная эпоха (ТТ)
1      Число спутников
{} {} {}
{} {} {}
|
| | | | ПРОГНОЗ
2018  9  8  0  0  0.000  Начальный момент прогноза (ТТ)
{} {} {} {} {} {}      Конечный момент прогноза (ТТ)
{} Шаг выдачи (сек)
1E-3   Ошибка большой полуоси (км)

УЛУЧШЕНИЕ
obsvyb.in  Файл {} наблюдениями
5          Число наблюдений
2822.172315 2201.434565 5279.171158 координаты обсерватории (км)
0.10       Точность улучшения (сходимости) (км)
1E-5 1E-8   Вариации начальных координат и скоростей (км и км/с)
0.1        Множитель, улучшающий сходимость
2          Начальные условия (1 - из файла; 2 - вычисляются в программе)
8000       Приблизительный радиус-вектор (км) на момент 1-го наблюдения

ИНТЕГРИРОВАНИЕ
10.        Постоянный шаг интегрирования (сек; для отриц. параметра)
19         Порядок интегратора (от 7 до 39 через 4)
10         Параметр интегратора
1000       Интервал промежуточных выдач на экран (в шагах интегрирования)"""
```

```

ВОЗМУЩЕНИЯ
0 0 Гармоники геопотенциала (NM)
  0 Луна
  0 Солнце
  0 Световое давление и ПР эффект
0 0 0 Релятивистские эффекты (Ф_0, Ф_1, Ф_2)
  0 Приливы
  0 Атмосфера
100. Высота сгорания (км)

СПУТНИК
500. Масса (кг)
0.5 Площадь миделева сечения (м^2)
2. Коэф-т лобового сопротивления
2. Коэф-т отражения
****

```

2. Процедура для обращения к интегратору

Библиотека subprocess была использована для запуска .exe файл.

```

def run_exe_file(self):
    sp.call(PATH_EXE)

```

3. Вычисления начальных координат и скоростей

Был выбран спутник «Геостационарный», который имеет следующие орбитальные параметры:

1. Большая полуось (a): 42.165 км
2. Эксцентриситет (e): 0.001
3. Наклонение (i): 1°
4. Долгота восходящего узла (Ω): 0°
5. Аргумент перицентра (ω): 0°
6. Средняя аномалия (M_0): 0
7. Период вращения (T): 24 часа (86400 секунд)

С помощью алгоритм задачи двух тел, можем получить начальные координаты и скорости спутника. Также был использован метод Ньютона для решения уравнения Кеплера. В итоге были получены следующие начальные координаты и скорости.

x	42122.835 км
y	0 км
z	0 км

\dot{x}	0 км/с
\dot{y}	1.66289164 км/с
\dot{z}	2.58980029 км/с

```
initial_coordinate, initial_velocity = calculate_initial_coordinate_velocity(
    semi_major_axis=SEMI_MAJOR_AXIS,
    eccentricity=ECCENTRICITY,
    longitude_of_ascending_node=LONGITUDE_OF_ASCENDING_NODE,
    argument_pericenter=ARGUMENT_OF_PERIAPSIS,
    inclination=INCLINATION,
    mean_anomaly=MEAN_ANOMALY
)
```

4. Подбор вариации

Требуется подбирать значения вариации для каждого компонента вектор положения и скорости. В итоге нужно построить график зависимости производной от вариации и выбрать значение вариации из середины плато на графике.

