AERO4800

# Checkpoint 2: A Novel Way to Get to Mars

**Flynn Devoy**

Student Number: 47442043

October 10, 2025

## Executive Summary

This report presents the development and application of a numerical modelling framework to evaluate interplanetary transfer trajectories from Earth to Mars, with specific emphasis on the use of atmospheric aerocapture as a mission optimisation strategy. The analysis, implemented in Python using the `hapsira` astrodynamics package, models complete Earth–Mars transfers from a 400 km circular parking orbit to a 300 km Mars orbit. The methodology includes planetary ephemeris generation, Lambert transfer solutions, hyperbolic excess and $\Delta v$ calculations, and comparative assessment of propulsive and aerocapture-assisted capture strategies. Validation against Curtis (*Orbital Mechanics for Engineering Students*, Example 8.8) confirmed excellent agreement in both orbital geometry and mission $\Delta v$ results.

The study extends this validated model to the upcoming 2026–2027 launch opportunity, generating $\Delta v$ and propellant "porkchop" plots to quantify performance across the launch window. Results show that aerocapture can reduce total mission $\Delta v$ from approximately 8.4 km/s to 4.0 km/s, saving over 5 tonnes of propellant for a 20 tonne spacecraft and enabling delivery of nearly twice the payload mass compared to a fully propulsive mission. Additionally, the optimal aerocapture transfer departs one week later than the equivalent propulsive case for the same arrival date, offering improved schedule flexibility and operational robustness.

While the aerocapture model assumes an idealised exponential atmosphere and neglects aerodynamic lift, thermal, and guidance constraints, it captures the dominant energy-exchange mechanisms driving entry dynamics. The results clearly demonstrate the strategic value of aerocapture for high-mass or crewed Mars missions—reducing fuel requirements, expanding payload capacity, and improving mission resilience within a given launch window. Overall, the investigation highlights aerocapture as a practical and efficient solution for next-generation Mars transfer architectures.

# Contents

# 1 Interplanetary Travel to Mars Including Aerocapture

## 1.1 Introduction

This section presents a complete end-to-end model of an Earth–to–Mars transfer from a circular Earth parking orbit to a captured Mars orbit, evaluated for both purely propulsive and aerocapture-assisted cases. The methodology was developed in Python using the `hapsira` astrodynamics package and follows the framework outlined in the AERO4800 assignment task sheet by James [1]. Model validation was performed against the benchmark case from Curtis [2] (*Example 8.8*), which considers an Earth–Mars Hohmann-type transfer between 7 Nov 1996 and 12 Sep 1997 (time of flight = 309 days).

The model simulates Lambert transfers between heliocentric positions of Earth and Mars, computes departure and arrival hyperbolic excess velocities, and evaluates the corresponding insertion $\Delta v$ with and without atmospheric braking. For aerocapture, a simplified energy–balance approach is used to represent drag-induced deceleration and post-entry trimming into a 300 km circular parking orbit.

**Key input parameters:**

$$m_0 = 2000\,\text{kg}, \quad I_{sp} = 450\,\text{s}, \quad h_{\text{parking}} = 300\,\text{km}. \tag{1}$$

**Modelling assumptions (applied throughout the assignment):**

- **Two-body dynamics:** All interplanetary and planetocentric motions are modelled as two-body problems, neglecting third-body perturbations, solar radiation pressure, and non-spherical gravity.
- **Coplanar Lambert solutions:** Transfers are computed between heliocentric positions of Earth and Mars at specified epochs using Lambert's problem, assuming impulsive burns at departure and arrival.
- **Aerocapture modelling:** Aerocapture is represented as an instantaneous energy loss sufficient to produce a bound elliptical orbit with periapsis at the entry altitude $r_{p,\text{entry}}$ and apoapsis near the desired final orbit. The subsequent circularisation $\Delta v$ is applied impulsively at apoapsis as the difference between the elliptical and circular orbital velocities.
- **Spacecraft and propulsion:** Constant specific impulse $I_{sp} = 450$ s and initial mass $m_0 = 2000$ kg are assumed for all cases; no staging or thrust–gravity losses are modelled.

**Model limitations:**

- **Atmospheric effects:** Variations in density, winds, and dust storms at Mars are neglected. The model assumes a uniform atmosphere capable of providing the required deceleration.
- **Thermal protection and GNC:** Aerothermal heating, TPS mass, and entry–guidance requirements are not modelled. Real vehicles must satisfy thermal and deceleration constraints and maintain entry–corridor accuracy.
- **Energy dissipation idealisation:** Perfectly efficient and stable aerocapture is assumed; no dispersions or over/under-shoot errors are modelled. In reality, residual propulsive $\Delta v$ is required for trimming and correction.
- **No environmental perturbations:** Effects from solar radiation pressure, planetary oblateness ($J_2$), and out-of-plane dynamics are neglected.
- **Performance margins:** Results represent idealised minimum $\Delta v$ and propellant requirements; practical missions require additional margin for contingencies, attitude control, and trajectory correction manoeuvres.

Overall, this modelling framework provides a physically consistent but computationally efficient tool for comparing the relative performance of propulsive versus aerocapture-assisted Mars transfers and assessing their influence on mission $\Delta v$, propellant mass, and schedule flexibility.

## 1.2 Planetary States (Ephemerides)

To begin the transfer analysis, the heliocentric positions and velocities of Earth and Mars must be determined at the selected departure and arrival dates. These state vectors are obtained from high-accuracy planetary ephemerides, which provide tabulated or interpolated data of celestial bodies over time. In astrodynamics problems, these serve as the initial boundary conditions for trajectory design: they specify *where the planets are* and *how they are moving* at the start and end of the transfer window.

Mathematically, the states are expressed as:

$$\big(\mathbf{R}_{\oplus}(t), \mathbf{V}_{\oplus}(t)\big),\ \big(\mathbf{R}_{\mathrm{Mars}}(t), \mathbf{V}_{\mathrm{Mars}}(t)\big), \tag{2}$$

where $\mathbf{R}$ denotes the heliocentric position vector and $\mathbf{V}$ the heliocentric velocity vector of the respective planet.

In the Python implementation, the `hapsira` library interfaces with `astropy` to generate these state vectors from built-in ephemerides:

```
earth_ephem = Ephem.from_body(Earth, ts)
mars_ephem = Ephem.from_body(Mars, ts)
earth_orbit = Orbit.from_ephem(Sun, earth_ephem, date_launch)
mars_orbit = Orbit.from_ephem(Sun, mars_ephem, date_arrival)
```

These states provide the necessary inputs to Lambert's problem, ensuring that the transfer orbit computed later is dynamically compatible with the true planetary motions.

## 1.3 Lambert Solution for the Heliocentric Transfer

Once the planetary states at departure and arrival are known, the next step is to determine the trajectory that links them. This is formulated as Lambert's problem: given two position vectors $\mathbf{R}_1$ and $\mathbf{R}_2$ at two specified times, and the elapsed time $t_{12}$ between them, find the unique conic orbit that connects the points under a central gravitational field. The solution provides the required heliocentric velocities at departure and arrival:

$$(\mathbf{V}_1^{(h)}, \mathbf{V}_2^{(h)}) = \mathrm{Lambert}\big(\mathbf{R}_1, \mathbf{R}_2, t_{12}\big). \tag{3}$$

In the Python code, the `Maneuver.lambert` function is used to construct the transfer orbit. This method applies a numerical solver to the Lambert boundary-value problem and produces both the maneuver and the resulting conic orbit:

```
man_to_mars = Maneuver.lambert(earth_orbit, mars_orbit)
transfer_orbit, _ = earth_orbit.apply_maneuver(man_to_mars, intermediate=True)
```

The output of this calculation is the heliocentric velocity vector immediately after Earth departure, $\mathbf{V}_1^{(h)}$, and the heliocentric velocity vector upon Mars arrival, $\mathbf{V}_2^{(h)}$. These vectors fully describe the transfer trajectory in the Sun-centered frame and will be validated if the correct hyperbolic excess velocities are calculated.

