

# Python Simulation Algorithms Report

Your Name

November 5, 2024

# Contents

# Chapter 1

## Introduction

This chapter provides an overview of the Python simulation algorithms implemented in this document.

## Chapter 2

# Hohmann Transfer Orbit Algorithm

The Hohmann transfer orbit algorithm is used to calculate the delta-v required for transferring from Earth to Mars.

### 2.1 Algorithm Overview

The algorithm calculates the required velocity change for the Hohmann transfer orbit.

### 2.2 Parameters and Constants

- $R_{\text{earth}}$ : Radius of Earth's orbit around the Sun in meters.
- $R_{\text{mars}}$ : Radius of Mars's orbit around the Sun in meters.
- $\mu_{\text{sun}}$ : Gravitational constant for the Sun in  $m^3/s^2$ .

### 2.3 Mathematical Model

The algorithm uses the following formulas:

$$v_{\text{earth}} = \sqrt{\frac{\mu_{\text{sun}}}{R_{\text{earth}}}} \quad (2.1)$$

$$v_{\text{mars}} = \sqrt{\frac{\mu_{\text{sun}}}{R_{\text{mars}}}} \quad (2.2)$$

$$v_{\text{transfer\_earth}} = \sqrt{2 \frac{\mu_{\text{sun}}}{R_{\text{earth}}} - \frac{\mu_{\text{sun}}}{R_{\text{transfer}}}} \quad (2.3)$$

$$v_{\text{transfer\_mars}} = \sqrt{2 \frac{\mu_{\text{sun}}}{R_{\text{mars}}} - \frac{\mu_{\text{sun}}}{R_{\text{transfer}}}} \quad (2.4)$$

$$\Delta v_{\text{earth}} = v_{\text{transfer\_earth}} - v_{\text{earth}} \quad (2.5)$$

$$\Delta v_{\text{mars}} = v_{\text{mars}} - v_{\text{transfer\_mars}} \quad (2.6)$$

$$\text{Total } \Delta v = |\Delta v_{\text{earth}}| + |\Delta v_{\text{mars}}| \quad (2.7)$$

## 2.4 Python Code

```

import numpy as np
import matplotlib.pyplot as plt

def calculate_hohmann_transfer(earth_orbit_radius, mars_orbit_radius):
    mu_sun = 1.32712440018e20

    v_earth = np.sqrt(mu_sun / earth_orbit_radius)
    v_mars = np.sqrt(mu_sun / mars_orbit_radius)

    transfer_orbit_semi_major_axis = (earth_orbit_radius + mars_orbit_radius) / 2

    v_transfer_at_earth = np.sqrt(2 * mu_sun / earth_orbit_radius - mu_sun / transfer_orbit_semi_major_axis)
    v_transfer_at_mars = np.sqrt(2 * mu_sun / mars_orbit_radius - mu_sun / transfer_orbit_semi_major_axis)

    delta_v_earth = v_transfer_at_earth - v_earth
    delta_v_mars = v_mars - v_transfer_at_mars

    total_delta_v = abs(delta_v_earth) + abs(delta_v_mars)

    return delta_v_earth, delta_v_mars, total_delta_v

```

## 2.5 Plots

Include any relevant plots or visualizations generated by the algorithm.