Алгоритм подбора вариации будет так:

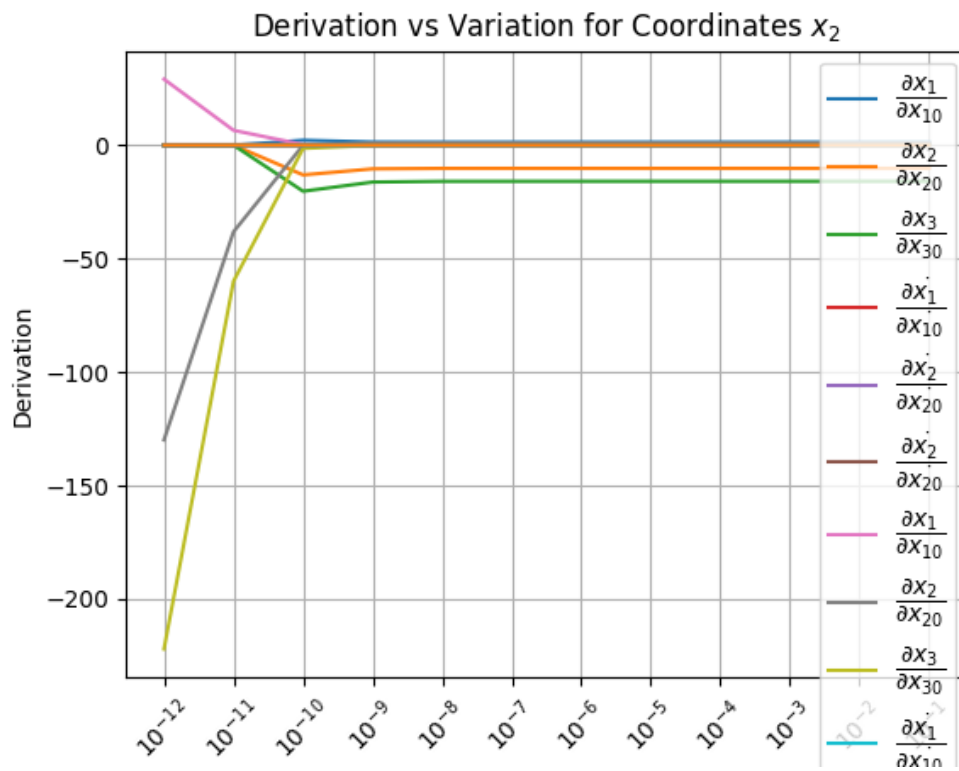
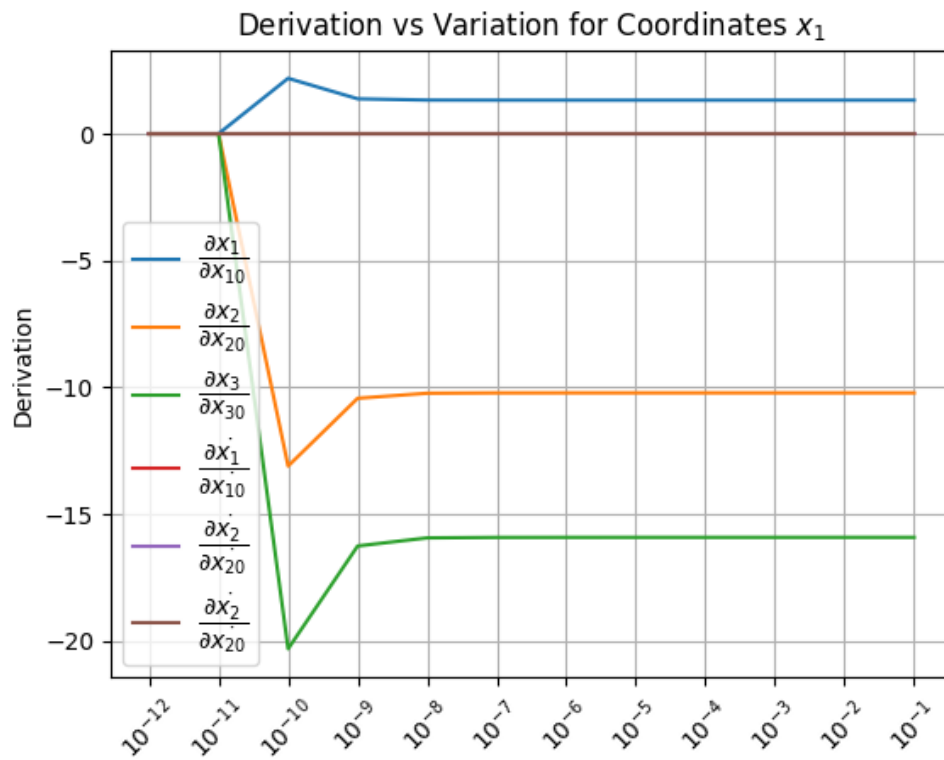
1. Вычисляем начальные координаты и скорости.
