Trajectory Optimization for the Earth-Mars 2024 Transfer Window

Physics 304 - Final Project

Winn Koster, working with Liam Lynch*
Haverford College Department of Physics and Astronomy
(Dated: May 12, 2018)

For this final project we explore fascinating landscape of spacecraft trajectory optimization. While Kepler's Laws and the Newton Law of Gravitation have well defined and simply analytic solutions, things get more complicated when we consider optimization. In this work generate a "porkchop plot," which is helpful optimizing fuel usage of spacecraft. We outline our process from start to finish, including a discussion of the physics involved. Ultimately, the plots we generate are limited by computational expense, and serve as a proof of concept for what agencies such as NASA produce when planning any space mission.

1. TRAJECTORY OPTIMIZATION BACKGROUND

Let us first discuss some background on trajectory optimization and, specifically, the use of porkchop plots. This begins with a brief overview of orbital mechanics.

A spacecraft's endurance may be measured in terms of the amount it can change its velocity. This is usually referred to as Δv. This may be calculated using the Tsiolkovsky rocket equation (1), and is the most important factor in determining the performance of a spacecraft.

$$\Delta v \;=\; v_e \ln\big(\frac{m_i}{m_f}\big) \tag{1}$$

In the expression above, v_e is the exhaust velocity of the rocket (this is important due to Newton's Third Law), and m_i and m_f are the "wet" and "dry" masses of the rocket, respectively.

Moving from one orbit to another requires a certain change in velocity, and while the fuel requirement for such a maneuver might vary from rocket to rocket (depending on its mass ratio and efficiency), the Δv required is constant for any rocket. While the Δv requirement for a maneuver might be more or less constant at all times (for example, transferring from equatorial low Earth orbit to geostationary orbit always requires roughly 1.6 km sec^{-1}) the amount of Δv required to perform a more complicated maneuvers (eg: a trans martian injection burn) may vary depending on certain conditions (in this case, the relative positions of Earth and Mars). It is complicated cases such these that porkchop plots are useful.

Simply put, a porkhop plot depicts how much Δv (or fuel) a spacecraft must expend to obtain some trajectory or satisfy some condition. These are especially useful in planning interplanetary transfers, as these maneuvers have Δv requirements that drastically change depending on the relative location of the departure and arrival bodies. Figure 1 depicts an example porkchop plot for the 2005 Earth-Mars transfer window.

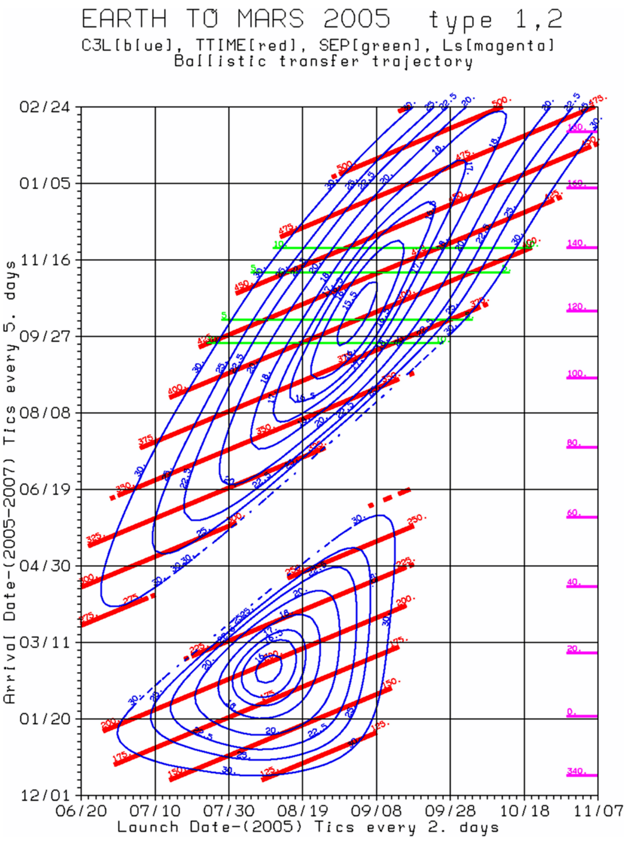

FIG. 1: This contour plot is a porkchop plot that depicts the energetic requirements for a transfer burn from Earth to Mars, depending on the departure date (x-axis) and the arrival date (y-axis). This plot opts to use characteristic energy, C_3, instead of Δv. We will explain the exact differences between these two unit choices later on, but it should be clear that converting between a velocity and kinetic energy is not a very in-depth process. Figure: [1].

