POLITECNICO
MILANO 1863

Orbital Mechanics
Module 3: Orbital manoeuvres

Academic year 2020/21

Juan Luis GONZALO GOMEZ, Giacomo BORELLI, Camilla COLOMBO
Department of Aerospace Science and Technology

Module Contents

Orbital manoeuvres

- **Lambert's Problem**
  - Motivation, geometrical properties
  - Lambert's theorem
  - Solving Lambert's problem
  - **Exercise 1: State reconstruction problem**
  - **Exercise 2: Orbit transfer problem**

- **Transfer design**
  - Parametric optimization problem
  - Time-free transfer between two orbits
  - Porkchop plots
  - **Exercise 3: Mars Express**
  - **Exercise 4: Mission Express**

Auxiliar functions available in Beep

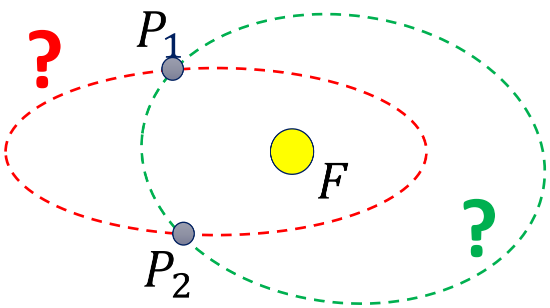
- A set of auxiliar MATLAB functions is now **available in Beep**:
  - `lambertMR`: Lambert solver
  - `uplanet`: Analytical ephemeris of planets of the Solar System
  - `ephNEO`: Analytical ephemerides of several asteroids/small bodies.
  - `ephMoon`: Analytical ephemeris of the Moon
  - `astroConstants`: Function with astrodynamics-related physical constants (e.g. gravitational parameter of the Sun and planets)
  - **timeConversion.zip**: Compressed folder with several time conversion routines
- You can use these functions for the labs and the assignments.

LAMBERT'S PROBLEM

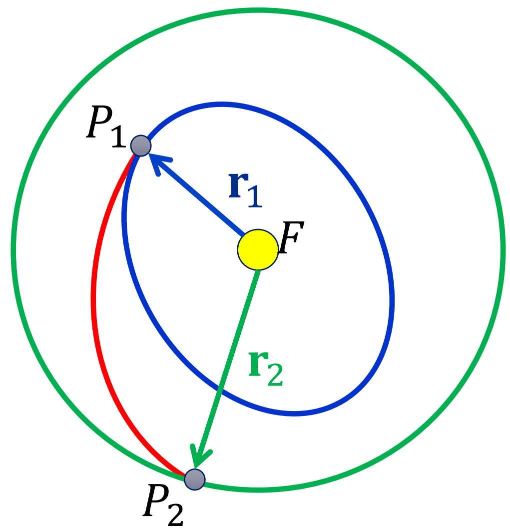
Motivation

Two-body orbital boundary value problem

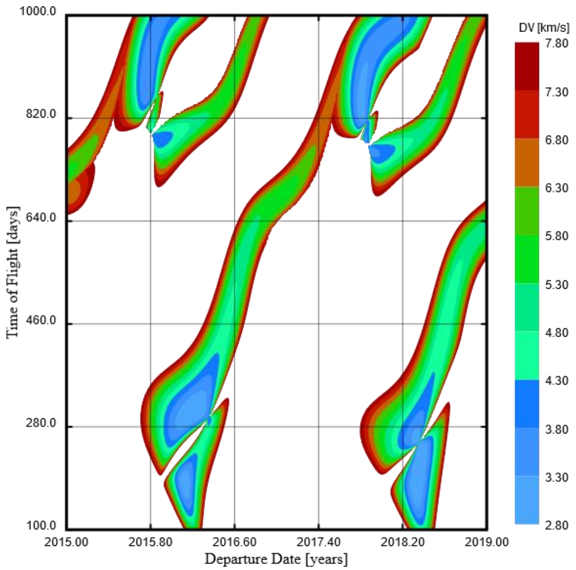
There are many practical situations where we are interested in constructing a **two-body problem orbit** (i.e. a conic) **that passes through given points**.



Orbit reconstruction



Orbital transfer



Mission design

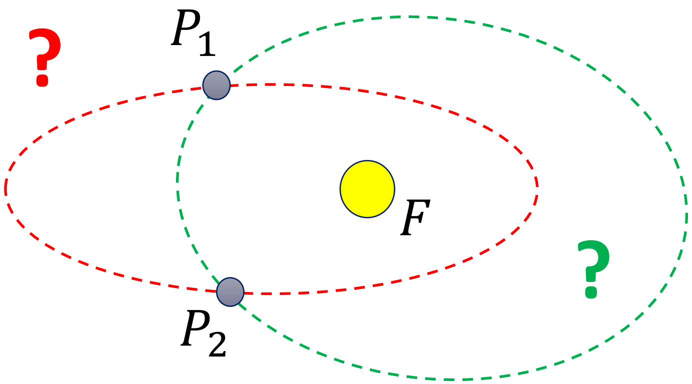
This is a **boundary value problem** (ODE system with values of the state **partially specified at more than one point**), compared to the **initial value problems** (state **completely defined at one point**) we have considered so far.

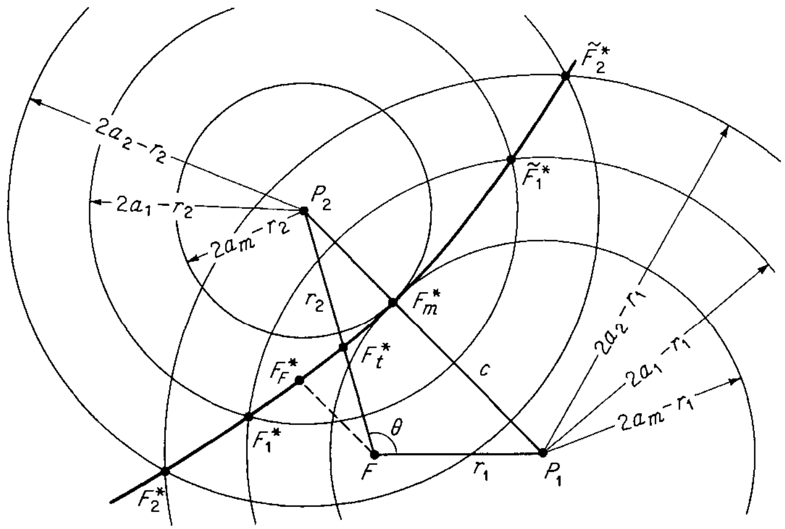
Geometrical Properties

Locus of the vacant focus

Two points P_1 and P_2 (and the focus F) are **not enough to univocally define an ellipse**.

(remember, the attracting body is always a focus in a two-body problem)





Locus of the vacant foci. Illustration taken from [1]

The vacant focus lies on a hyperbola with foci P_1 and P_2

Therefore, we have a **family of solutions with infinite possible orbits**.

[1] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, 1999

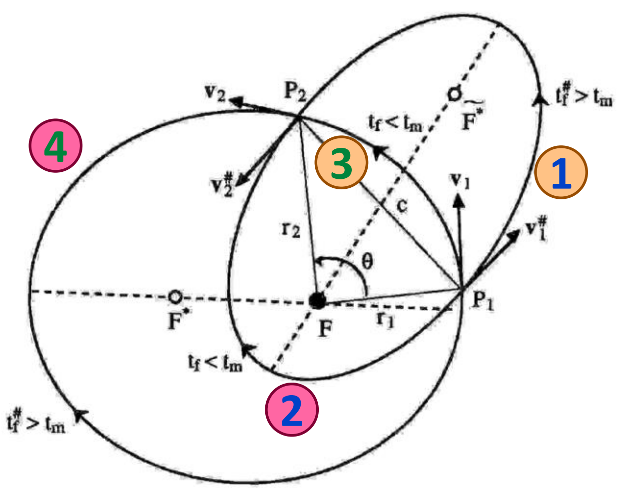
Geometrical Properties

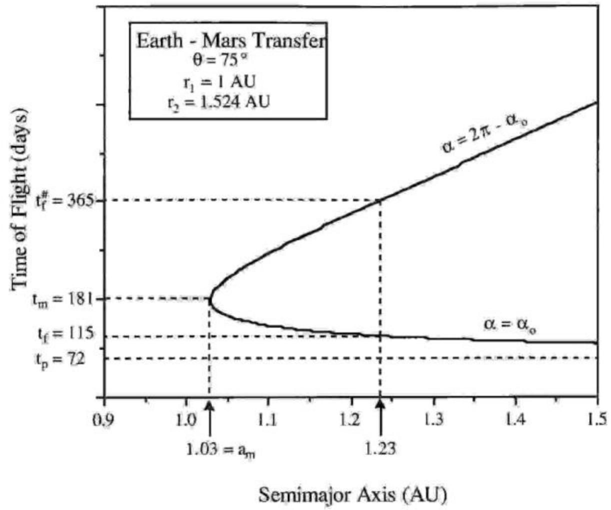
Semi-major axis

Setting the semi-major axis of the ellipse (i.e. the period of the orbit), reduces this family of solutions to two possible ellipses with different eccentricities.