## 1.4 Earth Departure: Hyperbolic Excess and $\Delta v$

Once the heliocentric transfer orbit has been established, the next task is to determine the velocity the spacecraft must achieve relative to Earth in order to inject into this orbit. This is quantified by the *hyperbolic excess velocity*, $v_\infty$, which represents the residual velocity a spacecraft would have at infinite distance from Earth after escaping its gravitational well.

The hyperbolic excess vector at Earth departure is found by subtracting the heliocentric velocity of Earth from the required transfer velocity at departure:

$$\mathbf{v}_\infty^{\mathrm{dep}} = \mathbf{V}_1^{(h)} - \mathbf{V}_{\oplus}, \qquad v_\infty^{\mathrm{dep}} = \left\|\mathbf{v}_\infty^{\mathrm{dep}}\right\|. \tag{4}$$

This ensures that the spacecraft leaves Earth's sphere of influence with exactly the additional velocity needed to match the transfer orbit computed in Section 3.

In the implementation, the departure velocity from the transfer orbit is compared directly with Earth's heliocentric velocity at the same epoch:

```
r_dep, v_dep = transfer_orbit.rv()
_, v_earth = earth_orbit.rv()
v_inf_dep = np.linalg.norm((v_dep - v_earth).to(u.km/u.s))
```

The next step is to relate this excess velocity to a practical $\Delta v$ requirement from a circular low Earth orbit (LEO). At a given parking orbit radius $r_0$, the circular speed and the local escape speed are:

$$v_{\text{circ}} = \sqrt{\frac{\mu_\oplus}{r_0}}, \tag{5}$$

$$v_{\text{esc}} = \sqrt{\frac{2\mu_\oplus}{r_0}}. \tag{6}$$

The spacecraft must depart on a hyperbolic trajectory such that its velocity at perigee is large enough to both escape Earth's gravity and supply the additional $v_\infty$ required for the interplanetary transfer. This is expressed by the Earth departure burn:

$$\Delta v_{\text{Earth}} = \sqrt{(v_\infty^{\text{dep}})^2 + v_{\text{esc}}^2} - v_{\text{circ}}. \tag{7}$$

Physically, this formula accounts for the fact that the hyperbolic perigee velocity combines the local escape speed with the hyperbolic excess, and the difference relative to the circular velocity in LEO is the actual impulse required from the launch vehicle or upper stage. This is implemented in the code as shown below:

```
mu_earth = Earth.k.to(u.km**3/u.s**2)
r_leo = (Earth.R + h_parking).to(u.km)
v_circ = np.sqrt(mu_earth / r_leo)
v_esc = np.sqrt(2 * mu_earth / r_leo)
dv_earth = np.sqrt(v_inf_dep**2 + v_esc**2) - v_circ
```

## 1.5 Mars Arrival: Hyperbolic Excess

After the interplanetary cruise, the spacecraft reaches the vicinity of Mars on the trajectory defined by the Lambert solution. Just as at Earth departure, the key parameter that governs the capture requirements is the hyperbolic excess velocity at arrival. This quantity represents the residual velocity the spacecraft has relative to Mars after accounting for the planet's heliocentric motion.

The arrival hyperbolic excess vector is obtained by subtracting Mars's heliocentric velocity from the transfer orbit velocity at the time of encounter:

$$\mathbf{v}_\infty^{\text{arr}} = \mathbf{V}_2^{(h)} - \mathbf{V}_{\text{Mars}}, \qquad v_\infty^{\text{arr}} = \|\mathbf{v}_\infty^{\text{arr}}\|. \tag{8}$$

This expresses the relative velocity between the spacecraft and Mars as the spacecraft crosses into Mars's sphere of influence. Its magnitude is the key input for determining whether the spacecraft can be captured into orbit by Mars, and if so, how much energy must be dissipated either by propulsive braking or by atmospheric drag in the case of aerocapture.

In the code implementation, this calculation is carried out by propagating the transfer orbit forward to the arrival epoch, then subtracting Mars's velocity:

```
r_arr, v_arr = transfer_orbit.propagate(date_arrival).rv()
_, v_mars = mars_orbit.rv()
v_inf_arr = np.linalg.norm((v_arr - v_mars).to(u.km/u.s))
```

## 1.6 Mars Capture: No Aerocapture

When the spacecraft approaches Mars with a hyperbolic excess velocity, its trajectory relative to the planet is a hyperbola. The speed at periapsis of this hyperbolic trajectory can be derived from the vis-viva equation applied to the two-body Mars–spacecraft system:

$$v_{\text{peri,hyperbola}} = \sqrt{(v_\infty^{\text{arr}})^2 + \frac{2\mu_{\text{Mars}}}{r_p}}, \qquad v_{\text{circ}}(r_p) = \sqrt{\frac{\mu_{\text{Mars}}}{r_p}}. \tag{9}$$

Here, $r_p$ is the chosen periapsis radius, $\mu_{\text{Mars}}$ is the standard gravitational parameter of Mars, $v_{\text{peri,hyperbola}}$ is the spacecraft's velocity at periapsis on the hyperbolic approach, and $v_{\text{circ}}$ is the velocity required for a circular orbit at the same radius.

The capture $\Delta v$ is then the difference between these two velocities. This represents the propulsive braking required to transition from the hyperbolic flyby trajectory into a circular bound orbit:

$$\Delta v_{\text{Mars,no aero}} = v_{\text{peri,hyperbola}} - v_{\text{circ}}(r_p). \tag{10}$$

These formulas were implemented in the python code as below:

```
r_mars = (Mars.R + 300 * u.km).to(u.km)
mu_mars = Mars.k.to(u.km**3/u.s**2)
v_circ_mars = np.sqrt(mu_mars / r_mars)
v_hyp_peri = np.sqrt(v_inf_arr**2 + 2 * mu_mars / r_mars)
dv_mars_no_aero = v_hyp_peri - v_circ_mars
dv_mars_aero = 0.0 * u.km/u.s
```

With $r_p = R_{\text{Mars}} + 300\,\text{km}$, this gives:

$$\Delta v_{\text{Mars,no aero}} = 2.171\,\text{km/s}.$$

This means that without atmospheric interaction, the spacecraft must perform a retrograde burn of roughly 2.2 km/s at periapsis to slow down enough to be captured by Mars's gravity.

## 1.7 Mars Capture: Aerocapture

In the aerocapture strategy, the spacecraft uses atmospheric drag during a carefully targeted pass through the upper Martian atmosphere to dissipate hyperbolic excess energy and convert the incoming hyperbolic trajectory into a bound elliptical orbit. The subsequent manoeuvre is a small trim burn to circularise the orbit at the desired parking altitude. This section outlines the simplified methodology used in the present model, which is implemented in the function `aerocapture_trim_dv`.