Having introduced the general idea of a porkchop plot, we will now use the laws of physics to build a model that we can optimize using code.

*Electronic address: wkoster@haverford.edu

2. NEWTON'S LAW OF GRAVITATION

We will now begin with Newton's Law of Gravitation and work produce a set of coupled differential equations for the computer to simulate. We then set up a boundary value problem for the computer to optimize; for what initial velocity vector do we intercept Mars?

2.1. Expression in a Differential From

Start with Newtons Law of Gravitation expressed in its form as a second order differential equation:

$$\ddot{\vec{r}} = \frac{-\mu}{|\vec{r}|^2}\,\hat{r}. \tag{2}$$

We prefer to use Cartesian coordinates, so we decompose using the identities below.

$$\begin{aligned} x &= r\,\cos\phi\,\sin\theta \\ y &= r\,\sin\phi\,\sin\theta \\ z &= r\,\cos\theta \end{aligned}$$

Ultimately, we found the equations of motion in Cartesian coordinates to be

$$\ddot{x} = \frac{-\mu x}{(x^2+y^2+z^2)^{3/2}} \tag{3}$$

$$\ddot{y} = \frac{-\mu y}{(x^2+y^2+z^2)^{3/2}} \tag{4}$$

$$\ddot{z} = \frac{-\mu z}{(x^2+y^2+z^2)^{3/2}} \tag{5}$$

where μ represents GM, the standard gravitational parameter. We then made the following substitution to convert each second order differential equation into two first order differential equations, which may be solved numerically:

$$dx = v \tag{6}$$

$$dv = \frac{-\mu x}{(x^2+y^2+z^2)^{3/2}}. \tag{7}$$

2.2. Computational Implementation

We wrote a code to iterate through the differentials, and checked for consistency. We used a Runge-Kutta 4th order method, and tested using two scenarios: one simulated earth year, and the Earth-Mars Hohmann transfer (both assuming circular orbits). We found 6 or more significant figure consistency for position, velocity, and orbital energy conservation for $N = 1,000,000$ steps, although we determined that $N = 40,000$ was sufficient for our cases.

Figures 2 and 3 depict our testing scenarios. We compared these simple cases with analytical solutions to determine simulation accuracy.

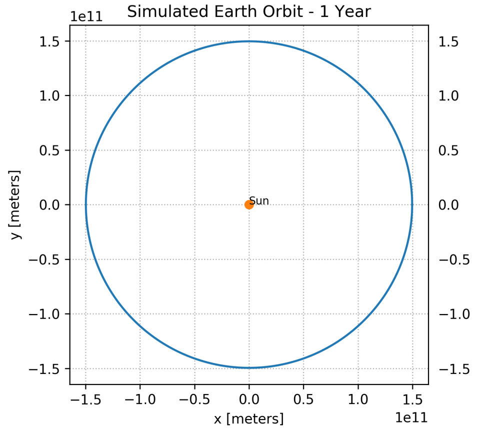

FIG. 2: A plot depicting the simulated orbital trajectory for our Earth year test scenario. Own work.

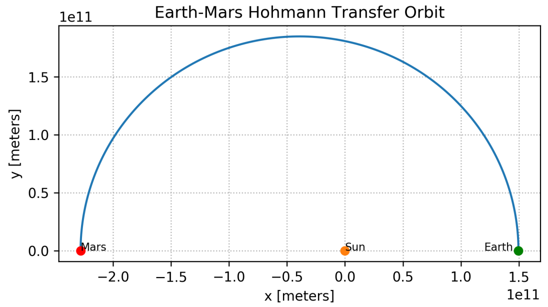

FIG. 3: A plot depicting the simulated orbital trajectory for our Earth-Mars Hohmann transfer test scenario. This simulated later provided a reference to compare with analytical solutions. Own work.