- Four possible connection arcs
  - Each ellipse has **1 prograde** and **1 retrograde** arc
- Two possible transfer times
  - Each ellipse has 1 arc with transfer time $\boldsymbol{\Delta t_1}$ and 1 arc with $\boldsymbol{\Delta t_2}$

For each **transfer time** there is **1 prograde** and **1 retrograde** arc (belonging to different ellipses)





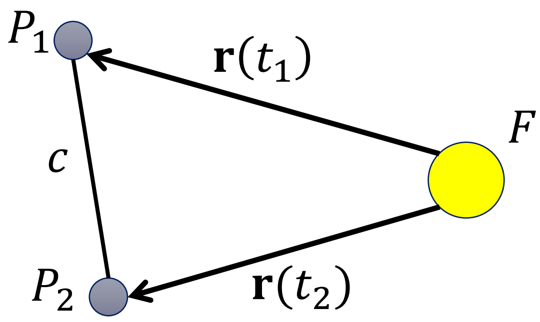
Not all values of semi-major axis lead to a solution: a minimum a is required.

Lambert’s Theorem

Definition

Lambert’s Theorem: *The orbital transfer time depends only upon the semi-major axis, the sum of the distances of the initial and final points of the arc from the center of force, and the length of the cord joining these points* [1]

$$\sqrt{\mu}(t_2 - t_1) = F(a, r_1 + r_2, c)$$



Note that $r_1 + r_2$ and c can be computed from position vectors $\mathbf{r}(t_1)$ and $\mathbf{r}(t_2)$

Knowing P_1 and P_2 and the time of flight $\Delta t = t_2 - t_1$, the semi-major axis can be obtained.

- Implies solving an **implicit equation**

[1] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, 1999

Lambert's Theorem

Proof

First analytic proof of Lambert's theorem was provided by Lagrange (see [1])

Kepler's equation
(time equation in the initial value problem)

$$\sqrt{\frac{\mu}{a^3}}(t_2 - t_1) = E_2 - E_1 - e(\sin E_2 - \sin E_1)$$

Geometrical manipulations

Lambert's equation
(transfer-time equation in the boundary value problem)

$$\sqrt{\frac{\mu}{a^3}}(t_2 - t_1) = (\alpha - \sin\alpha) - (\beta - \sin\beta)$$

$$\alpha, \beta = f(c, a, r_1 + r_2)$$

[1] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, 1999

Lambert’s Problem

Lambert’s problem: *Definition of an orbit, having a specified transfer time and connecting two position vectors.*

Many algorithms have been developed to tackle this problem:

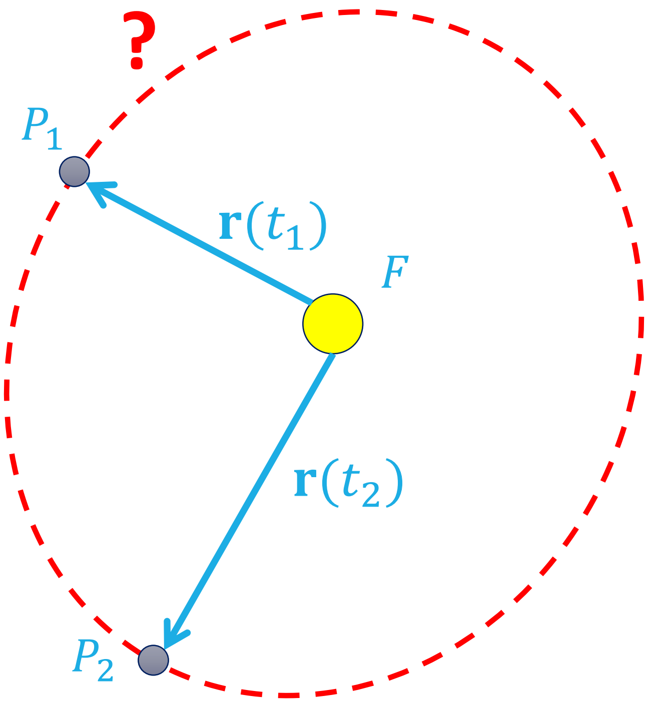
- First one by C.F. Gauss (in his book *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*)
- More recent results:
  - Battin, R. H., and Vaughan, R. M., “An elegant Lambert algorithm,” *Journal of Guidance, Control, and Dynamics*, *7*(6):662-670, 1984.
  - Avanzini, G., “A simple Lambert algorithm,” *Journal of Guidance, Control, and Dynamics*, *31*(6):1587-1594, 2008.
  - Arora, N., and Russell, R., “A Fast and Robust Multiple Revolution Lambert Algorithm Using a Cosine Transformation,” *Astrodynamics 2013, Advances in the Astronautical Sciences*, 150, AAS Paper 13-728, 2013.
  - Gooding, R., “A Procedure for the Solution of Lambert’s Orbital Boundary-Value Problem,” *Celestial Mechanics and Dynamical Astronomy*, 48(2):145–165, 1990.
  - Izzo, D., “Revisiting Lambert’s Problem,” *Celestial Mechanics and Dynamical Astronomy*, 121(1):1–15, 2015.
  - Bombardelli, C., Gonzalo, J. L., and Roa, J., “Approximate analytical solution of the multiple revolution Lambert’s targeting problem,” *Journal of Guidance, Control, and Dynamics*, 41(3):792-801, 2018. Online app available.
- Typical output are the velocities at the initial and final points, $\mathbf{v}(t_1)$ and $\mathbf{v}(t_2)$
- **For the lab, you are provided a Lambert solver (download it from Beep)**

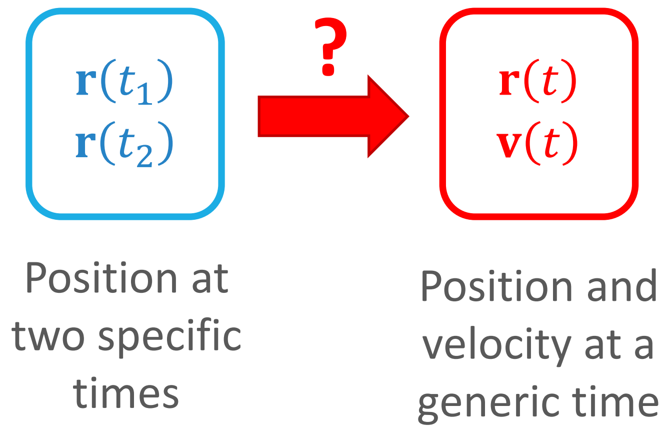
Exercise 1: State reconstruction problem

Where will we be?

We have two position vectors evaluated at different times t_1 and t_2 (for instance, coming from telescope or radar observations)

Can we reconstruct the orbit?



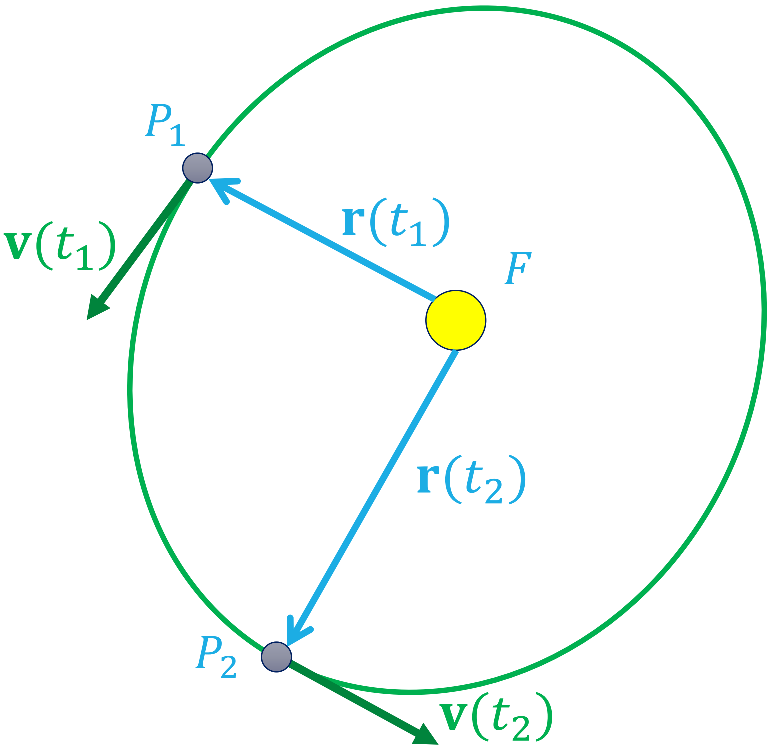


Exercise 1: State reconstruction problem

Where will we be?