### Step 1: Hyperbolic periapsis velocity

The periapsis speed of the incoming hyperbolic approach, already defined in Eq. 9, is reused here as the initial condition for the aerocapture analysis.

```
v_peri_hyp = np.sqrt(v_inf.to(u.km/u.s).value**2 +
                    2*mu.value/rp_entry.to(u.km).value)
```

### Step 2: Post-aerocapture elliptical orbit

It is assumed that aerocapture yields a bound ellipse with periapsis at $r_{p,\text{entry}}$ and apoapsis approximately at the target parking orbit radius $r_{p,\text{final}}$. The semi-major axis is

$$a = \frac{r_{p,\text{entry}} + r_{p,\text{final}}}{2}. \tag{11}$$

```
rp = rp_entry.to(u.km).value
ra = rp_final.to(u.km).value
a = (rp + ra) / 2.0
```

### Step 3: Velocity at apoapsis

The spacecraft velocity at apoapsis is

$$v_{\text{apo}} = \sqrt{\mu\left(\frac{2}{r_{p,\text{final}}} - \frac{1}{a}\right)}. \tag{12}$$

```
v_apo = np.sqrt(mu.value * (2/ra - 1/a))
```

### Step 4: Circular orbit velocity

The circular velocity at the target radius is the same expression already used for Mars capture.

```
v_apo_new = np.sqrt(mu.value / ra)
```

### Step 5: Residual trim $\Delta v$

Finally, the required impulsive correction at apoapsis is

$$\Delta v_{\text{trim}} = |v_{\text{circ}} - v_{\text{apo}}| \,. \tag{13}$$

```
dv_trim = abs(v_apo_new - v_apo) * u.km/u.s
```

This $\Delta v_{\text{trim}}$ represents the effective propulsive cost of aerocapture. While in the idealised case $\Delta v_{\text{trim}} \to 0$, in practice tens to hundreds of m/s are often required due to dispersions, imperfect guidance, and energy dissipation errors.

### Numerical Drag Integration Model

Atmospheric deceleration during aerocapture was simulated using a numerical integration of the planar equations of motion under Mars gravity and drag forces:

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} - \frac{1}{2}\frac{\rho(v)C_D A}{m} v\vec{v}, \qquad \rho(h) = \rho_0 e^{-h/H},$$

where $\rho_0 = 0.020$ kg/m$^3$ and $H = 11.1$ km. For a nominal spacecraft of $m = 2000\,\text{kg}$, $C_D = 0.7$, and $A = 15\,\text{m}^2$, the ballistic coefficient is $\beta = m/(C_D A) \approx 1.9 \times 10^5$ kg/m$^2$.

```
rho0, H = 0.020, 11100.0 # kg/m^3, m
beta = m.to(u.kg).value / (Cd * A.to(u.m**2).value)
```

At each timestep, gravitational and drag accelerations are computed as follows:

```
def deriv(t, y):
    x, y_pos, vx, vy = y
    r = np.hypot(x, y_pos); v = np.hypot(vx, vy)
    h = r - Rm

    ax_g = -mu * x / r**3
    ay_g = -mu * y_pos / r**3

    if 0 < h < 120e3:
        rho = rho0 * np.exp(-h/H)
        a_drag = 0.5 * rho * v**2 / beta
        ax_d, ay_d = -a_drag * vx / v, -a_drag * vy / v
    else:
        ax_d = ay_d = 0.0

    return [vx, vy, ax_g + ax_d, ay_g + ay_d]
```

The integration terminates when the spacecraft exits the upper atmosphere at $h = 120\,\text{km}$:

```
def exit_event(t, y):
    return np.hypot(y[0], y[1]) - Rm - 120e3
exit_event.terminal, exit_event.direction = True, 1

sol = solve_ivp(deriv, [0, 5000], y0, max_step=1.0, events=exit_event)
```

The solution provides the complete trajectory and deceleration history from atmospheric entry (80 km) to exit (120 km), typically showing a gradual velocity loss of several hundred m/s. Although the model assumes constant $C_D$, no lift, and an exponential static atmosphere, it effectively captures the key physics of drag-induced energy dissipation during aerocapture.

## 1.8 Totals and Propellant Mass

With the Earth departure and Mars arrival requirements quantified, the overall mission $\Delta v$ budget can be assembled.

For a fully propulsive capture scenario, the total mission cost is the sum of the Earth departure burn and the Mars capture burn. By contrast, with aerocapture the cost reduces to the Earth departure burn plus only a small trim $\Delta v$ to circularise at the final parking orbit.

These totals represent the cumulative velocity change that must be supplied by the spacecraft's propulsion system over the course of the interplanetary transfer. The large reduction in required $\Delta v$ highlights the potential benefit of aerocapture: by using the Martian atmosphere to dissipate most of the arrival energy, the mission avoids a major propulsive maneuver at Mars.

To connect these $\Delta v$ budgets to practical design, the Tsiolkovsky rocket equation is used:

$$\Delta v = I_{sp} g_0 \ln \frac{m_0}{m_f}, \tag{14}$$

where $m_0$ is the initial mass, $m_f$ the final mass after the burn, $I_{sp}$ the specific impulse, and $g_0$ standard gravity. This allows direct estimation of propellant mass consumption for each mission strategy, illustrating how even modest changes in $\Delta v$ can translate into substantial differences in propellant fraction.

$$\Delta v = I_{sp} g_0 \ln \frac{m_0}{m_f} \quad \Rightarrow \quad m_p = m_0 - m_f = m_0 \left(1 - e^{-\Delta v/(I_{sp} g_0)}\right). \tag{15}$$

This is used in the code as shown below:

```
g0 = 9.81 * u.m / u.s**2
def mass_after_burn(m0, dv, Isp):
    return m0 * np.exp(-dv.to(u.km/u.s).value / (Isp * g0).to(u.km/u.s).value)

dv_total_no_aero = dv_earth + dv_mars_no_aero
mf_no_aero = mass_after_burn(m0, dv_total_no_aero, Isp)
prop_no_aero = m0 - mf_no_aero

dv_total_aero = dv_earth + dv_mars_aero
mf_aero = mass_after_burn(m0, dv_total_aero, Isp)
prop_aero = m0 - mf_aero
```

Here $m_0$ is the initial mass of the spacecraft (including propellant), $m_f$ the final mass after the burn, $I_{sp}$ the specific impulse of the propulsion system, $g_0$ the standard gravitational acceleration, and $m_p$ the propellant mass consumed. This equation captures the exponential penalty associated with increasing $\Delta v$: even modest increases in velocity demand translate to disproportionately higher propellant requirements.

## 1.9 Results and Validation against Curtis Example 8.8

The developed model was validated against the analytical case in Curtis, *Orbital Mechanics* (Example 8.8), which considers an Earth–Mars transfer from November 7, 1996 to September 12, 1997 (time of flight 309 days). Validation proceeds in two steps: first, the transfer orbit elements are compared, and second, the mission $\Delta v$ and propellant requirements are assessed.

**Transfer Orbit Elements**

Representative orbital elements of the heliocentric transfer are shown in Table 1 for both the Curtis example and the Python `hapsira` implementation.

Table 1: Comparison of transfer orbit elements: Curtis Example 8.8 vs `hapsira` output

| Element | Curtis (Ecliptic frame) | Code Output (J2000 Equatorial) |
|---|---|---|
| Semi-major axis, $a$ [km] | $1.8475 \times 10^8$ | $1.8473 \times 10^8$ |
| Eccentricity, $e$ | 0.20581 | 0.20413 |
| Inclination, $i$ [°] | 1.662 | 24.65 |
| RAAN, $\Omega$ [°] | 44.90 | 2.85 |
| Argument of periapsis, $\omega$ [°] | 19.97 | 61.19 |
| True anomaly, $\nu$ [°] | 340.04 | $-18.39$ |

The semi-major axis and eccentricity match closely, confirming that the transfer orbit shape is consistent with the analytical solution. The angular elements (inclination, RAAN, argument of periapsis, true anomaly) differ significantly. This discrepancy arises because Curtis presents the orbit in a two-dimensional ecliptic frame, while the `hapsira` implementation calculates elements in the full J2000 equatorial frame using planetary ephemerides. As a result, the orientation parameters differ, but the underlying transfer geometry remains consistent.

**Mission $\Delta v$ and Propellant Validation**

The model computes the hyperbolic excess velocities, departure and capture $\Delta v$, and propellant requirements. A direct comparison with Curtis's analytical results is shown in Table 2. Agreement in $v_\infty$ at Earth and Mars (errors of 1–3%) validates the transfer solution, while the additional quantities follow consistently from these values.