3. SOLVING THE BOUNDARY VALUE PROBLEM

We now seek to solve the boundary value problem that is at the core of this analysis. If we have Earth position at some initial time, $\vec{r_E}(t_1)$, and Mars position at some final time, $\vec{r_M}(t_2)$, what initial velocity vector, $\vec{v_i}$, connects these two points in the time elapsed, such that we obey the differential $\ddot{\vec{r}} = \frac{-\mu}{|\vec{r}|^2}\,\hat{r}$ at all times? (We then ask what is the Δv expenditure, but this comes later).

To reiterate, we start with the following fixed constraints:

$$\begin{aligned}\vec{r_i} &= \vec{r}_{Earth}(t_1)\\ \vec{r_f} &= \vec{r}_{Mars}(t_2)\\ \Delta t &= t_2 - t_1\\ \ddot{\vec{r}} &= \frac{-\mu}{|\vec{r}|^2}\,\hat{r},\end{aligned}$$

and wish to find some initial conditions vector, (v_x, v_y, v_z) that satisfies all of the above.

3.1. Hohmann Transfer Analysis

The Hohmann transfer is an ideal case for comparison with our simulation, as the time of flight and velocity change required may be calculated analytically. Throughout this project, we used the Earth-Mars simplified Hohmann transfer case as a benchmark for our simulation. By simplified, we mean that we have assumed circular, co-planar orbits. This is required to make the transfer solvable analytically.

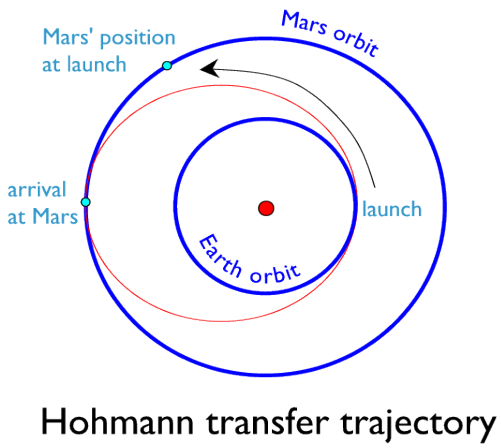

FIG. 4: This schematic depicts the geometry of the Hohmann transfer orbit, which is often the most efficient means of transfer between two co-planar orbits. In this case we start at Earth's orbit, use some Δv to accelerate into a trans-Martian injection orbit, and ultimately cross paths with Mars 180^o from the initial position. Our plot, shown as Figure 3, depicts the result of our simulation on this same trajectory. Note that an additional burn is required to match velocities with Mars and fall into a planetary orbit. Figure: [2].

To begin solving the Hohmann transfer case analytically, we use Kepler's Third Law, which relates an orbital period to its semi-major axis:

$$T = 2\pi\sqrt{\frac{a^3}{GM}}. \qquad (8)$$

We then note that for the transfer burn, the semi-major axis is exactly the average of those of Earth and Mars. $a_{Hohmann} = (a_{Earth} + a_{Mars})/2$, where $a_{Earth} = 1.496 \times 10^{11}$ m, and $a_{Mars} = 2.279 \times 10^{11}$ m. So we find the Hohmann semi-major axis to be:

$$a_H = 1.888 \times 10^{11} \text{ m}.$$

Further noting that the transfer is exactly one half of an orbital period (dividing period equation above by two), we find an elapsed time:

$$\Delta t = 22,368,649 \text{ sec} \approx 258 \text{ days}.$$

In order to determine the velocities required, we use the vis-viva equation, shown.

$$v^2 = GM\left(\frac{2}{r} - \frac{1}{a}\right) \qquad (9)$$

Note that G, M, a are all constant for the Hohmann case, so the only variable is r. Plugging in the Earth's semi major axis, we find the initual velocity (relative to the Sun) to be

$$|v_i| = 32,735 \text{ m s}^{-1}.$$

One could consider the differences between the Earth and Sun rest frames to calculate the true Δv – it's not as simple as subtracting the Earth's velocity from the number above – but we can revisit that later. For now, we may work in the Sun's rest frame and compare the computer simulation to this analytical solution.