We have two position vectors evaluated at different times t_1 and t_2 (for instance, coming from telescope or radar observations)

We can reconstruct the orbit



In many Lambert solvers, outputs are the velocity vectors at P_1 and P_2 (either one can be used to define the orbit)

Exercise 1: State reconstruction problem

Workflow

INPUTS
$\mathbf{r}(t_1)$
$\mathbf{r}(t_2)$
Δt
prograde/retrograde

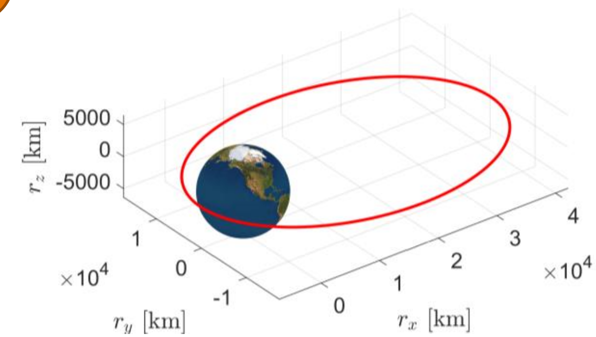
Lambert Solver

OUTPUTS
$\mathbf{v}(t_1)$
$\mathbf{v}(t_2)$

Initial conditions
$\mathbf{r}(t_0), \mathbf{v}(t_0)$
[either $\mathbf{r}(t_1), \mathbf{v}(t_1)$ or $\mathbf{r}(t_2), \mathbf{v}(t_2)$]

Orbit propagation

Orbit propagation: prediction of the orbital characteristics of a body at some future date given the current orbital characteristics.



Exercise 1: State reconstruction problem

Exercise 1: State reconstruction problem

1. Write a script to solve Lambert's problem for the values of $\mathbf{r}(t_1)$, $\mathbf{r}(t_2)$ and $\Delta t = t_2 - t_1$ given below
   - Use the provided Lambert solver `lambertMR.m`. Check sample script `call_lambertMR.m` to learn how to use it
2. Propagate and plot the resulting orbit
   - Reuse the orbit propagation functions from **Module 1**
   - Use as initial conditions either $\mathbf{r}(t_1), \mathbf{v}(t_1)$ or $\mathbf{r}(t_2), \mathbf{v}(t_2)$

Data

Prograde orbit

$\mathbf{r}(t_1) = [-21800\,;\ \ 37900\,;\ \ 0\ ]\ \mathrm{km}$

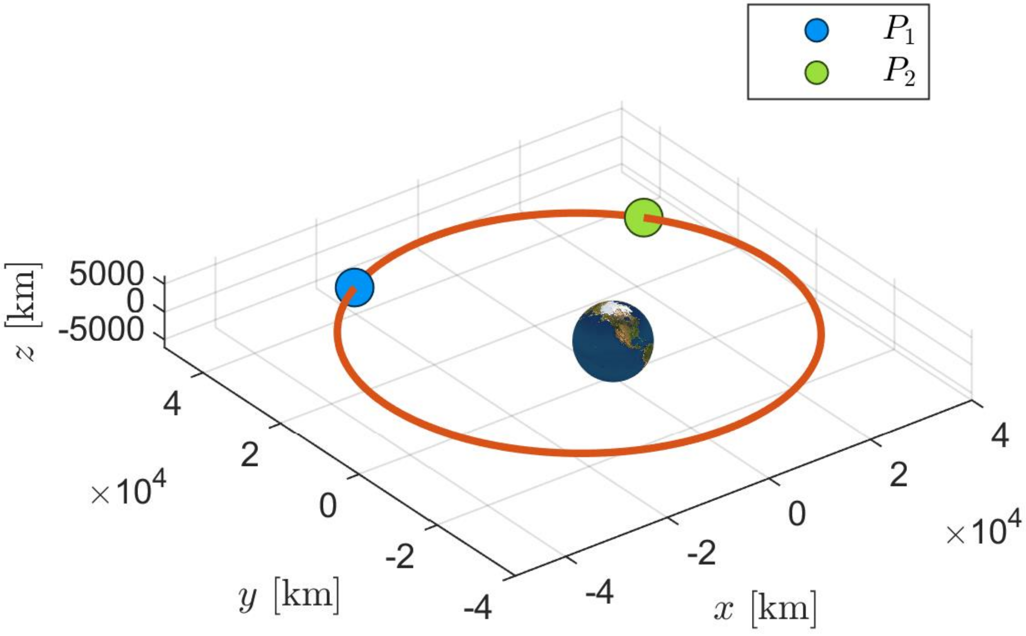
$\mathbf{r}(t_2) = [\,27300\,;\ \ 27700\,;\ \ 0\ ]\ \mathrm{km}$

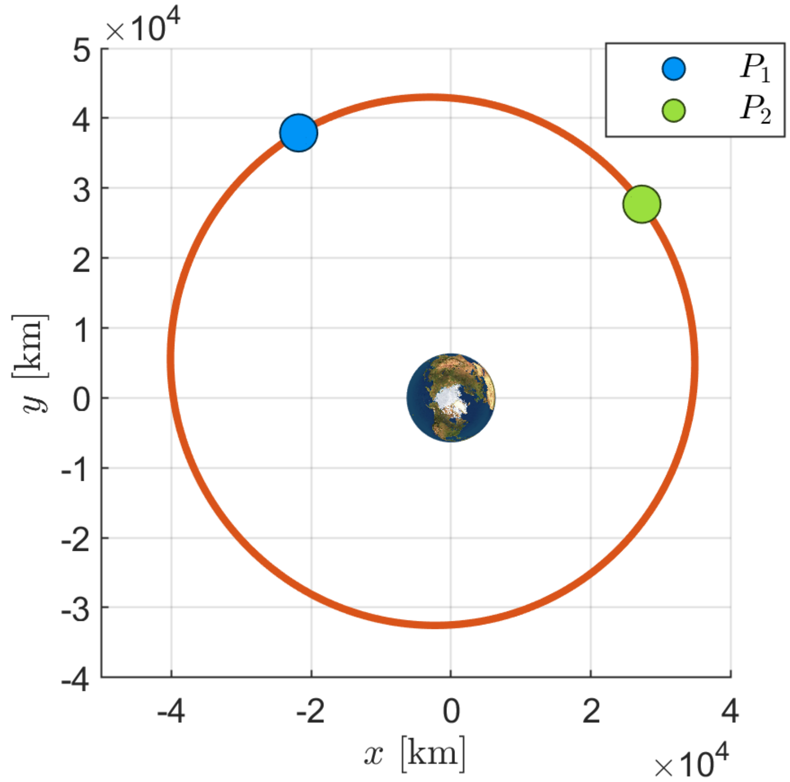
$\Delta t = 15\ \mathrm{h}, 6\ \mathrm{min}, 40\ \mathrm{s}$

$\mu_\oplus$ from `astroConstants.m`

Exercise 1: State reconstruction problem

Sample solution





Solution for Lambert's problem:

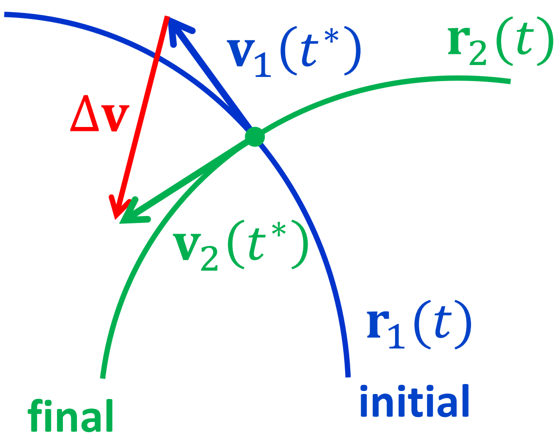
$a = 3787.1212$ km

$v(t_1) = [\,-2.3925, -1.4086, 0]$ km/s

$v(t_2) = [\,-1.8849, 2.5338, 0\,]$ km/s

Exercise 2: Orbit transfer problem

Transfer problem with intersection



For intersecting orbits, orbit transfer can be performed through a single manoeuvre at intersection time t^*

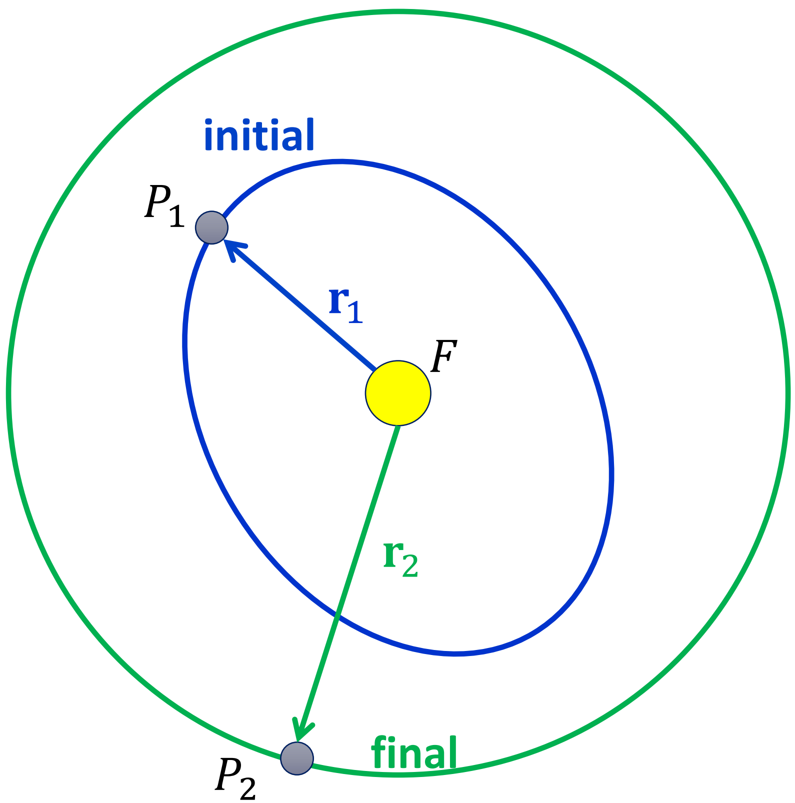
At intersection time t^*: $\mathbf{r}_1(t^*) = \mathbf{r}_2(t^*)$ ☑ $\mathbf{v}_1(t^*) \neq \mathbf{v}_2(t^*)$ ☒