Table 2: Comparison of Analytical Example and Code Outputs (with percentage error)

| Parameter | Analytical (Curtis Ex. 8.8) | Code Output | Error |
|---|---|---|---|
| $\lvert\mathbf{v}_\infty^{\text{dep}}\rvert$ | 3.166 km/s | 3.200 km/s | 1.1% |
| $\lvert\mathbf{v}_\infty^{\text{arr}}\rvert$ | 2.885 km/s | 2.812 km/s | 2.5% |
| $\Delta v_{\text{Earth}}$ | 3.674 km/s | 3.659 km/s | 0.41% |
| $\Delta v_{\text{Mars,no aero}}$ | – | 2.171 km/s | – |
| $\Delta v_{\text{Mars,aero}}$ | – | 0.043 km/s | – |
| $\Delta v_{\text{total,no aero}}$ | – | 5.830 km/s | – |
| $\Delta v_{\text{total,aero}}$ | – | 3.702 km/s | – |
| Propellant (no aero) | – | 1466.1 kg | – |
| Propellant (aero) | – | 1135.3 kg | – |
| Time of Flight | 309 days | 309.0 days | 0.0% |

The comparison confirms that the model reproduces Curtis's transfer geometry with small error. Furthermore, the results quantify the large propellant savings possible with aerocapture, which removes the $\sim$2.2 km/s Mars capture burn and reduces propellant mass consumption by over 300 kg for the chosen spacecraft parameters.

## 1.10 Flow Chart of Task 1 Code

To illustrate the logic of the implemented Python model, a flow chart is presented in Figure 1. The chart shows the sequential steps from inputs, through ephemerides and Lambert's problem, to the calculation of Earth departure and Mars arrival $\Delta v$ values, with separate branches for aerocapture and no-aerocapture cases. Finally, the total $\Delta v$, propellant mass, and validation against Curtis Example 8.8 are obtained.

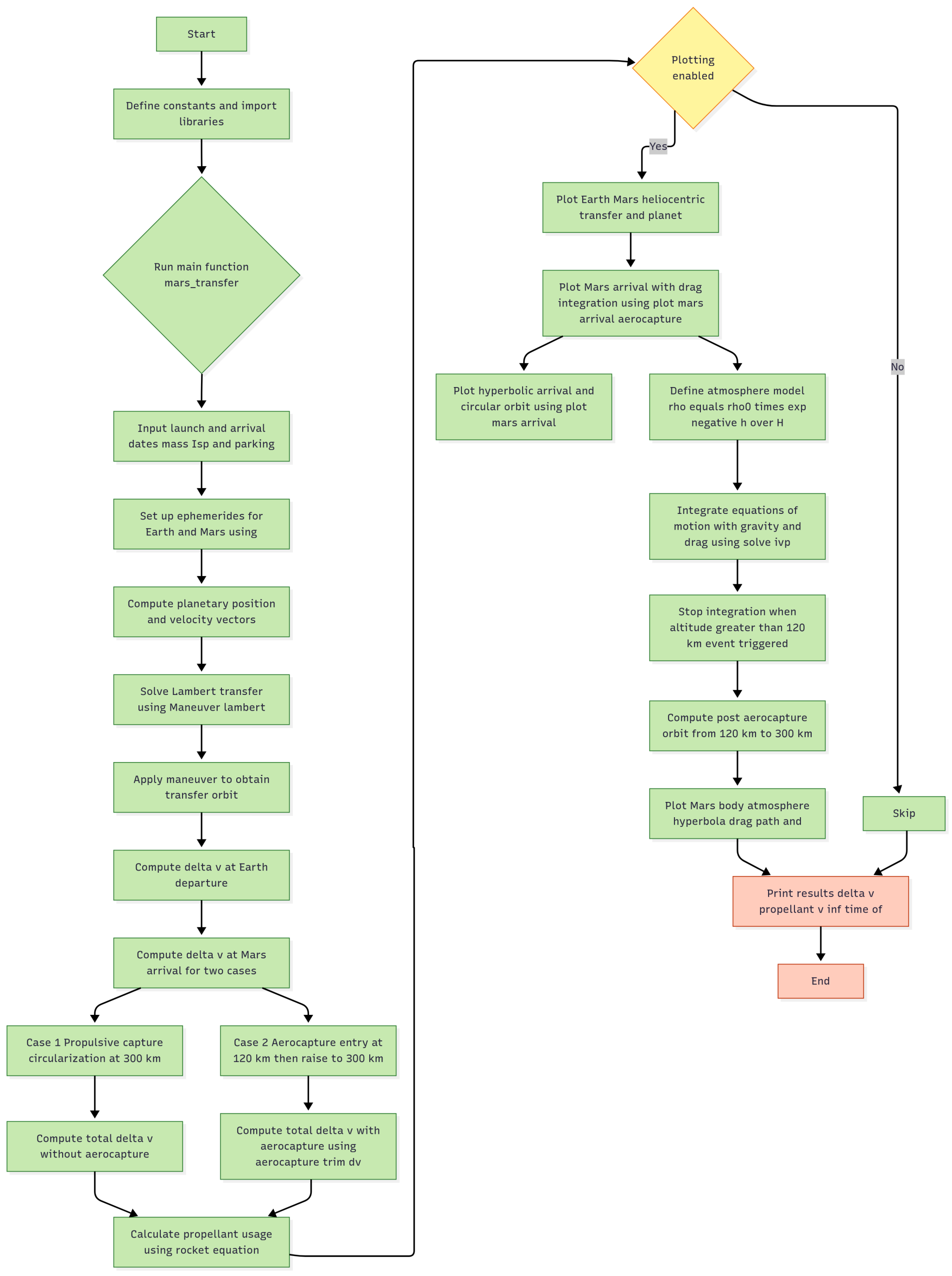
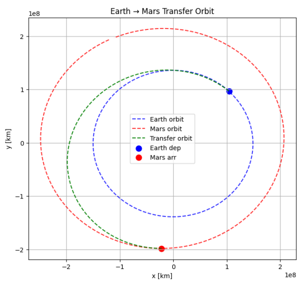

Start
Define constants and import libraries
Run main function mars_transfer
Input launch and arrival dates mass Isp and parking
Set up ephemerides for Earth and Mars using
Compute planetary position and velocity vectors
Solve Lambert transfer using Maneuver lambert
Apply maneuver to obtain transfer orbit
Compute delta v at Earth departure
Compute delta v at Mars arrival for two cases
Case 1 Propulsive capture circularization at 300 km
Case 2 Aerocapture entry at 120 km then raise to 300 km
Compute total delta v without aerocapture
Compute total delta v with aerocapture using aerocapture trim dv
Calculate propellant usage using rocket equation
Plotting enabled
Yes
No
Plot Earth Mars heliocentric transfer and planet
Plot Mars arrival with drag integration using plot mars arrival aerocapture
Plot hyperbolic arrival and circular orbit using plot mars arrival
Define atmosphere model rho equals rho0 times exp negative h over H
Integrate equations of motion with gravity and drag using solve ivp
Stop integration when altitude greater than 120 km event triggered
Compute post aerocapture orbit from 120 km to 300 km
Plot Mars body atmosphere hyperbola drag path and
Skip
Print results delta v propellant v inf time of
End

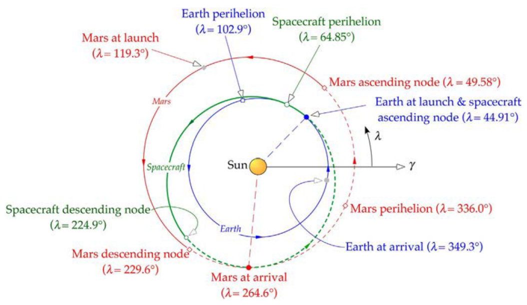
Figure 1: Flowchart outlining the structure of the `mars_transfer()` simulation code. The program sets up planetary ephemerides, solves Lambert's problem to obtain the transfer orbit, computes $\Delta v$ at Earth and Mars for both propulsive and aerocapture cases, and plots the resulting heliocentric and Mars-centered trajectories.