2. Вычисляем наблюдаемые координаты и скорости, передаем начальные координаты и скорости к интегратору.
3. Добавляем вариацию в начальные координаты и скорости по порядку. Вариация от 10^{-12} до 10^{-1} . Мы сначала добавляем эту вариацию в 1й компонент координат, далее в 2й компонент координат, далее в 3й координат, далее в 1й компонент скоростей, потом в 2й компонент скоростей и в 3й компонент скоростей. Выполняем итерационные процессы во всех значениях вариации.
4. В конце вычисляем производные от вариации на конечном моменте интегрирования. Формула:

$$\frac{\partial x_{j,N}}{\partial x_{i,0}} = \frac{x_{j,N}^{\text{var}(x_i)} - x_{j,N}}{\text{var}},$$

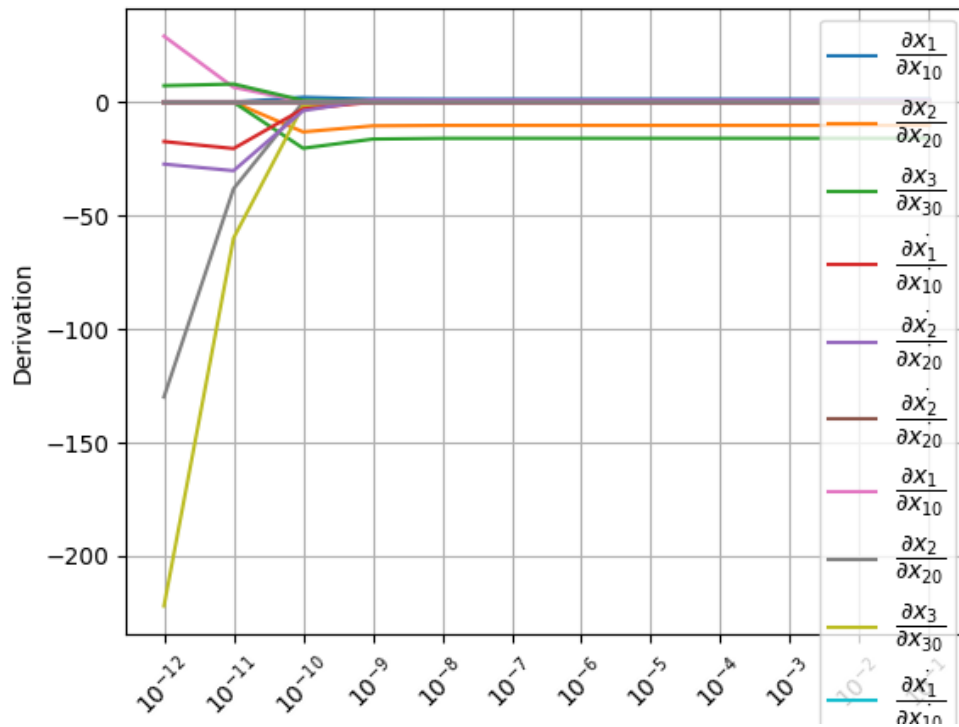
$$\frac{\partial \dot{x}_{j,N}}{\partial \dot{x}_{i,0}} = \frac{\dot{x}_{j,N}^{\text{var}(\dot{x}_i)} - \dot{x}_{j,N}}{\text{var}},$$

$$i = 1, 2, 3; j = 1, 2, 3$$

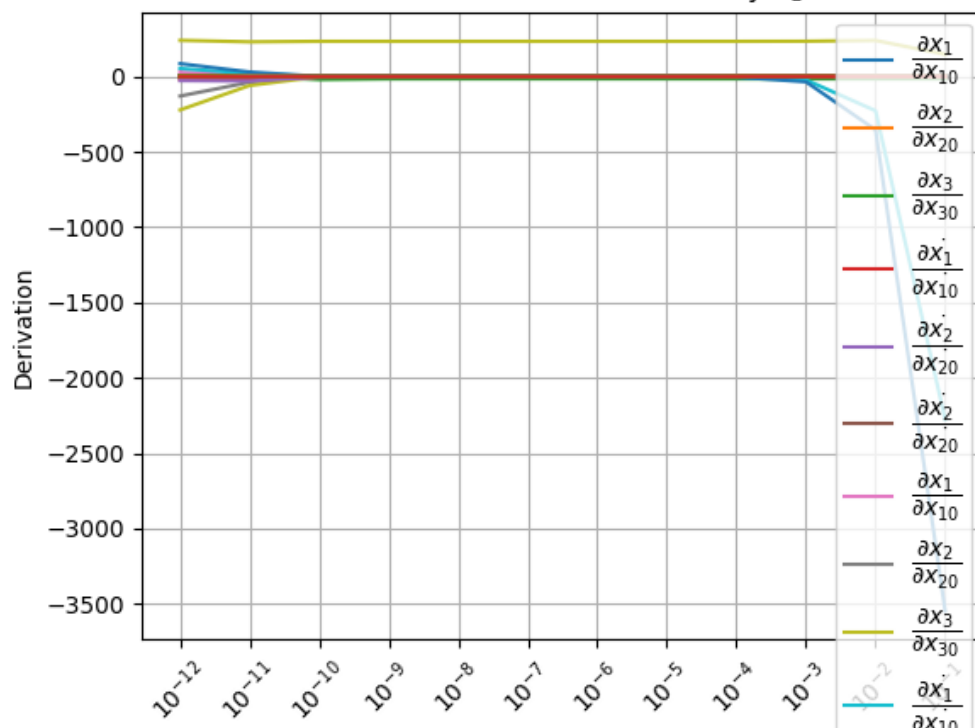
5. Построим график зависимостей.

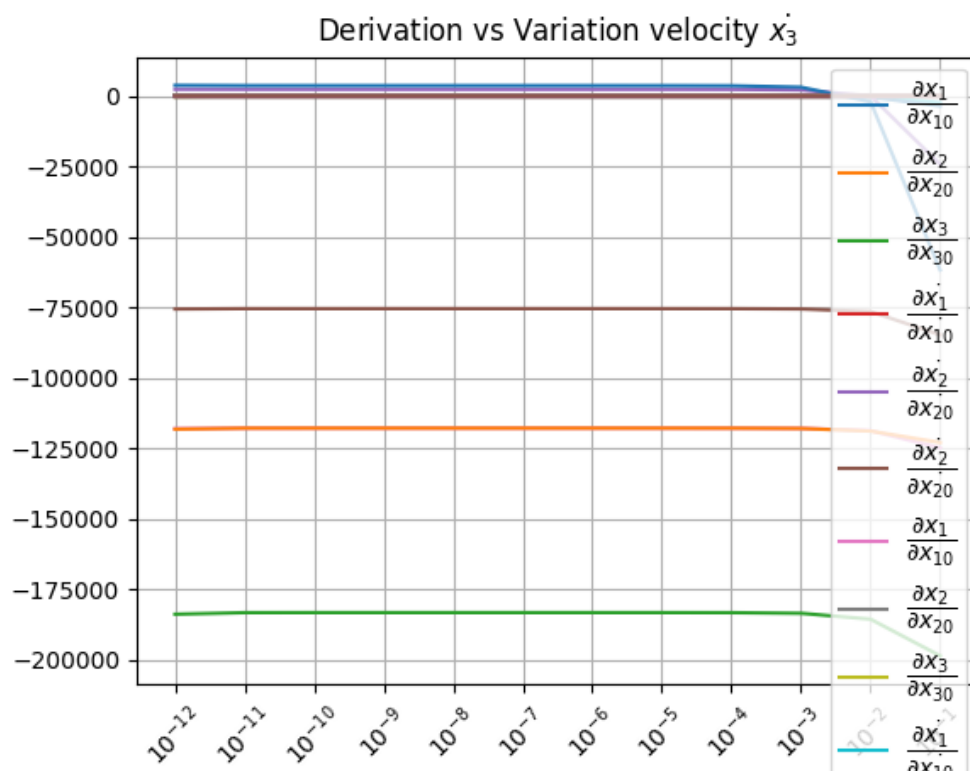
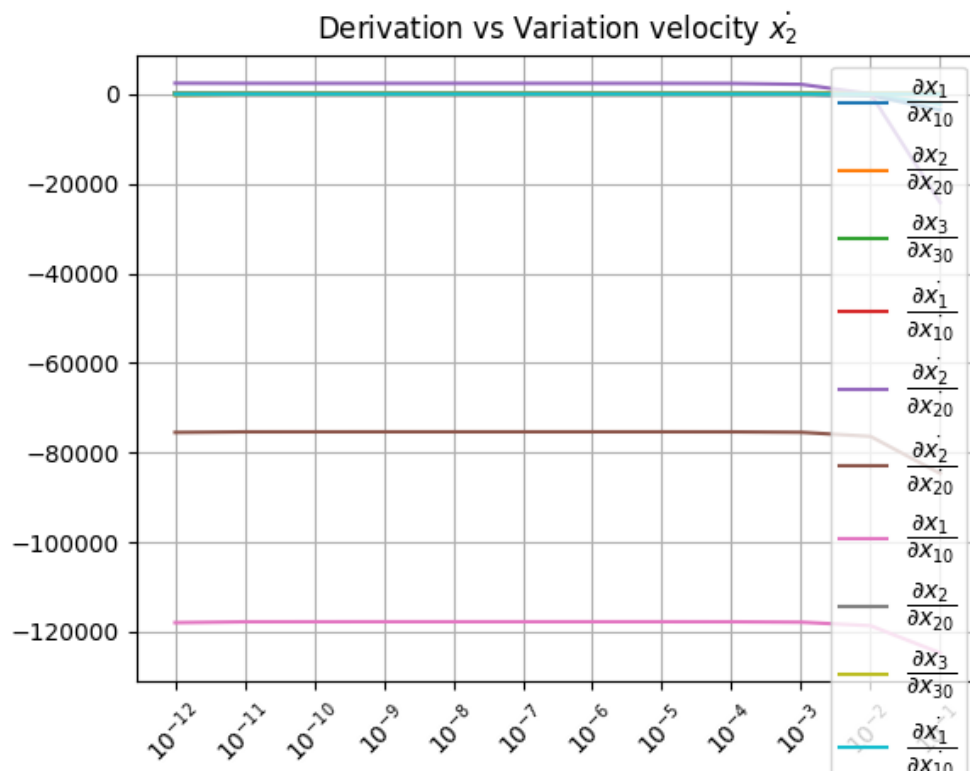


Derivation vs Variation for Coordinates x_3



Derivation vs Variation velocity \dot{x}_1





Отсюда мы можем сделать вывод что наилучшие значения вариации:

x	y	z	\dot{x}	\dot{y}	\dot{z}
10^{-4}	10^{-4}	10^{-4}	10^{-6}	10^{-6}	10^{-6}

5. Улучшение орбиты

Заданы x_{i0} и \dot{x}_{i0} — начальные вектор положения и вектор скорости спутника, $i=1,2,3$.

1. Моделируем наблюдения спутника x_{ik}^o с помощью «Численной модели движения ИСЗ» на заданные моменты t_k при заданных параметрах спутника, $k=1 \dots 100$.
2. Вносим ошибку в начальные параметры x_{i0}, \dot{x}_{i0} ,

$$\bar{x}_{i0} = x_{i0} + \Delta x_i,$$

$$\bar{\dot{x}}_{i0} = \dot{x}_{i0} + \Delta \dot{x}_i,$$

$$\Delta x_i = \xi \cdot 0.01'' \cdot \|\vec{x}_0\|,$$

$$\Delta \dot{x}_i = \xi \cdot 0.01'' \cdot \|\vec{\dot{x}}_0\|.$$

где,

ξ — случайная величина $[-1,1]$

$\|\vec{x}_0\|$ — норм вектор начального положения

$\|\vec{\dot{x}}_0\|$ — норм вектор начальной скорости

$0.01''$ — секунд дуги

Нужно перевести от секунды дуги в радиан.