We want to move from **initial** to **final** orbit (that is, we have to **change our state**) → $\mathbf{v}_1(t^*) + \Delta\mathbf{v} = \mathbf{v}_2(t^*)$

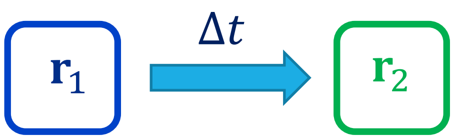
Cost of the manoeuvre: $||\Delta\mathbf{v}||$

Exercise 2: Orbit transfer problem

Transfer problem with no intersections

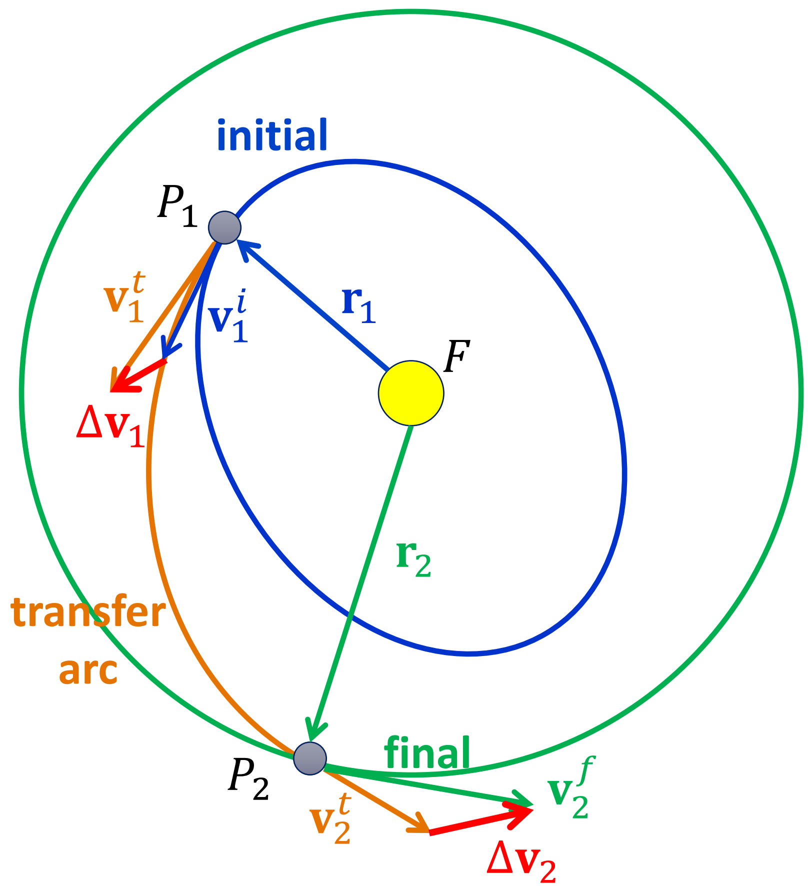


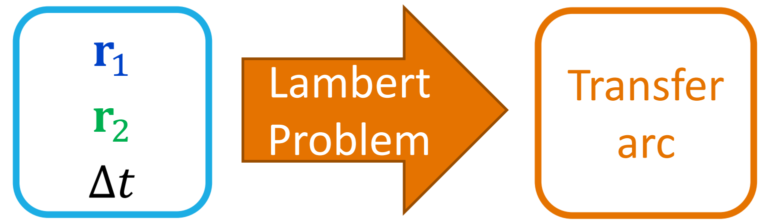
At least two manoeuvers are required



Exercise 2: Orbit transfer problem

Transfer problem with no intersections





Injection manoeuvre (from initial orbit to transfer arc):

$$\Delta\mathbf{v}_1 = \mathbf{v}_1^t - \mathbf{v}_1^i$$

Arrival manoeuvre (from transfer arc to final orbit):

$$\Delta\mathbf{v}_2 = \mathbf{v}_2^f - \mathbf{v}_2^t$$

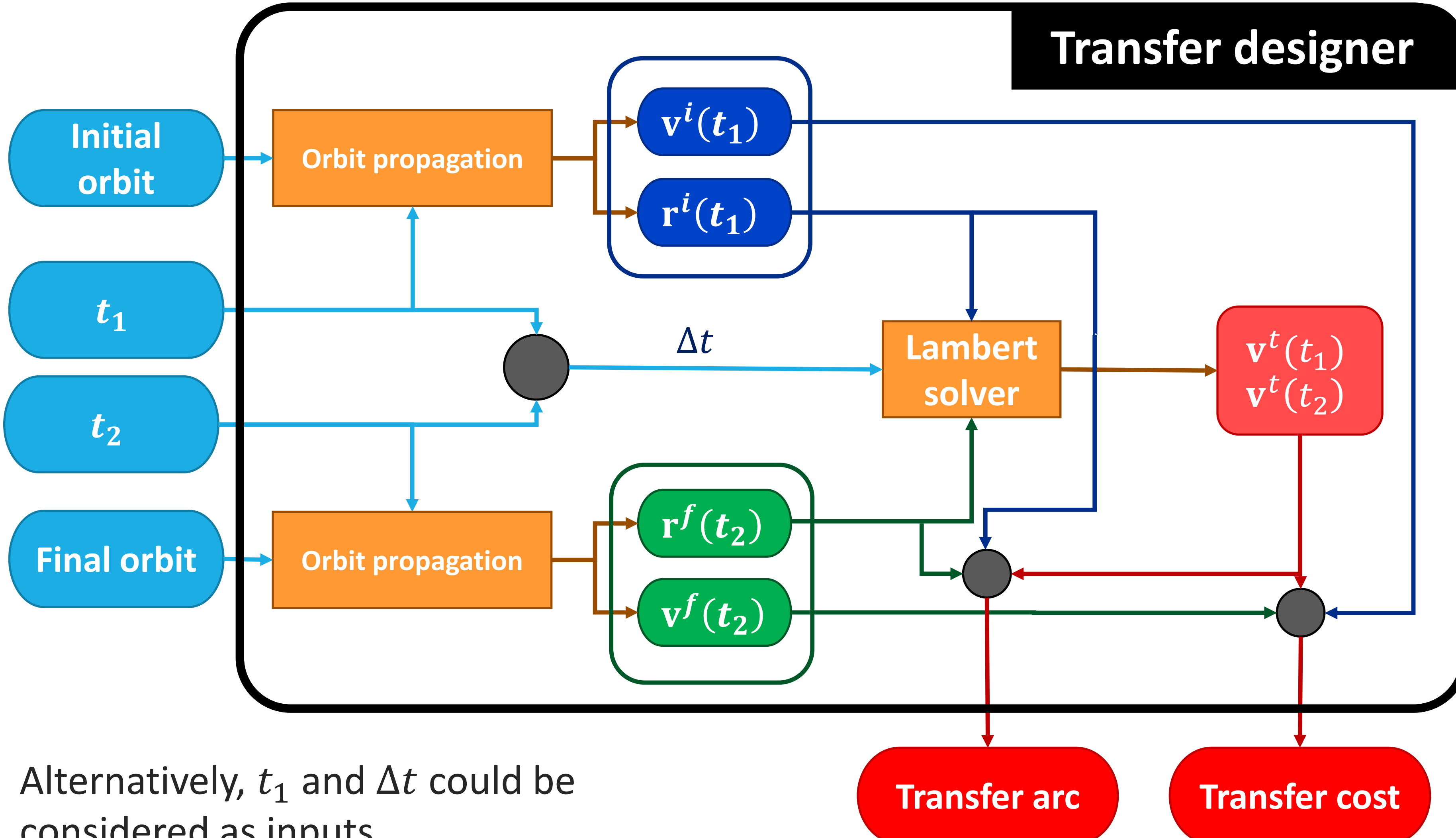
The problem has 0 degrees of freedom for given P_1, P_2, and Δt

Total cost of the mission:

$$\Delta v_{tot} = \|\Delta\mathbf{v}_1\| + \|\Delta\mathbf{v}_2\|$$

Exercise 2: Orbit transfer problem

Workflow for a fixed-time transfer



Alternatively, t_1 and Δt could be considered as inputs

Exercise 2: Orbit transfer problem

Exercise 2: Orbit transfer problem

1. Compute the initial and final states in Cartesian coordinates.
2. Solve Lambert's problem for the transfer arc $v_1 - v_{1L}$ $v_{2L} - v_2$
3. Compute the total cost of the manoeuvre $||\Delta\mathbf{v}_1|| + ||\Delta\mathbf{v}_2||$
4. Propagate the transfer arc, from t_1 to t_2
5. Plot the initial and final orbits, and the transfer arc