## 1.11 Task 1 Plots

The following figures present the outputs of the code developed for Task 1 of Checkpoint 2.



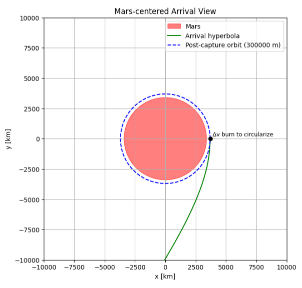
(a) Heliocentric Earth–Mars transfer orbit from the Python implementation.


Earth perihelion ($\lambda$ = 102.9°)
Spacecraft perihelion ($\lambda$ = 64.85°)
Mars at launch ($\lambda$ = 119.3°)
Mars ascending node ($\lambda$ = 49.58°)
Earth at launch & spacecraft ascending node ($\lambda$ = 44.91°)
Mars
$\lambda$
Spacecraft
Sun
$\gamma$
Spacecraft descending node ($\lambda$ = 224.9°)
Mars perihelion ($\lambda$ = 336.0°)
Earth
Mars descending node ($\lambda$ = 229.6°)
Earth at arrival ($\lambda$ = 349.3°)
Mars at arrival ($\lambda$ = 264.6°)

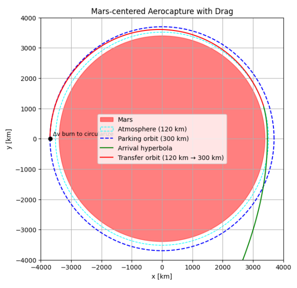
(b) Analytical Earth–Mars transfer from Curtis Example 8.8.

Figure 2: Comparison of heliocentric Earth–Mars transfer trajectories from the Python implementation (left) and the analytical solution in Curtis Example 8.8 (right). The strong geometric similarity confirms the model's validity.

Figure 2 shows the side-by-side comparison of the heliocentric transfer orbits generated numerically and analytically. The local arrival conditions are further examined in Figure 3.



(a) Propulsive capture: a retrograde burn at periapsis circularises the orbit.



(b) Aerocapture: atmospheric drag removes excess energy before a trim $\Delta v$.

Figure 3: Local planetocentric arrival trajectories at Mars. Comparison between (a) propulsive braking and (b) aerocapture with a small trim burn to achieve a 300 km parking orbit.

Figure 3 compares the two capture strategies at Mars: a purely propulsive burn versus aerocapture followed by a small circularisation manoeuvre.

# 2 How Aerocapture Will Allow Us to Get More Mass to Mars

## 2.1 Introduction

In this section, we investigate how aerocapture can be used to increase the mass which may be delivered into orbit around Mars. Starting from a 20 tonne spacecraft in a 400 km circular parking orbit at Earth, we compare two mission architectures: one that relies on a fully propulsive capture at Mars, and one that employs aerocapture to eliminate the Mars orbit insertion burn. By avoiding the large $\Delta v$ penalty typically associated with Mars capture, aerocapture has the potential to significantly reduce propellant consumption and therefore allow more of the spacecraft mass to be delivered as payload.

The aim of this analysis is to determine how much additional payload mass can be transported to Mars in the upcoming 2026–2027 launch window when aerocapture is used, relative to the conventional propulsive approach. Porkchop plots are generated to visualise the total $\Delta v$ requirements across the launch and arrival date space, and corresponding fuel requirement plots are used to evaluate the mass delivered in both scenarios. The optimal launch and arrival dates are then selected to maximise payload delivery, and the results are supported with heliocentric transfer orbits and Mars-centred arrival plots to illustrate the trajectory geometry.

This section therefore provides both a quantitative and qualitative assessment of the benefits of aerocapture for interplanetary transfers, linking trajectory design, mass fraction analysis, and mission feasibility.

## 2.2 Methodology

### 2.2.1 Spacecraft Assumptions

The baseline vehicle is a 20 tonne spacecraft initially inserted into a 400 km circular parking orbit about Earth, as obtained from the results of Checkpoint 1. The propulsion system is modelled with a realistic high-efficiency cryogenic engine, with a specific impulse of $I_{sp} = 450$ s (representative of the RL10 engine family). Gravitational acceleration is taken as $g_0 = 9.81$ m/s$^2$, and the Tsiolkovsky rocket equation is used to calculate the required propellant mass for each manoeuvre.

### 2.2.2 Launch Window Selection and Optimisation Process

To identify the most effective transfer opportunities, a sweep was performed across the 2026–2027 Earth–Mars launch window. The departure epoch was varied between August and December 2026 in 7-day increments, while the corresponding arrival epoch was varied between June and November 2027. This span was chosen to encompass the October 2026 launch opportunity highlighted in recent literature as a promising minimum-energy transfer [3]. A minimum time-of-flight constraint of 50 days was imposed to exclude unrealistically short trajectories.

For each departure–arrival pair, the transfer problem was solved using Lambert's method to obtain the required $\Delta v$ at both Earth departure and Mars arrival. Two cases were considered: a fully propulsive capture manoeuvre at Mars and an aerocapture case where the orbit insertion burn is replaced with atmospheric braking. The results of each case were stored in four arrays: total $\Delta v$ with and without aerocapture, and the associated propellant mass required, computed using the Tsiolkovsky rocket equation.

Listing 1: Initialisation of launch/arrival date arrays and storage matrices.

```
dep_dates = Time(np.arange("2026-08-01", "2026-12-01", dtype="datetime64[7D]"))
arr_dates = Time(np.arange("2027-06-01", "2027-11-01", dtype="datetime64[7D]"))

DV_no_aero = np.zeros((len(dep_dates), len(arr_dates)))
DV_aero = np.zeros((len(dep_dates), len(arr_dates)))
FUEL_no_aero = np.zeros((len(dep_dates), len(arr_dates)))
FUEL_aero = np.zeros((len(dep_dates), len(arr_dates)))
```

The sweep procedure was automated using a Python routine. Each iteration called the `mars_transfer()` function, which returned the $\Delta v$ components and propellant consumption for a given date pair:

Listing 2: Looping through departure and arrival dates to evaluate $\Delta v$ and fuel mass.

```
for i, d_dep in enumerate(dep_dates):
    for j, d_arr in enumerate(arr_dates):
        if d_arr > d_dep + 50*u.day: # enforce minimum flight time
            res = mars_transfer(d_dep.iso, d_arr.iso,
                                m0=m0, Isp=Isp, h_parking=h_parking,
                                plot=False, mars_plot=False, porkchop=False)

            DV_no_aero[i,j] = res["dv_total_no_aero"].to(u.km/u.s).value
            DV_aero[i,j] = res["dv_total_aero"].to(u.km/u.s).value
            FUEL_no_aero[i,j] = res["propellant_no_aero"].to(u.kg).value
            FUEL_aero[i,j] = res["propellant_aero"].to(u.kg).value
```

Finally, the results were assembled into contour plots, commonly referred to as *porkchop plots*. Two sets of porkchop plots were generated:

1. **$\Delta v$ porkchops:** Contours of total mission $\Delta v$ as a function of departure and arrival date.
2. **Fuel porkchops:** Contours of required propellant mass (in tonnes).

Listing 3: Example of porkchop plotting with calendar date axes.

```
cs1 = ax1.contourf(dep_dates.plot_date, arr_dates.plot_date, DV_no_aero.T,
                   levels=np.arange(3,9,0.5), cmap="viridis")
plt.colorbar(cs1, ax=ax1, label="Total␣$\Delta$v␣[km/s]")
ax1.xaxis.set_major_formatter(mdates.DateFormatter("%Y-%m-%d"))
ax1.yaxis.set_major_formatter(mdates.DateFormatter("%Y-%m-%d"))
```

The plots were formatted with calendar dates on both axes for clarity, using the `matplotlib.dates` library. This enabled visual identification of the optimal launch period and provided a direct comparison of how aerocapture improves payload delivery by eliminating the Mars capture $\Delta v$.