$$\frac{0.01'' \cdot \pi}{180 \cdot 60 \cdot 60} \approx 4.84813681109536 \cdot 10^{-9}$$

3. С помощью «Численной модели движения ИСЗ» на заданные моменты t_k находим “вычисленные” — x_{ik}^c положения ИСЗ.
4. Далее находим “О-С” по следующей формуле

$$\Delta x_{ik}^{o-c} = x_{ik}^o - x_{ik}^c,$$

и образуем вектор столбец $c = \{\Delta x_{ik}^{o-c}\}$ по принципу

$$c = \{\Delta x_{11}^{o-c}, \Delta x_{21}^{o-c}, \Delta x_{31}^{o-c}, \Delta x_{12}^{o-c}, \Delta x_{22}^{o-c}, \Delta x_{32}^{o-c}, \dots, \Delta x_{1N}^{o-c}, \Delta x_{2N}^{o-c}, \Delta x_{3N}^{o-c}\}.$$

5. Вычисляем матрицу изохронных производных (матрицу А) методом вариации параметра.

$$A = \begin{pmatrix} \frac{\partial x_{11}}{\partial x_{10}} & \frac{\partial x_{11}}{\partial x_{20}} & \frac{\partial x_{11}}{\partial x_{30}} & \frac{\partial x_{11}}{\partial \dot{x}_{10}} & \frac{\partial x_{11}}{\partial \dot{x}_{20}} & \frac{\partial x_{11}}{\partial \dot{x}_{30}} \\ \frac{\partial x_{21}}{\partial x_{10}} & \frac{\partial x_{21}}{\partial x_{20}} & \frac{\partial x_{21}}{\partial x_{30}} & \frac{\partial x_{21}}{\partial \dot{x}_{10}} & \frac{\partial x_{21}}{\partial \dot{x}_{20}} & \frac{\partial x_{21}}{\partial \dot{x}_{30}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial x_{3N}}{\partial x_{10}} & \frac{\partial x_{3N}}{\partial x_{20}} & \frac{\partial x_{3N}}{\partial x_{30}} & \frac{\partial x_{3N}}{\partial \dot{x}_{10}} & \frac{\partial x_{3N}}{\partial \dot{x}_{20}} & \frac{\partial x_{3N}}{\partial \dot{x}_{30}} \end{pmatrix}.$$

где,

$$\frac{\partial x_{j,N}}{\partial x_{i,0}} = \frac{x_{j,N}^{\text{var}(x_i)} - x_{j,N}}{\text{var}},$$

$$\frac{\partial \dot{x}_{j,N}}{\partial \dot{x}_{i,0}} = \frac{\dot{x}_{j,N}^{\text{var}(\dot{x}_i)} - \dot{x}_{j,N}}{\text{var}},$$

$$i = 1, 2, 3; j = 1, 2, 3$$

6. Полученную систему уравнений $A \cdot y = c$ решаем методом наименьших квадратов. Для этого умножаем систему уравнений слева на A^T . Обозначим $Q = A^T A$, $d = A^T c$. Решение запишется в виде $y = Q^{-1} d$. y — искомая поправка. Эта поправка прибавляем в координат и скорости предыдущей итерации.
7. Находим среднеквадратическую ошибку веса σ_0 и среднеквадратические ошибки параметров σ_i по следующим формулам

$$\sigma_0 = \sqrt{\frac{\sum_{i=1}^{3N} (\Delta x_{ik}^{o-c})^2}{3N-6}},$$

$$\sigma_i = \sigma_0 q_{ii},$$

где q_{ii} — диагональные элементы матрицы Q^{-1} .

Выполняем итерационный процесс до тех пор, пока σ_0 ещё уменьшается.