Data

Earth-bound orbits, $\mu_\oplus$ from `astroConstants.m`

Prograde transfer arc

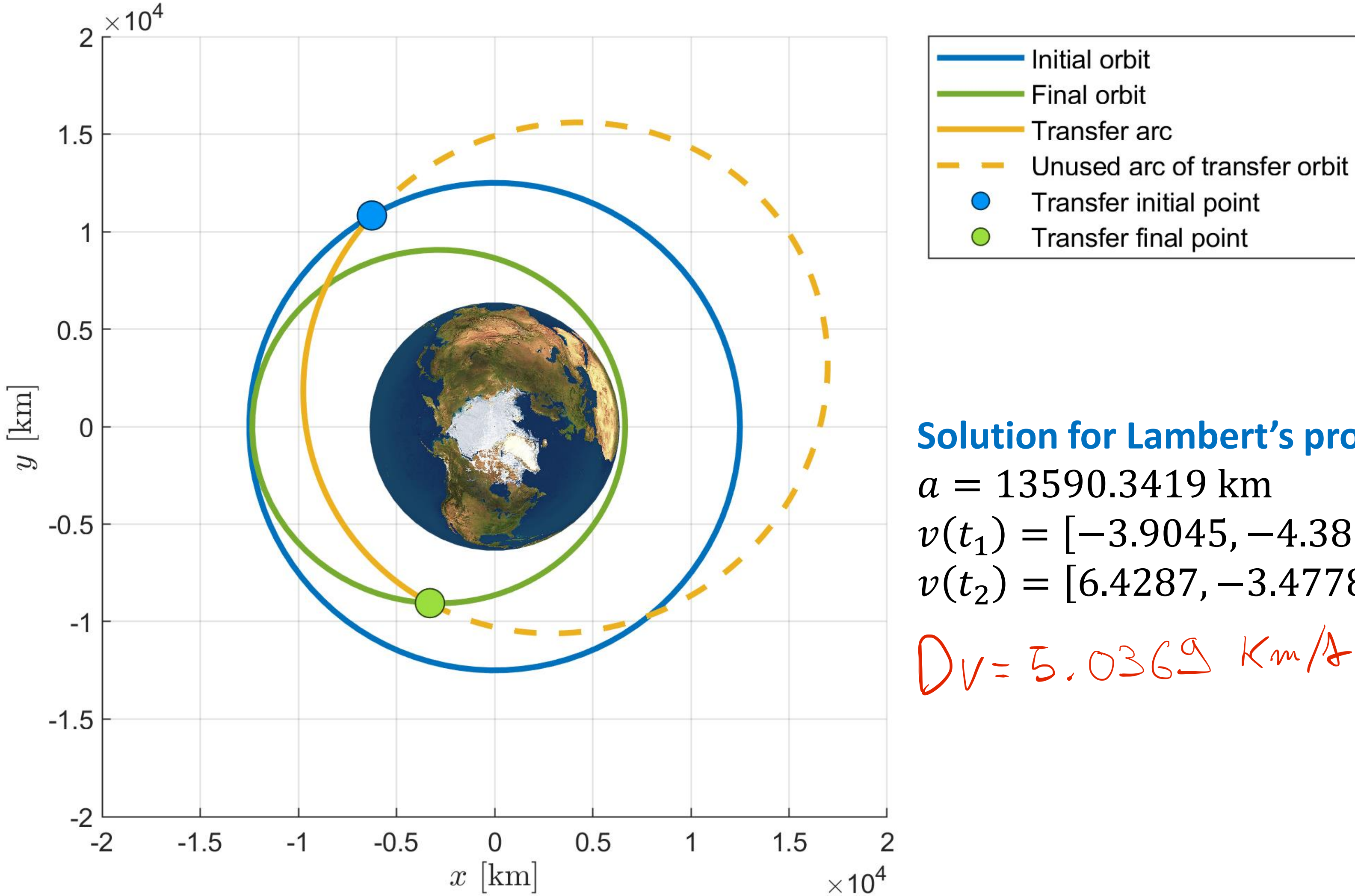
$kep_1 = [a_1; e_1; i_1; \Omega_1; \omega_1; f_1] = [\ 12500 \text{ km};\ 0\ ; 0 \text{ deg}; 0 \text{ deg}; 0 \text{ deg};\ 120 \text{ deg}\ ]$

$kep_2 = [a_2; e_2; i_2; \Omega_2; \omega_2; f_2] = [\ 9500 \text{ km};\ 0.3\ ; 0 \text{ deg}; 0 \text{ deg}; 0 \text{ deg}; 250 \text{ deg}]$

$tof = \Delta t = 3300 \text{ s}$

Exercise 2: Orbit transfer problem

Sample solution



Solution for Lambert's problem:

$a = 13590.3419$ km

$v(t_1) = [-3.9045, -4.3819, 0]$ km/s

$v(t_2) = [6.4287, -3.4778, 0\ ]$ km/s

$Dv = 5.0369$ Km/s

MJD2000 - number of DAYS from 00:00 1/1/2000

TRANSFER DESIGN

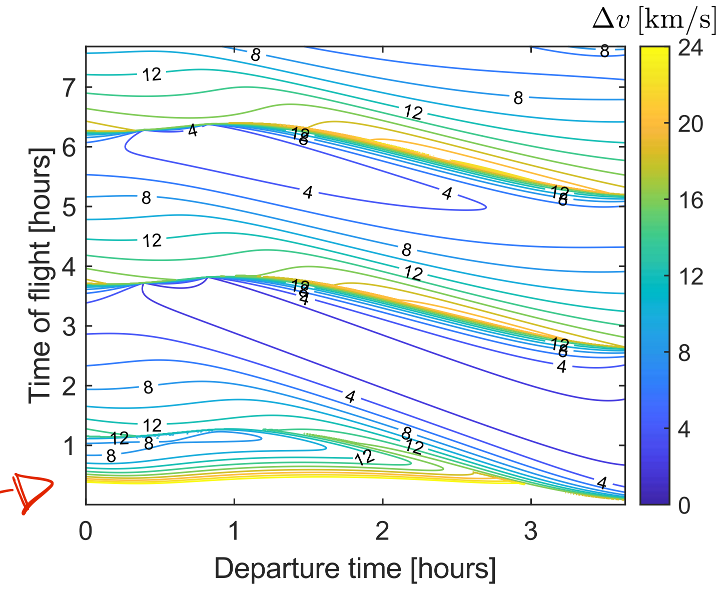
Transfer design

A parametric optimization problem

In **Exercise 2**, initial time t_1 and final time t_2 were fixed, leading to a **single possible transfer arc**.

What happens if instead we want to design a transfer between two celestial bodies, without *a priori* values for departure and arrival time?

- **Departure and arrival time are free parameters**, leading to a **family of possible transfer arcs**, each one with different Δv
- State (position and velocity) at the initial and final orbits is a known function of time. Therefore, we have just **2 degrees of freedom**
- $\Delta v(t_1, t_2)$ can be plotted as a contour plot known as **porkchop plot**
- This is a **powerful tool for mission design**



Transfer design

Choice of design variables and ranges

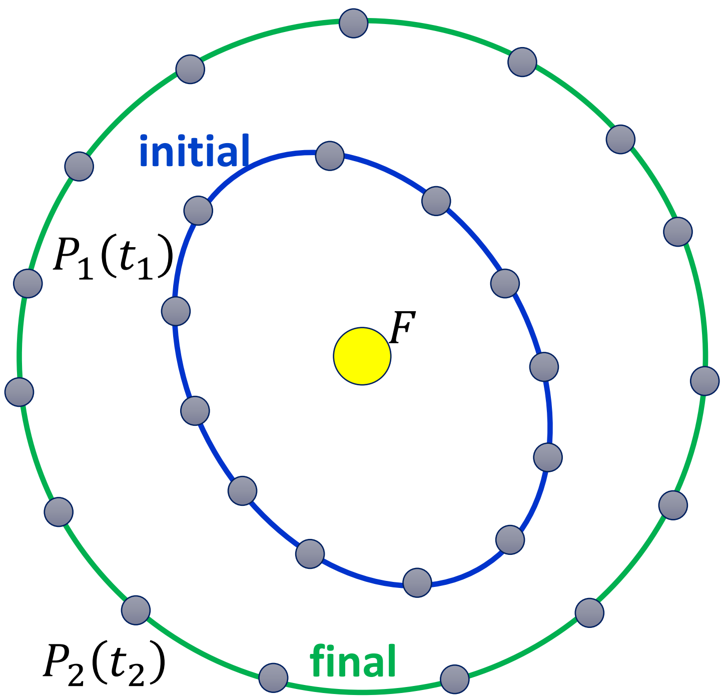
- The 2 degrees of freedom can be parametrized in different ways. The simplest ones are:
  - **Departure time** t_1 and **arrival time** t_2
  - **Departure time** t_1 and **time of flight** $\Delta t = t_2 - t_1$
    - Keep in mind that departure point P_1 changes only with the departure time t_1, whereas arrival point P_2 changes with both departure time and time of flight because $\mathbf{r}(t_2) = \mathbf{r}(t_1 + \Delta\mathrm{t})$

- In order to locate the minima, it is important to choose **time windows large enough to capture all possible configurations**:
  - For departure window, try to include all relative positions between both bodies. The synodic period is a useful first estimation
  - For time of flight, you can make initial estimations from simplified transfers (e.g. assume coplanar, circular orbits and compute the Hohmann transfer)
  - In many cases, operational constraints may limit the feasible size for the time windows (for instance, due to the lifetime of the spacecraft systems)

Time-free transfer between two orbits

A 2 degrees of freedom problem in time

We want to transfer from a **body in the initial orbit** to another **body in the final orbit**



$\mathbf{r}_1(\boldsymbol{t_1})$
$\mathbf{r}_2(\boldsymbol{t_2})$
$\Delta t = t_2 - t_1$

Lambert Problem

$\Delta v_{tot}(t_1, t_2)$

Departure and **arrival** points are **functions of the departure and arrival times** (**within the respective windows**).