## 2.3 $\Delta v$ and Fuel Porkchop Analysis

To visualise the variation in mission performance across the 2026–2027 Earth–Mars launch window, porkchop plots were generated for both $\Delta v$ requirements and corresponding fuel consumption. Figures 4 and 5 show the results for the propulsive capture and aerocapture cases.

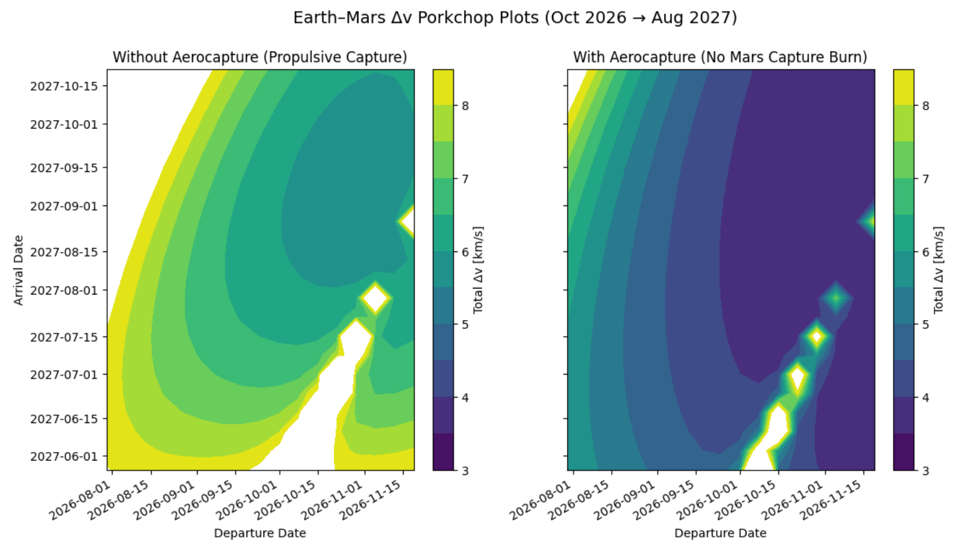


Figure 4: Total mission $\Delta v$ porkchop plots for the August–December 2026 departure window and June–November 2027 arrival window. Left: propulsive capture at Mars. Right: aerocapture at Mars.

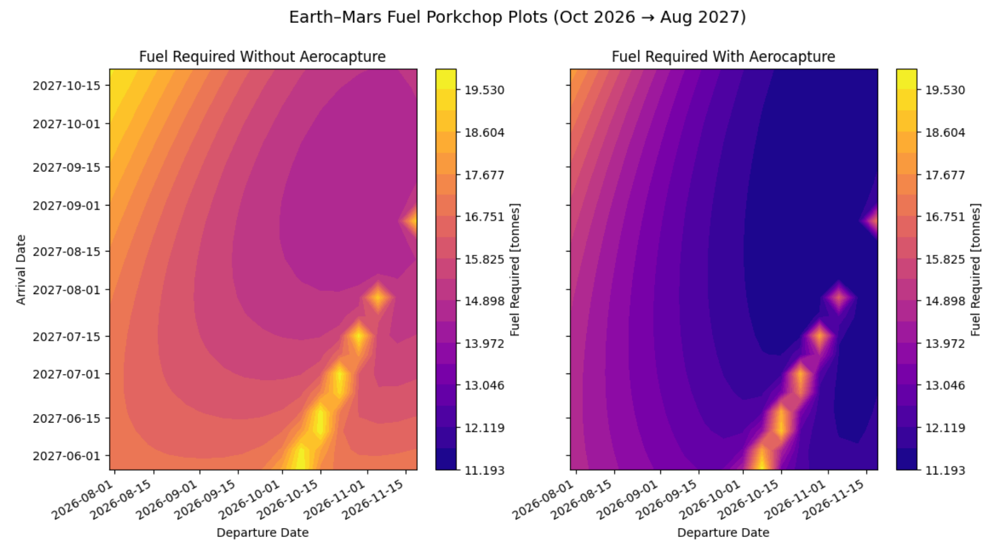


Figure 5: Fuel requirement porkchop plots, showing propellant consumption in tonnes for the same departure/arrival windows. Left: propulsive capture at Mars. Right: aerocapture at Mars.

### 2.3.1 Identification of Optimal Trajectory Regions

The $\Delta v$ plots clearly highlight regions of minimum cost transfer. For the purely propulsive scenario, the minimum $\Delta v$ is approximately 5.75 km/s, corresponding to a departure in the end of October 2026 and arrival in early September 2027. In the aerocapture case, the Mars orbit insertion burn is eliminated, reducing the total mission $\Delta v$ to approximately 3.70 km/s for the same window.

The corresponding fuel porkchops confirm this trend. Without aerocapture, the optimal trajectory consumes around 15 tonnes of propellant, leaving only $\sim$5 tonnes of dry mass available from the original 20 tonne spacecraft. With aerocapture, the propellant required drops to only $\sim$11 tonnes, meaning $\sim$9 tonnes remain for payload and structure.

### 2.3.2 Comparison and Sensitivity

The comparison between cases demonstrates that aerocapture provides a reduction of $\sim$2.05 km/s in $\Delta v$ and a corresponding saving of 5 tonnes of propellant. This represents a payload delivery improvement of more than 50% for the same launch mass.

The plots also highlight the sensitivity of trajectory design to departure epoch. Departures outside the October 2026 minimum-energy window rapidly increase both $\Delta v$ and fuel consumption; for example, shifting the departure to late November can increase propellant demand by 3–4 tonnes.

## 2.4 Trajectory Visuals

To provide geometric context to the porkchop analysis, trajectory visualisations were generated for the selected Earth–Mars transfer, departing on 27 October 2026 and arriving on 21 September 2027 (time of flight $\sim$329 days).

Figure 6 illustrates the heliocentric geometry of the transfer, showing the orbits of Earth and Mars and the Lambert arc connecting the departure and arrival positions. The corresponding Mars-centred arrival views are given in Figure 7, which compare the hyperbolic approach under a purely propulsive capture scenario against the aerocapture case. These plots highlight the fundamental difference in arrival conditions: a large retrograde burn at periapsis versus atmospheric drag with only a small trim manoeuvre. Together, these visualisations link the interplanetary phasing seen in the porkchop plots to the local capture dynamics at Mars.

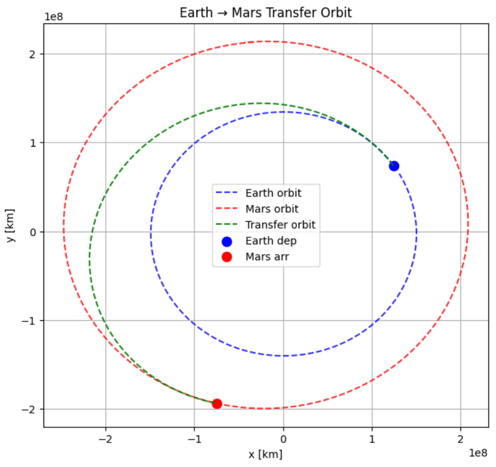


Figure 6: Heliocentric transfer orbit for departure on 27 October 2026 and arrival at Mars on 21 September 2027. Earth and Mars orbits are shown as dashed lines, with the transfer trajectory in green.