3.2. Hohmann Transfer Simulation

Ultimately, we knew that we would use the "shooting method" to converge on valid solutions to this boundary value problem. Later on, we show an optimization algorithm that we used to converge on numerical solutions relatively quickly. But at the early stages of this project, we wanted to get a more complete sense of the landscape of initial conditions. To accomplish this, we used a grid sampler method to try thousands of different initial conditions (ie: initial velocities) for the Hohmann Transfer case. Figure 5 depicts the landscape of possible solutions.

We were also interested in knowing how sensitive the solution was to changes in the initial conditions. We wanted to know this for two reasons: first, we wanted to verify that using a smaller number of steps in our Runge-Kutta simulation was not introducing any artificial noise into the solution, and second, we wanted to have a sense of how close to Mars is "close enough." That is, we were hoping to determine what a reasonable stop criterion would be for an optimization algorithm.

To that end, we plotted another grid plot, showing the exact numerical solution and the velocity space immediately around it.

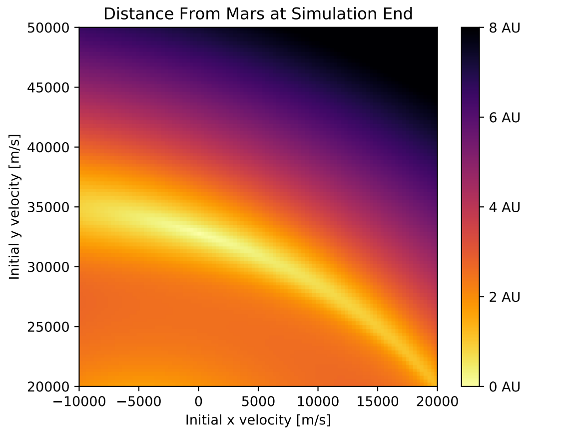

FIG. 5: This is a 100×100 grid plot of possible initial conditions in the v_x, v_y solution space (z was suppressed for this example). The analytical solution of $|v_i| \to v_y = 32,735$ m s^{-1} is clearly shown as an absolute minima in this plot. Ultimately, we used an algorithm to locate the minima *without* running $10,000$ individual orbit simulations, as it took a new MacBook Pro running all night to generate this plot. Own work.

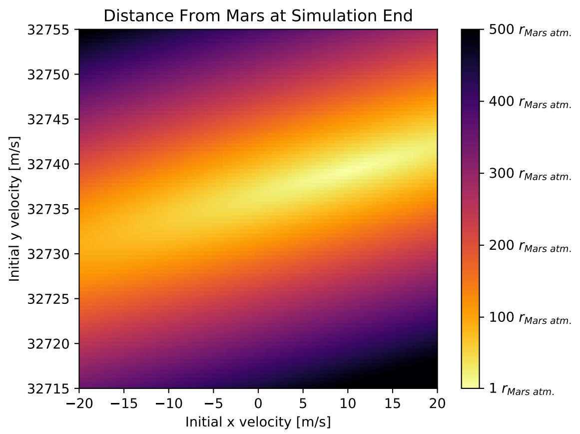

FIG. 6: This is a 100×100 grid plot of possible initial conditions in the v_x, v_y solution space immediately around the numerical solution (z was suppressed for this example). Here we have defined an "atmospheric radius" as the radius of Mars (3390 km) plus the orbit altitude of the Mars Reconnaissance Orbiter (255 km). This plot shows that a small change of just 10 m s^{-1} can throw off the end result by hundreds of thousands of kilometers. For reference, Mars's closer moon orbits at less than 10 atmospheric radii, and the Mars-Sun L1 point is at about 300 atmospheric radii. Own work.

4. OPTIMIZATION ALGORITHM

Figures 5 and 6 gave us a lot of intuition for the topography of the solution space, but each of these plots took eight hours to run, and amount to only one data point on a porkchop plot with potentially hundreds or thousands of points. To speed up the process, we needed to use an optimization algorithm.