The problem has 2 degrees of freedom for given departure and arrival bodies

Not all the transfer arcs will fulfill the launcher constraint $\|\Delta \mathbf{v_1}\| \leq \boldsymbol{v}_\infty$

Workflow

Fixed-time transfer (single solution)

r_I, v_I

Initial orbit

t_1

t_2

Final orbit

r_F, v_F

Transfer Designer

$TOF = t_2 - t_1$

Transfer arc

Transfer cost

Alternatively, t_1 and Δt could be considered as inputs

Workflow

Time-free transfer (parametric solution)

Initial orbit

Departure window

Arrival window

Final orbit

The **fixed-time transfer** is solved for each feasible combination of t_1 and t_2 (or t_1 and Δt) within the corresponding time windows.

Transfer Designer

Reuse code!!!

$\Delta \mathbf{v}_{tot}(t_1, t_2)$

Output is the mission cost for each (t_1, t_2). Other outputs could also be needed to check mission constraints.

Porkchop plots

Making contour plots

- The required Δv as a function of **departure** and **arrival time** (or **departure time** and **ToF**) can be represented on a **contour plot known as porkchop plot**.
  - Design tool for the analysis of possible launch opportunities.
  - Named for its resemblance to a pork chop for some missions (e.g. Earth to Mars).
- **Contour plots** can be plotted in Matlab with the `contour` function.
  - Check the **documentation center** to learn how to use `contour`.
  - Remember to add a `colorbar` with ticks and labels.
- $\Delta v(t_1, t_2)$ can also be plotted as a 3D surface using the `surf` function (but keep in mind that this is not a porkchop plot).
- Use enough discretization points for the time windows to get smooth plots.

Ephemeris

Locating objects in space

- *A table of the coordinates of celestial bodies as a function of time is called an **ephemeris*** [1].
  - Refer to **Module 2** for more details

- **Instead of propagating the orbits of the departure and arrival bodies, we will use the analytical ephemerides available in Beep**:
  - `uplanet`: Analytical ephemerides of planets of the Solar System

- **Be careful with the units!**
  - The ephemeris functions take as input the date in MJD2000 (i.e. days). Lambert solver takes as input the time of flight in seconds.

[1] Curtis, H. D.. *Orbital mechanics for engineering students,* Butterworth-Heinemann , 2014

Exercise 3: Mars Express

use mjd2000 2 date.m
date 2 mjd 2000.m

Mission definition

Mars Express: **Design an interplanetary transfer with minimum Δv_{tot} between Earth and Mars, under the following mission requirements:**

[datenum function (with from mjd 2000 to MATLAB time)]

- Departure planet: **Earth**
- Target planet: **Mars**

You might HAVE to transpose MATRIX

- Earliest departure requirement: **2003 April 1**
- Latest departure requirement: **2003 August 1**

X: ROWS
Y: COLUMNS
for plot:

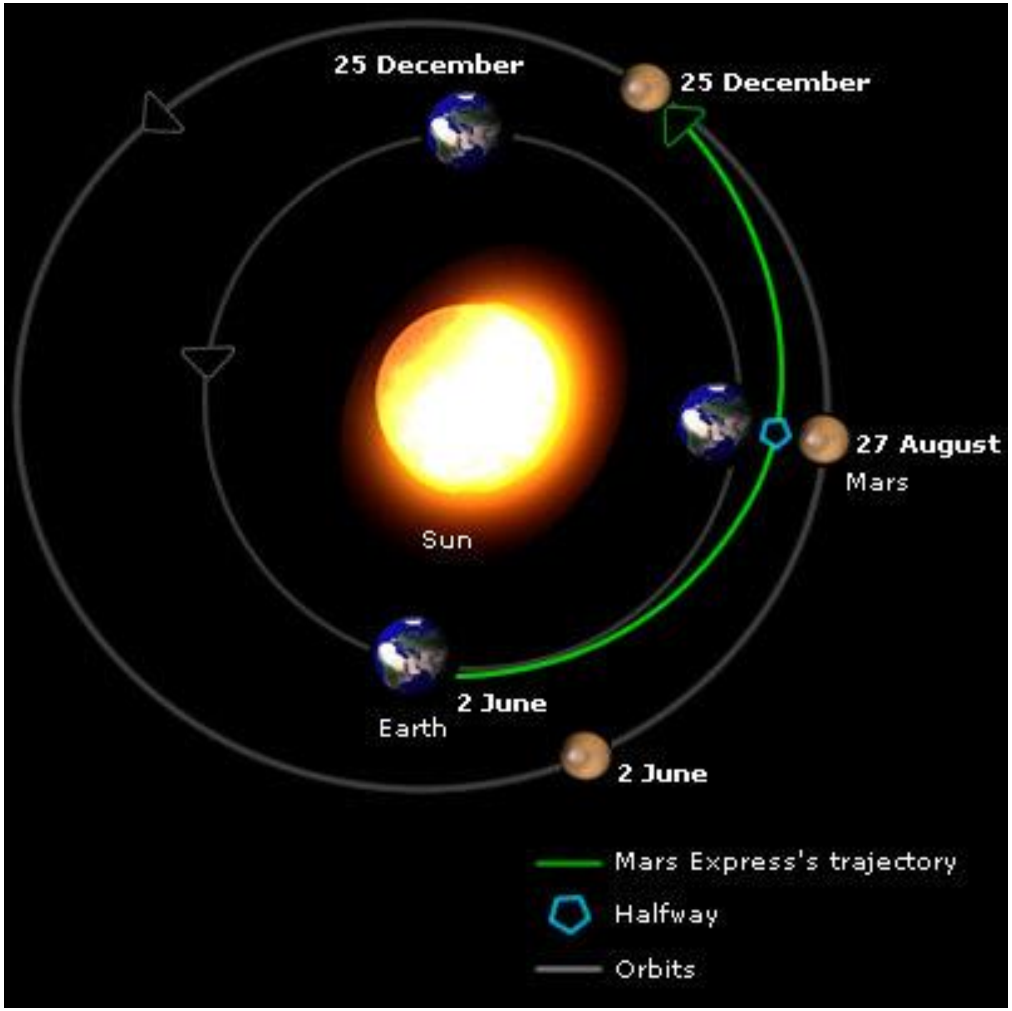
- Earliest arrival requirement: **2003 September 1**
- Latest arrival requirement: **2004 March 1**

→ datetick(...)
→ xtickangle(45)

hcb= colorbar;

Exercise 3: Mars Express

This is an actual mission!



Results should be very close to **ESA's Mars Express Mission**

- **Departure date:** **2 June 2003**
- **Arrival date:** **25 December 2003**
- $\Delta v_{\text{tot}} = 5.67\ - 5.7\ \text{km/s}$