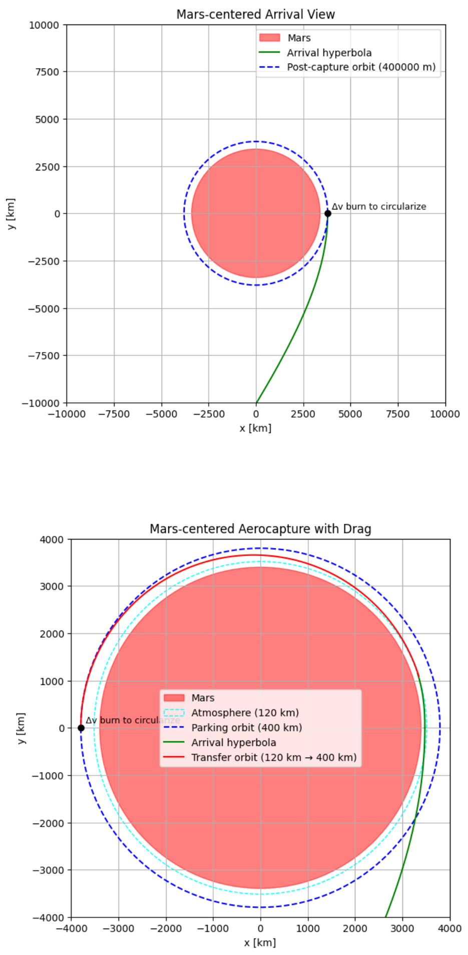


Figure 7: Mars-centred arrival geometry for 21 September 2027. Top: propulsive capture, requiring a large retrograde insertion $\Delta v$. Bottom: aerocapture case, where atmospheric drag removes most of the hyperbolic excess velocity and only a small trim burn is needed.

# 3 Determining the Fastest Feasible Transfers for a Fixed Spacecraft Mass

## 3.1 Introduction

While the previous section compared total $\Delta v$ and propellant requirements across the entire 2026–2027 launch window, it did not explicitly determine which of these transfer opportunities could be flown by a spacecraft with a fixed propellant budget. In practice, mission design is constrained by the total wet mass that can be launched and the proportion of that mass allocated to propellant. To account for these limitations, a post-processing routine was developed to identify the fastest viable Earth–Mars transfer that a 20 tonne spacecraft can perform, given realistic payload and structural mass fractions.

The purpose of this analysis is to quantify how the use of aerocapture affects both the achievable time of flight and the detailed $\Delta v$ distribution for a spacecraft with limited fuel reserves. In particular, by eliminating the 2 km/s Mars orbit insertion burn, aerocapture should permit higher-energy departures from Earth, enabling a shorter cruise duration within the same propellant budget. The results therefore demonstrate the trade-off between fuel, travel time, and trajectory energy for the two mission architectures.

## 3.2 Methodology

### 3.2.1 Spacecraft Mass Model

The spacecraft is assumed to have a total mass of $m_0 = 20\,000$ kg, consisting of a 1 tonne payload and a 2 tonne structural dry mass. This leaves a maximum available propellant mass of $m_{\text{fuel,avail}} = 17\,000$ kg. The propulsion system has a specific impulse of $I_{sp} = 450$ s, consistent with the previous analysis. Any trajectory that requires more than 17 tonnes of propellant is deemed infeasible. This constraint ensures that only missions physically achievable by the baseline vehicle are considered.

### 3.2.2 Filtering Feasible Transfers

The routine imports the data arrays generated by the `porkchop_sweep_2026()` function, which contain the total $\Delta v$, propellant mass, and time-of-flight (TOF) values for every departure–arrival combination. These arrays are then filtered to exclude trajectories that exceed the available propellant:

Listing 4: Masking infeasible trajectories based on fuel mass constraint.

```
mask_no_aero = np.ma.masked_where(FUEL_no_aero > m_fuel_available.value, TOF_days)
mask_aero = np.ma.masked_where(FUEL_aero > m_fuel_available.value, TOF_days)
```

The `numpy.ma.masked_where()` function applies a logical mask, leaving only the transfer opportunities that satisfy the spacecraft's fuel limit. This filtering is performed separately for the propulsive and aerocapture cases.

### 3.2.3 Fastest Valid Transfer Search

Within the set of feasible transfers, the algorithm searches for the shortest time-of-flight by identifying the minimum unmasked TOF value:

Listing 5: Locating the minimum time-of-flight among feasible transfers.

```
i_no, j_no = np.unravel_index(np.nanargmin(mask_no_aero), mask_no_aero.shape)
i_a, j_a = np.unravel_index(np.nanargmin(mask_aero), mask_aero.shape)
```

The corresponding indices are used to retrieve the launch and arrival dates of each optimal trajectory. These dates are then passed back into the `mars_transfer()` function to recompute the detailed $\Delta v$ components (Earth departure and Mars arrival) for both cases. This ensures that the output includes not only total $\Delta v$, but also the contribution of each manoeuvre:

Listing 6: Re-evaluating detailed $\Delta v$ components for fastest transfers.

```
res_no_aero = mars_transfer(dep_dates[i_no].iso, arr_dates[j_no].iso,
                         m0=m0, Isp=Isp, h_parking=400*u.km,
                         plot=False, mars_plot=False, porkchop=False)

res_aero = mars_transfer(dep_dates[i_a].iso, arr_dates[j_a].iso,
                       m0=m0, Isp=Isp, h_parking=400*u.km,
                       plot=False, mars_plot=False, porkchop=False)
```

### 3.2.4 Output and Comparison

The function then prints a detailed summary of both optimal transfers, including the time of flight, total propellant consumed, and the individual $\Delta v$ requirements at Earth departure and Mars arrival:

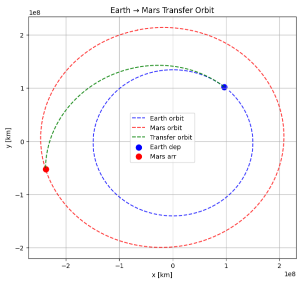
Table 3: Comparison of optimal Earth–Mars transfers for a 20 tonne spacecraft with and without aerocapture.

| **Parameter** | **Propulsive Capture** | **Aerocapture** |
|---|---|---|
| Time of Flight [days] | 196 | 189 |
| Propellant Used [kg] | 16 986 | 11 953 |
| Departure $\Delta v$ (Earth) [km/s] | 3.82 | 3.97 |
| Arrival $\Delta v$ (Mars) [km/s] | 2.31 | 0.20 |
| Total Mission $\Delta v$ [km/s] | 6.13 | 4.17 |
| Launch Date | 2026–11–12 | 2026–11–19 |
| Arrival Date | 2027–05–27 | 2027–05–27 |
| **Difference (Aerocapture − Propulsive)** | $\Delta t_{\text{flight}} = -7$ days, $\Delta v_{\text{saved}} = 1.96$ km/s, Propellant saving $\approx 5$ tonnes | |

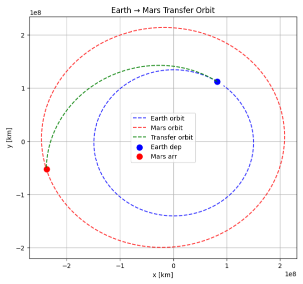
This output reveals that the aerocapture trajectory departs one week later, arrives on the same date, and consumes approximately 5 tonnes less propellant. The 2 km/s reduction in Mars arrival $\Delta v$ directly translates into a lighter fuel requirement and a modest 7-day reduction in flight time, demonstrating the efficiency advantage of aerocapture within the 2026–2027 launch opportunity.

## 3.3 Task 3: Trajectory Visualisation and Comparison

Figures 8–9 present the interplanetary trajectories and Mars-centred arrival geometries for both the aerocapture and non-aerocapture mission scenarios. The solutions correspond to optimised transfers between **Earth departure in late 2026** and **Mars arrival in May 2027**, computed using the `hapsira` astrodynamics package to solve the Lambert transfer and propagate the resulting trajectories.