Улучшаемые координаты:

№ итерации	x_0 (км)	y_0 (км)	z_0 (км)
1	42122.8345579391	0.000122019503834863	-0.000232407379506672
2	42122.8348434102	4.32333058035524e-05	-8.23383306411186e-05
3	42122.8349745231	7.03886740644574e-06	-1.34024899408809e-05
4	42122.834999219	2.16664431630754e-07	-4.12024710564794e-07
5	42122.8349999992	2.41726792730057e-10	-4.38131627576418e-10
6	42122.835	-9.34145379872516e-12	-9.19901284551027e-12

7	42122.8350000001	5.39721211399859e-12	7.56035448587494e-12
---	------------------	----------------------	----------------------

Улучшаемые скорости:

№ итерации	\dot{x}_0 (км/с)	\dot{y}_0 (км/с)	\dot{z}_0 (км/с)
1	8.96433835599589e-08	1.66289155027589	2.5898003365058
2	3.1754081490175e-08	1.66289161119811	2.58980030814492
3	5.16633327868511e-09	1.66289163917854	2.58980029511909
4	1.58371349308145e-10	1.66289164444868	2.58980029266555
5	1.55702111317304e-13	1.66289164461513	2.58980029258804
6	9.32951709860443e-16	1.6628916446153	2.58980029258796
7	-4.87521718579272e-16	1.6628916446153	2.58980029258796

Среднеквадратические ошибки координат:

№ итерации	σ_x	σ_y	σ_z
1	2.378185015083319e-05	0.00017955958971301692	0.00012217235570523584
2	7.975381838744087e-06	5.415783762576585e-05	5.979020262174674e-05
3	2.526937453658725e-06	7.848645001165896e-06	1.1933539875824529e-05
4	9.031476055062984e-08	2.3423781462995386e-07	3.8214348008903113e-07
5	9.264826721974822e-11	2.388989444112455e-10	3.906430859095589e-10
6	1.193241997185746e-12	3.0768252965555174e-12	5.031179747259355e-12
7	1.008900949778526e-11	2.6014920633426158e-11	4.2539217881850384e-11

Среднеквадратические ошибки скоростей:

№ итерации	$\sigma_{\dot{x}}$	$\sigma_{\dot{y}}$	$\sigma_{\dot{z}}$
1	3.8110519655192704e-13	3.924861544841164e-12	4.4733732656040333e-13
2	5.894292578484968e-14	4.4165142788685004e-13	6.468455001575365e-14
3	5.3684647582191567e-14	2.374010703158165e-14	1.2636960028606354e-14
4	2.130102818663165e-15	5.968156004562706e-16	4.453165560110484e-16
5	2.191655869599038e-18	6.061381815551188e-19	4.567785939860519e-19
6	2.822700254871579e-20	7.806537993353851e-21	5.882974110030915e-21
7	2.386626586714744e-19	6.600519314858057e-20	4.974124314556738e-20

Среднеквадратические ошибки веса:

№ итерации	σ_0
1	0.0036906247395490674
2	0.0013072938839609435
3	0.00021268360962512574
4	6.518033245702055e-06
5	6.6535302349899446e-09
6	8.56921830292502e-11
7	7.245374297292165e-10

Видно, что на 7-й итерации значения σ_0 уже увеличится по сравнению с предыдущей итерацией. То есть на 6-й итерации это наилучшие координаты и скорости.

6. Вывод

В данной работе были реализованы модули для работы с интегратором, написанным на фортране (включая процедура перезаписи входного файла), модули для подбора вариации и для решения задачи улучшения орбиты.

Данная работа была сделана на языке Python. Используемые библиотеки:

1. subprocess: Библиотека для запуска внешних процессов (.exe файл).
2. numpy: Библиотека для работы с многомерными массивами и математическими вычислениями.
3. matplotlib: Библиотека для построения графиков и визуализации данных.

Все коды выложены на GitHub, ссылка: <https://github.com/Steven2110/Orbit-Improvements>. Все описания файлов и инструкция запуска программы написаны в файле README.md.