The SciPy package [3] contains several optimization algorithms, and we decided to use the Nelder-Mead algorithm (alternatively known as the simplex method), as it is most useful for systems with unknown gradients and sharp topography.

While the algorithm as a whole was ideal for our problem, we found one issue with the SciPy implementation of the algorithm. As written in SciPy, it is not possible to define a stop criterion for distance. That is, we could not tell the computer "when you've converged to 10,000 km, stop optimizing, this is good enough." Left unhindered, the algorithm converges to within centimeters of the Mars position vector, but obviously we do not need this kind of accuracy, and this level of precision took 5 to 10 times longer than a simple convergence to a few thousand kilometers.

It is, however, possible to define a maximum number of iterations in SciPy, so we wrote a loop that runs the algorithm for 100 iterations, checks to see whether it has suitably converged to a radius of our choosing (we ultimately chose 100,000 km), and runs another 100 iterations if it has not.

Using our modified algorithm, we were able to optimize a function that returned our spacecraft distance from mars at the end of simulation. That is, we found a solution vector, $\vec{v_i} = (v_x, v_y, v_z)$, that satisfies the constraint

$$f(\vec{v_i}, \Delta t) \le 100,000 \text{ km}.$$

5. GENERATING NUMERICAL SOLUTIONS

We are now in a position where we may compare the results of of our optimization algorithm to the analytical solution we described earlier. Recall that we had previously found the initial velocity (in the Sun's rest frame) to be $|v_i| = 32,735$ m s^{-1}. Using our optimization algorithm, we found the solution vector to be $(10.5, 32730.6, -2.7)$, which we are satisfied with.

5.1. Determining Δv Requirements using Characteristic Energy

Up to now we have worked exclusively in the Sun's rest frame to perform calculations. However, calculating the Δv requirement is not as simple as subtracting the Earth's velocity vector from the spacecraft velocity vector.

While our simulation does not account for the Earth's gravitation, we must account for the escape velocity from the Earth from an energetic standpoint. The difference between the spacecraft velocity vector and the Earth velocity vector may be thought of as how much faster than Earth's *escape velocity* the spacecraft is travelling. That is, once the spacecraft is suitably far from the Earth such that Earth's gravity is negligible, it must carry this initial velocity vector to travel to Mars.

It is now useful to introduce the concept of characteristic energy, or C_3. Characteristic energy is the excess velocity at infinity, squared. It has units of energy, although it is not exactly surplus kinetic energy, as it is off by a factor of two.

Suppose we place a spacecraft at *exactly* escape velocity of the Earth. It will continue on its parabolic trajectory, trading kinetic energy for gravitational potential energy. If the spacecraft is at *exactly* escape velocity, then its velocity at infinity, v_{inf}, would be zero. Likewise, $C_3 = 0$. Now suppose that after we placed the spacecraft at escape velocity, it performed a burn, gaining additional velocity, and obtaining a *hyperbolic* trajectory. In this case, $v_{\text{inf}} > 0$, and $C_3 > 0$. And so, C_3 is a measurement of a spacecraft's excess energy compared to the escape energy of the body it is orbiting.

Let us now introduce the relationship between Δv and C_3:

$$\Delta v = \sqrt{C_3 + \frac{2GM}{r_i}} - \sqrt{\frac{GM}{r_i}}. \tag{10}$$

The total burn requirement, Δv, is equal to the excess velocity at infinity plus the escape velocity for the body, added in quadrature, minus the initial orbital velocity (assuming a circular orbit). Here, we have assumed that our spacecraft is already in some orbit with radius r_i; this calculation does not work from launch, only circular orbit.

For our specific departure case, then, we can express this equation as

$$\Delta v = \sqrt{C_3 + \frac{2GM_e}{r_{LEO}}} - \sqrt{\frac{GM_e}{r_{LEO}}}, \tag{11}$$

where M_e is the mass of the earth, and r_{LEO} represents a low Earth orbit, which we defined $7,000$ m s^{-1}, which corresponds to an altitude of approximately 207 km. The beauty of this calculation is that this number is inherently modular. If we wanted to, for example, assume a departure from the International Space Station, we need only insert the ISS altitude of 400 km, $r_i = 6,771$ km, and the Δv requirements are updated accordingly.