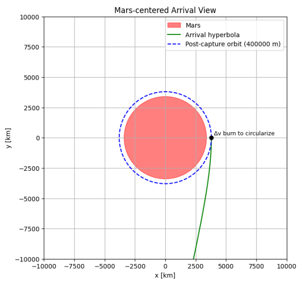


(a) No-aerocapture case: Earth departure on **12 Nov 2026**, $v_{\infty,\text{dep}} = 4.06$ km/s, Mars arrival on **27 May 2027**, $v_{\infty,\text{arr}} = 6.19$ km/s. Total $\Delta v_{\text{total,no aero}} = 6.36$ km/s.

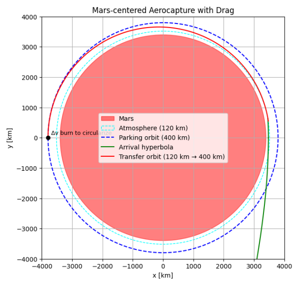


(b) Aerocapture case: Earth departure on **19 Nov 2026**, Mars arrival on **27 May 2027**. Atmospheric braking replaces most of the capture burn, leaving only $\Delta v_{\text{Mars,aero}} = 0.065$ km/s.

Figure 8: Comparison of heliocentric Earth–Mars transfer trajectories for the no-aerocapture (left) and aerocapture (right) mission profiles.



(a) No-aerocapture: Mars arrival with $v_{\infty,\text{arr}} =$ 6.19 km/s, requiring a $\Delta v_{\text{Mars,no aero}} = 4.44$ km/s capture burn at periapsis to enter a 300 km circular orbit.



(b) Aerocapture: entry at $r_p = R_M + 80$ km enables drag to remove nearly all hyperbolic excess energy. A small $\sim 64$ m/s trim burn circularises the orbit at 300 km altitude.

Figure 9: Mars-centred hyperbolic arrival trajectories for the no-aerocapture (left) and aerocapture (right) scenarios.

## Orbital parameter and mission comparison

Table 4 summarises the transfer orbital elements and key mission results for the aerocapture and no-aerocapture cases (values computed using `hapsira`).

Table 4: Comparison of transfer orbital elements and Mars transfer results

| | **Aerocapture** | **No Aerocapture** |
|---|---|---|
| *Orbital elements (transfer orbit)* | | |
| Semi-major axis, $a$ [km] | $2.0608 \times 10^8$ | $2.0595 \times 10^8$ |
| Eccentricity, $e$ | 0.2865 | 0.2842 |
| Inclination, $i$ [°] | 24.3746 | 24.7088 |
| RAAN, $\Omega$ [°] | 3.2493 | 3.3917 |
| Argument of periapsis, $\omega$ [°] | 48.0364 | 47.5026 |
| True anomaly, $\nu$ [°] | 5.1670 | $-1.5075$ |
| *Mars transfer results* | | |
| Time of flight (days) | 189.0000 | 196.0000 |
| $v_{\infty,\text{dep}}$ [km/s] | 4.1802 | 4.0585 |
| $v_{\infty,\text{arr}}$ [km/s] | 6.2726 | 6.1904 |
| $\Delta v_{\text{Earth}}$ [km/s] | 3.9541 | 3.9110 |
| $\Delta v_{\text{Mars}}$ [km/s] | 0.0747 | 4.4440 |
| Total $\Delta v$ [km/s] | 4.0289 | 8.3550 |
| Propellant mass [kg] | 11 970.7910 | 16 986.4761 |
| *Differences / savings (aero vs no-aero)* | | |
| Total $\Delta v$ reduction [km/s] | 4.3261 | (51.8% reduction) |
| Propellant mass saved [kg] | 5 015.6851 | (29.5% saved) |
| Launch date difference | 2026-11-19 vs 2026-11-12 | (7 days later for aero) |

**Interpretation.** Table 4 shows that both transfers have nearly identical heliocentric geometries, with minimal differences in semi-major axis, eccentricity, and orbital orientation. This confirms that the interplanetary leg is largely independent of the Mars capture method. The mission-level contrast, however, is significant: aerocapture reduces total $\Delta v$ from **8.3550 km/s** to **4.0289 km/s** (a **4.3261 km/s** or **51.8%** reduction), and lowers propellant usage from **16,986.4761 kg** to **11,970.7910 kg** (**29.5%** saving).

The smaller percentage change in propellant mass relative to $\Delta v$ reflects the exponential dependence of the rocket equation and the fixed vehicle structural mass fraction. Additionally, the aerocapture case allows a **seven-day later launch window** (19 Nov 2026 vs 12 Nov 2026) for the same arrival date, improving operational flexibility and reducing schedule risk while maintaining an identical transfer geometry.

## Discussion

The results in Table 4 highlight the significant performance advantage of using atmospheric braking for Mars capture. Although both transfers share nearly identical orbital geometries, replacing the propulsive insertion with aerocapture more than halves the total $\Delta v$, reducing propellant demand and launch mass while increasing delivered payload capability.

Operationally, the aerocapture case also allows a seven-day later launch for the same arrival date (19 Nov 2026 vs 12 Nov 2026), providing improved schedule flexibility and reduced sensitivity to phasing constraints. Overall, the inclusion of aerocapture introduces manageable complexity in exchange for substantial efficiency and mass savings—making it an attractive option for future high-mass or crewed Mars missions.

## 4 Conclusion

This study developed a complete numerical framework for analysing Earth–Mars transfer trajectories and evaluating the performance benefits of atmospheric aerocapture. Using the `hapsira` astrodynamics package, the model successfully reproduced analytical benchmark results from Curtis (Example 8.8) and extended the analysis to the 2026–2027 launch opportunity.

Across all tasks, results demonstrated the substantial efficiency gains achievable through aerocapture. By using Mars' atmosphere to dissipate hyperbolic excess energy, the required arrival $\Delta v$ decreased by more than 4 km/s, halving the total mission $\Delta v$ and saving approximately 5 tonnes of propellant for a 20 tonne spacecraft. These savings directly translate into increased payload capacity and launch flexibility without altering the interplanetary transfer geometry.

The porkchop analyses and trajectory visualisations confirmed that both the propulsive and aerocapture transfers share nearly identical heliocentric orbits, but differ significantly in capture energy and mass efficiency. Aerocapture also allowed a later Earth departure for the same Mars arrival date, enhancing operational flexibility and reducing schedule sensitivity.

While the analysis was idealised—neglecting thermal protection, aerodynamic lift, and entry guidance effects—the findings clearly illustrate the mission-level advantages of incorporating atmospheric braking into Mars capture design. Future extensions of this work could include detailed aerodynamic modelling, thermal load prediction, and probabilistic guidance simulations to validate practical feasibility.

In summary, aerocapture represents a transformative technique for high-mass or crewed Mars missions, enabling lower launch mass, higher delivered payload, and greater mission robustness within the same transfer opportunity.

## References

[1] C. M. James. Aero4800 assignment and task sheet: Astrodynamics and interplanetary mission design. Lecture and assessment material, Centre for Hypersonics, The University of Queensland, 2025. Course material distributed to AERO4800 students, September 2025.

[2] Howard D. Curtis. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, Oxford, United Kingdom, 4th edition, 2020. ISBN 978-0-08-102133-0. A comprehensive reference on astrodynamics and orbital transfer theory.

[3] Dae-Hyun Kim, Min-Soo Lee, and Ji-Won Park. Transfer orbits and launch window analysis for earth-to-mars. *Journal of Astronomy and Space Sciences*, 42(2):97–111, 2025. doi: 10.5140/JASS.2025.42.2.97. URL `https://www.koreascience.or.kr/article/JAKO202519750403407.page`. Available from KoreaScience. Identifies the October 2026 Earth–Mars launch opportunity as a low-energy transfer window.