Exercise 3: Mars Express

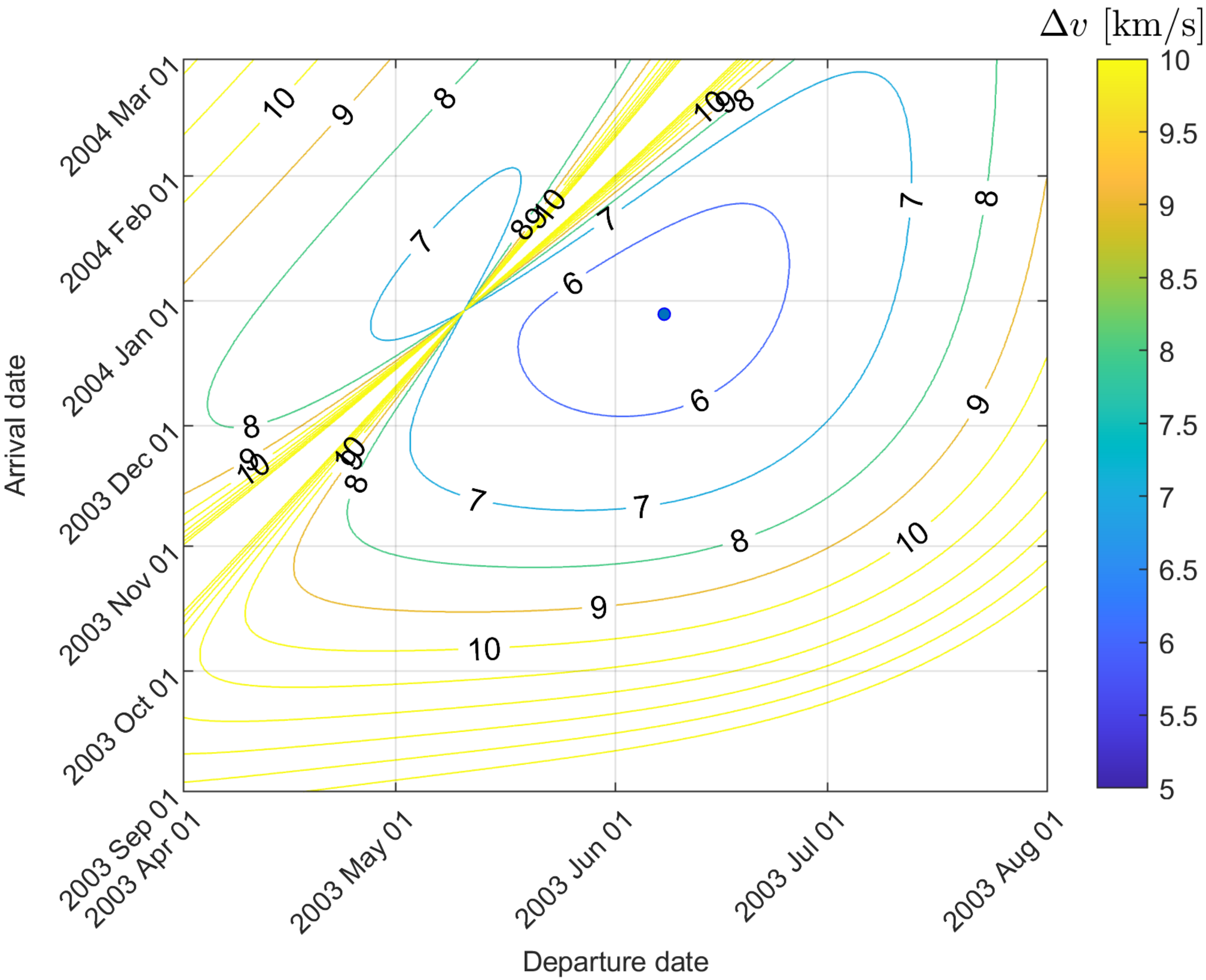
Mission analysis outputs

1. Implement a function to compute $\Delta v_{\text{tot}}(t_1, t_2)$.
2. Evaluate Δv_{tot} for a grid of departure and arrival times within the given time windows.
3. Draw the **porkchop plot** of the **Mars Express Mission**.
   Plot Δv_{tot} as a function of departure (x-axis) and arrival (y-axis) times, within their respective windows. Overlap to the contour plot some lines indicating Δt in days.
4. Find the cheapest mission (minimum of Δv_{tot}).
   Use function `min` over the array of Δv_{tot} values.
5. Plot the transfer trajectory for this mission, together with the orbits and initial/final positions of Earth and Mars.
6. **OPTIONAL**: Refine the solution using Matlab's `fminunc` or `fmincon` (unconstrained or constrained gradient-based optimization, respectively), taking the solution in **4.** as initial guess.

use the fixed time Transfer Designer

Exercise 3: Mars Express

Porkchop plot



Minimum Δv transfer:

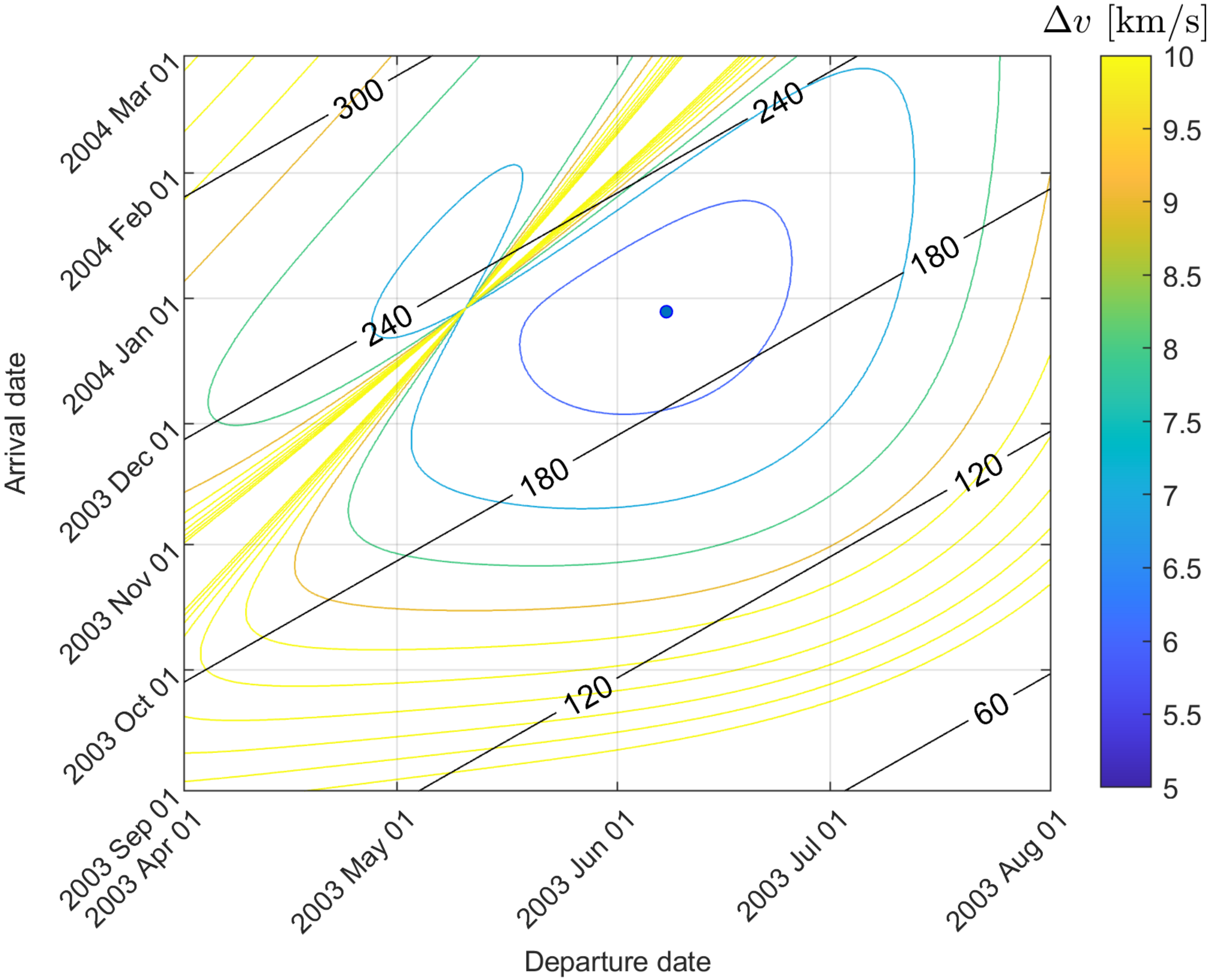
$\Delta v = 5.6670\ \mathrm{m/s}$

Departure:
2003/06/07 21:35:51.35

Arrival:
2003/12/28 14:24:51.89

Exercise 3: Mars Express

Porkchop plot



Minimum Δv transfer:

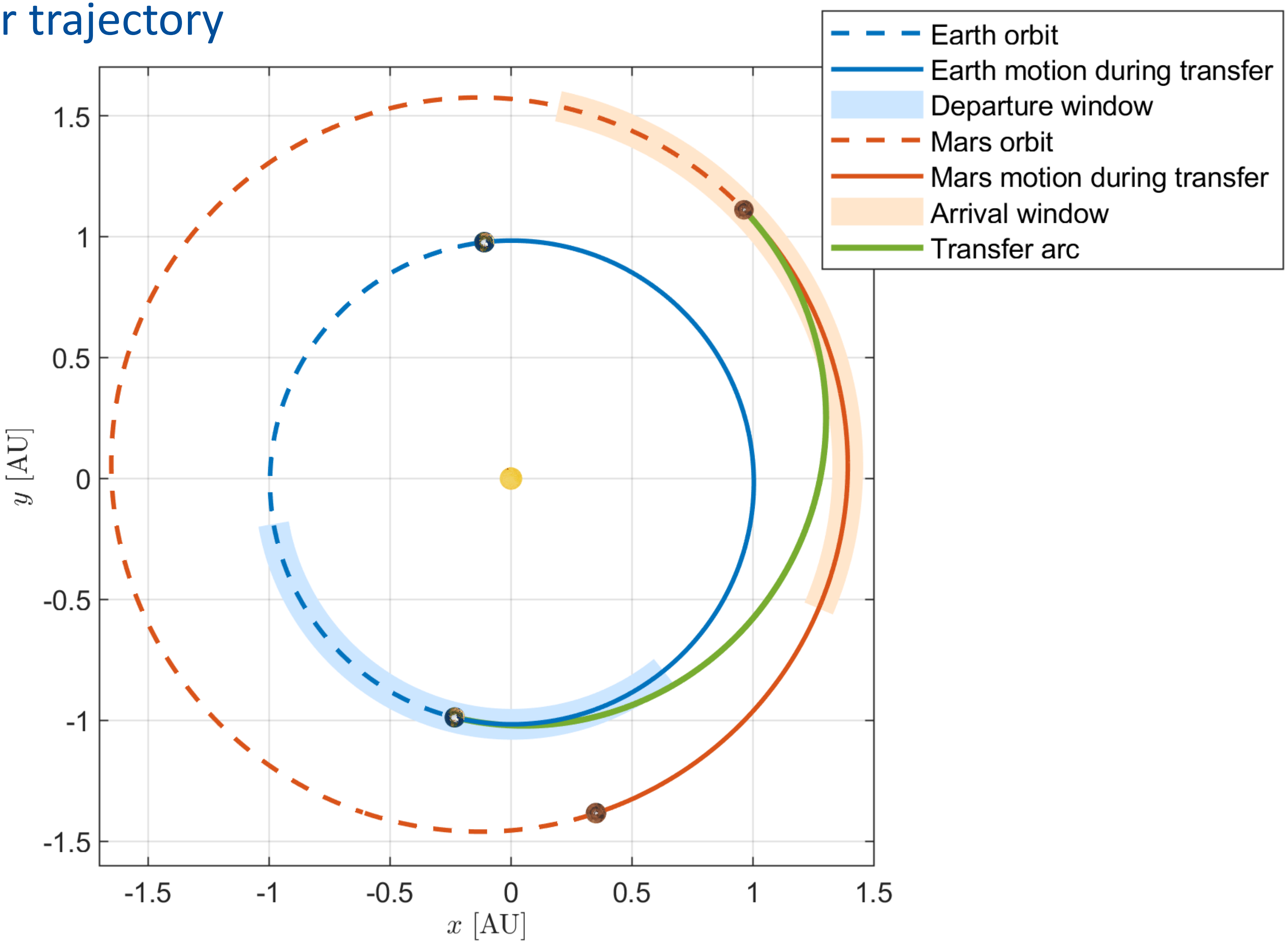
$\Delta v = 5.6670$ m/s

Departure:
2003/06/07 21:35:51.35

Arrival:
2003/12/28 14:24:51.89

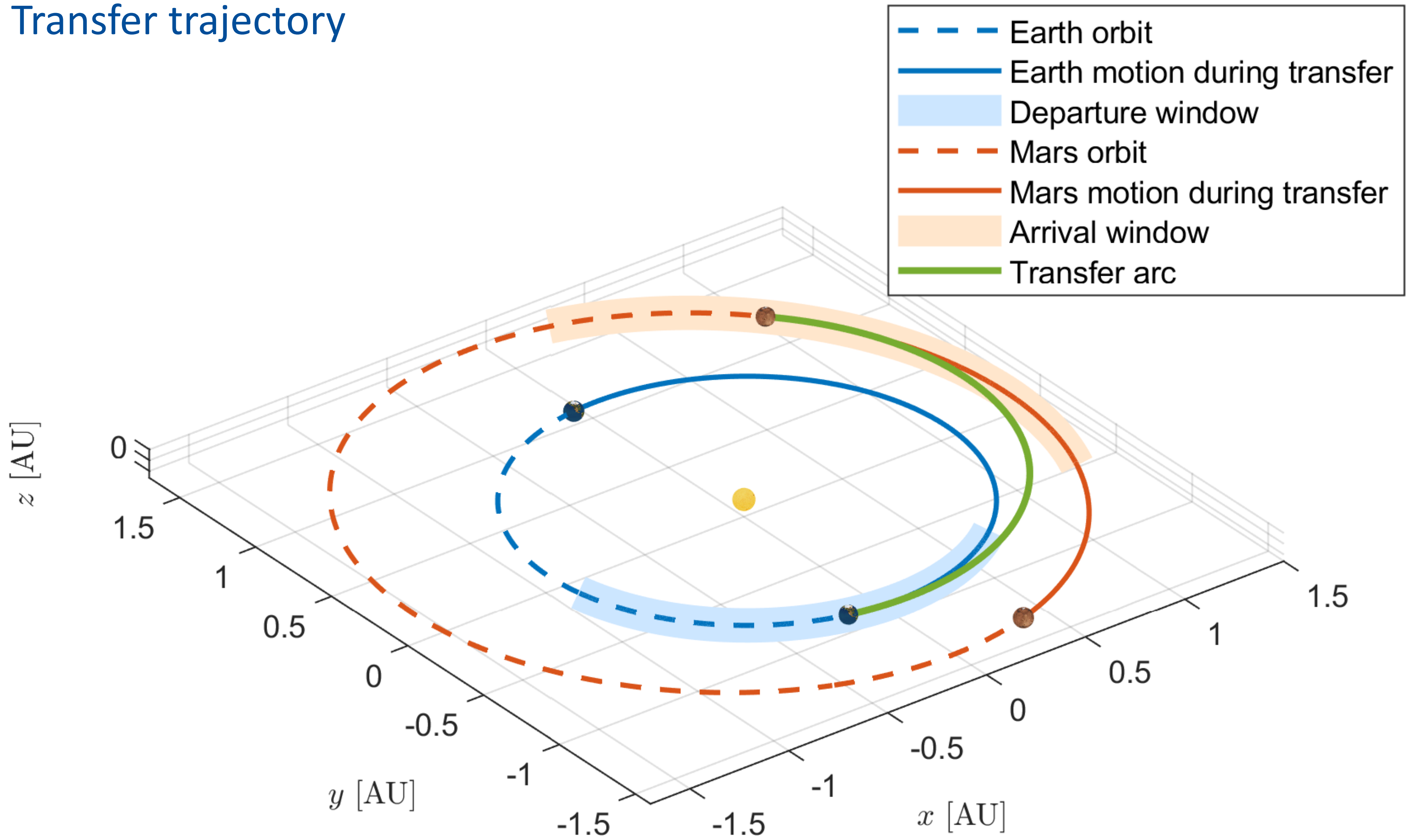
Exercise 3: Mars Express

Transfer trajectory



Exercise 3: Mars Express

Transfer trajectory



Mission Express

We have a mission!

As part of the mission analysis team of the **PoliMi Space Agency**, you are requested to perform the **preliminary mission analysis of an Express Mission** to rendezvous with a planet of the Solar System.

The launcher will inject the spacecraft directly into the interplanetary heliocentric transfer orbit. The maximum excess velocity and the available launch window are design constraints given by our launch provider.

The target planet and the arrival window are set by our science team.

Calculate the transfer options from Earth to the target planet/asteroid within the launch and arrival windows of your **Express Mission**, and select the one with minimum cost in terms of Δv.

Exercise 4: Mission Express

Mission definition

Mission Express: Design a direct transfer from Earth to a planet, with restricted launcher excess velocity.

Requirements for several missions are provided in the slides, with the following data:

Target planet	**n**	Integer (ephemeris ID)
Launch window	$t_1 = [t_{1\,min}\,, t_{1\,max}\,]$	Date format [yyyy, mm, dd]
Arrival window	$t_2 = [t_{2\,min}, t_{2\,max}]$	Date format [yyyy, mm, dd]
Maximum excess velocity from launcher	v_∞	[km/s]

Exercise 4: Mission Express

Mission analysis outputs

OK

1. Evaluate Δv_{tot} for a grid of departure and arrival times within the given time windows.

2. Draw the porkchop plot of the **Mission Express**. → OK

3. Find the minimum Δv_{tot}, without considering the launcher constraint. Although $||\Delta \mathbf{v}_1||$ is given by the launcher, we want to include it in Δv_{tot} because it gives a measure of the mission cost.

TO DO

Indices of PV_{MIN} : i_{MIN}, K_{MIN}

4. Find the cheapest mission (minimum Δv_{tot}) fulfilling the launcher constraint.

→ from r_0, v_0 → PROPAGATE → Plot TF ARC from departure → $T1(i_{MIN})$ to $T2(K_{MIN})$

↳ get a from car2kep(r_0, v_0) → T_{ORB} *

arrival time

5. Plot the transfer trajectory from **4.**, together with the orbits and initial/final positions of Earth and the target planet.

TO DO

↳ Plot with EPHEMERIDES vector A: (T_DEP_OPT : 1 : TDEP_OPT+TA)

6. **OPTIONAL**: Refine the solution using Matlab's `fminunc` or `fmincon`.

* Plot TF ORBIT from T_2 to T_1

$\Delta t = T_2 - T_1$ $\quad \Delta t_{ARC} + \Delta t_{ORB} = T_{ORB}$

Exercise 4: Mission Express

Mission data

Planet (ID)	Departure	Arrival	v_∞ [km/s]
Mercury (1)	2023/11/01 - 2025/01/01	2024/04/01 - 2025/03/01	7.0
Venus (2)	2024/06/01 - 2026/11/01	2024/12/01 - 2027/06/01	3.0
Mars (4)	2025/08/01 - 2031/01/01	2026/01/01 - 2032/01/01	3.5
Jupiter (5)	2026/06/01 - 2028/06/01	2028/06/01 - 2034/01/01	9.1
Saturn (6)	2027/09/01 - 2029/10/01	2030/04/01 - 2036/03/01	11.5
Uranus (7)	2027/01/01 - 2029/01/01	2031/04/01 - 2045/12/01	12.1
Neptune (8)	2025/01/01 - 2026/10/01	2036/01/01 - 2055/06/01	12.5