5.2. Hohmann Transfer Test Case

Using these numbers, we found the following for our numerically optimized Hohmann Transfer, which are consistent with sources we found online for simplified (ie: co-planar circular orbits) transfer cases [4, 5]:

Earth Departure Burn:	3617 m s^{-1}
Mars Arrival Burn:	2107 m s^{-1}
Total:	5714 m s^{-1}

These numbers are not, however, fully consistent with NASA numbers (cf. Figure 1, as well as others), which indicate that the Hohmann transfer injection burn is closer to the $4-5$ km s^{-1} range. We attribute this discrepancy to the relative inclination of the two orbits, and found that NASA technical manuals agree with our assessment. Depending on the location of the ascending and descending nodes (relative to the perihelion and aphelion of the injection burn), the Δv required for the Hohmann transfer can vary *considerably* across an approximately 17 year period [6].

And thus, we may no longer rely on our basic Hohmann transfer test case. From here on out, we must incorporate real Earth and Mars data, and optimize numerically, trusting that since our code works for the analytically checkable case, it must work for other cases as well.

5.3. Numerical Calculations using Ephemeris Data

After doing some research to determine when exactly the 2024 transfer window takes place, we used the JPL HORIZONS ephemeris tool [7] for data. Initially, we selected Earth data from July 1, 2024, through December 31, 2024, and Mars data from November 28, 2024, through February 21, 2026, at daylong intervals, although we quickly realized that this was too much data to be viable, and that the edges of the data set contained nothing interesting.

We then cut down arrival data to March 15, 2025 through October 1, 2025, keeping the same departure data to run from July 2024 through December 31, 2024, sampling every 20 days. The porkchop plots for Earth departure, Mars arrival, and total Δv are included, although we believe there may be an issue with the Mars Δv data (and thus the total as well). The Mars data appears to be a scalar multiple of the Earth data, which is unexpected. We can find nothing wrong with our code, but we do not feel that the Mars data is correct. There is nothing to suggest that the Earth plot is flawed however, as it resembles other plots [6].

In the plots following, the grey areas in the top left indicate cells that did not converge within 1,000 algorithm iterations, indicating that they got stuck in a loop.

6. OUTLOOK

Overall, this has been an incredibly rewarding project. I learned quite a bit about orbital mechanics, differential equations, numerical simulation, and algorithms. I learned the strengths and weaknesses of several algorithms, and learned to work around several things I couldn't fix on my own.

Looking back, there are definitely more analytic steps we could have taken to cut down on computer time. We more or less just told the computer how Newton's Law of Gravitation worked, and told it to figure out everything

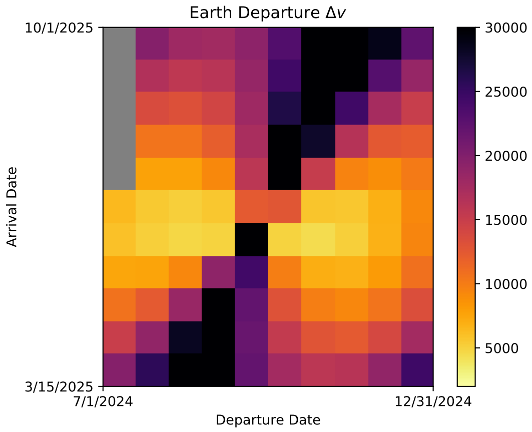

FIG. 7: This plot depicts the Δv required to place a spacecraft on a trans-Martian injection orbit from a low-Earth orbit 200 km parking orbit. The grey cells in the top left corner did not suitably converge within the number of samples allowed.

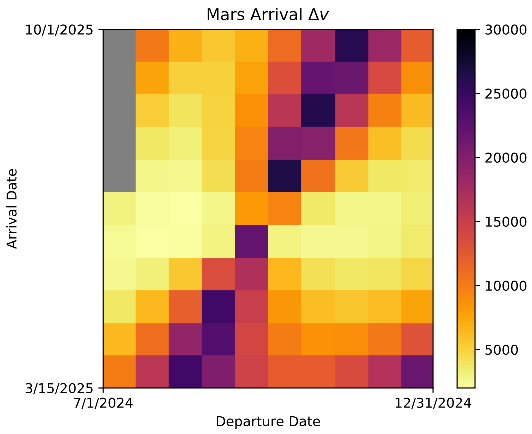

FIG. 8: This plot depicts the Δv required to place a spacecraft in a 255 km Martian orbit after arrival at Mars. The grey cells in the top left corner did not suitably converge within the number of samples allowed. If a spacecraft were able to perform an aerobraking maneuver through Mars's atmosphere, the majority of the Δv shown here could be saved. We do not believe this plot is accurate, as it appears to simply be a scalar multiple of the Earth porkchop plot. We do not expect this to be the case in reality, however.

else. But I'm sure that getting creative with the orbital mechanics might yield some simplifications that could reduce computational expense. I'll have to keep my eye out for this going forward.

7. CONCLUSION

In this project we have generated porkchop plots for the Earth-Mars 2024 transfer window. Using RK-4 simulation methods, a Nelder-Mead optimization algorithm, and extreme computational expense, we have shown the Δv requirements for a variety of departure and arrival scenarios.

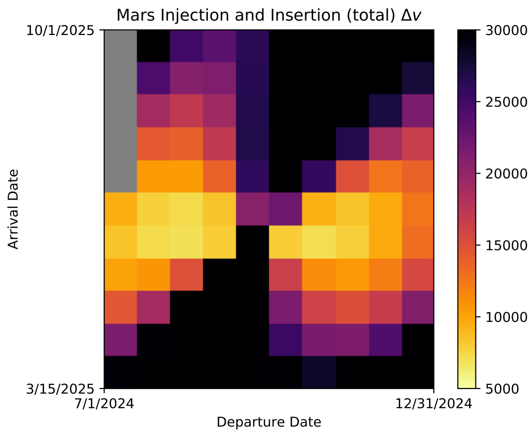

FIG. 9: This total plot represents the sum of the two previous plots plots. Since we do not believe the Mars plot to be correct, this plot, then, is also not correct. We have shown this here for completeness. If a spacecraft were unable to perform an aerobraking maneuver (eg: manned spacecraft have never attempted this), this plot would represent the Δv required to orbit Mars. Leaving Mars, however, is another story.

[1] N. Aeronautics and S. Administration, *"porkchop" is the first menu item on a trip to mars* (2005), URL https://mars.jpl.nasa.gov/spotlight/porkchopAll.html.

[2] W. B. Hubbard, *Lecture notes - planetary sciences 206* (2014), URL https://www.lpl.arizona.edu/~hubbard/PtyS206/Lectures1/Jan21.htm.

[3] E. Jones, T. Oliphant, P. Peterson, et al., *SciPy: Open source scientific tools for Python* (2001–), [Online; accessed 5/10/18], URL http://www.scipy.org/.

[4] /u/CuriousMetaphor, *Delta-v map of the solar system* (2013), submitted by a Reddit user, full name name unknown. Online; accessed 5/11/18], URL https://www.reddit.com/r/space/comments/1ktjfi/deltav_map_of_the_solar_system/.

[5] en:User:Wolfkeeper, *Mars/moon/earth delta-vs* (2008), submitted by a Wikipedia user, full name name unknown. Online; accessed 5/11/18], URL https://en.wikipedia.org/wiki/Delta-v_budget#

/media/File:Delta-Vs_for_inner_Solar_System.svg.
[6] L. George and L. Kos, Tech. Rep. NASA/TM 1998 208533, National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama 35812 (1998), URL http://www.ltas-vis.ulg.ac.be/cmsms/uploads/File/InterplanetaryMissionDesignHandbook.pdf.
[7] N. Solar System Dynamics Group, JPL, *HORIZONS online ephemeris system (web interface)*, https://ssd.jpl.nasa.gov/?horizons.