An Automatic Tree Search Algorithm for the Tisserand Graph

David de la Torre Sangrà
Polytechnic University of Catalonia, Physics Department, ESEIAAT, Colom 11, 08222 Terrassa, Spain.

Elena Fantino*
Department of Aerospace Engineering, Khalifa University of Science and Technology, P.O. Box 127788, Abu Dhabi, United Arab Emirates.

Roberto Flores
Department of Aerospace Engineering, Khalifa University of Science and Technology, P.O. Box 127788, Abu Dhabi, United Arab Emirates.
Centre Internacional de Mètodes Numèrics en Enginyeria CIMNE, 08034 Barcelona, Spain.

Oscar Calvente Lozano and Celestino García Estelrich
Polytechnic University of Catalonia, Physics Department, ESEIAAT, Colom 11, 08222 Terrassa, Spain.

The Tisserand graph (TG) is a graphical tool commonly employed in the preliminary design of gravity-assisted trajectories. The TG is a two-dimensional map showing essential orbital information regarding the Keplerian orbits resulting from the close passage by one or more massive bodies, given the magnitude of the hyperbolic excess speed (v_∞) and the minimum allowed pericenter height for each passage. Contours of constant v_∞ populate the TG. Intersections between contours allow to link consecutive flybys and build sequences of encounters en route to a selected destination. When the number of perturbing bodies is large and many v_∞ levels are considered, the identification of all the possible sequences of encounters through visual inspection of the TG becomes a laborious task. Besides, if the sequences are used as input for a numerical code for trajectory design and optimization, an automated examination of the TG is desirable. This contribution describes an automatic technique to explore the TG and find all the encounter paths. The technique is based on a tree search method, and the intersections between contours are computed using the regula-falsi scheme. The method is validated through comparisons with solutions available in the open literature. Examples are given of application to interplanetary mission scenarios, including the coupling with a trajectory optimizer.

---

*Corresponding author: elena.fantino@ku.ac.ae

I. Introduction

T‍he Tisserand graph (TG) is a graphical tool used in the preliminary design of gravity-assisted trajectories. By displaying essential orbital information about the Keplerian orbits resulting from close passages to a set of massive bodies, the TG helps construct a sequence of encounters between a starting and a destination orbit.

The TG is named after 19th century astronomer François Félix Tisserand, who developed a method -the Tisserand's criterion [1]- to identify an object (a comet or an asteroid) after a passage by a planet. The orbital elements of the object may change after the close approach, but the Tisserand parameter, a function of semi-major axis, eccentricity and inclination of the orbit, stays approximately constant and can be used to identify the object after the event. The outcomes of flybys with a planet are shown in the TG as curves, the v_∞ contours, corresponding to all hyperbolic passages with a given excess speed v_∞. This velocity is closely related to the Tisserand parameter. Each point along the contour corresponds to a different angle between the hyperbolic excess velocity of the spacecraft and the velocity of the planet. The intersections between contours link encounters with different planets. Hence, they can be used to build paths to a selected destination.

Figure 1 illustrates a TG for an Earth-to-Jupiter trajectory in which Earth and Venus flybys are the options considered [2]. The map shows the orbital periods and the perihelia of the heliocentric orbits that a spacecraft (S/C) can follow as a result of gravity assists with Earth (with v_∞ of 3 and 9 km/s) and Venus (v_∞ of 6 km/s) before approaching Jupiter at a relative speed of 6 km/s. The intersections between contours yield the following sequence of encounters: Earth (launch), Venus, Earth, Earth, Jupiter (arrival). **Two consecutive Earth flybys are required to respect the restriction of the maximum deflection angle during each passage (this is further ellaborated in Sect. III.A).**

The basic assumption of the TG is that all the planets are on circular coplanar orbits. The planet positions are not considered, i.e., flybys are assumed to occur whenever the orbit of the S/C intersects that of a planet. For this reason, this tool is **used during the** preliminary mission design stage, which must be followed by an analysis considering the phasing constraints.

The TG has been employed in interplanetary trajectory design for many years. **Strange & Longuski** [2] discussed Tisserand's theory in great depth, applied the TG to a wide number of transfers and highlighted the importance of an automatic exploration of the graph for complex scenarios. **Miller & Weeks** [3] reviewed the theoretical background of Tisserand's criterion and illustrated its application to the preliminary design of Cassini's interplanetary trajectory. **Heaton et al.** [4] used the TG in the design of tours in the Jovian system for the Europa Orbiter mission. They found sequences of lunar encounters (Europa, Ganymede, Callisto) which were then input to the Satellite Tour Design Program [5] designed for the Galileo S/C by JPL. That work emphasized the importance of an automatic method to search for transfers within the TG. **Heaton & Longuski** [6] designed a tour of the Uranus system using the TG to adjust the inclination of the target science orbit around Ariel. **Okutsu & Longuski** [7] employed a TG to design Mars free-return trajectories via gravity assists with Venus. **Khan et al.** [8] conducted the mission analysis for a two-S/C (relay and
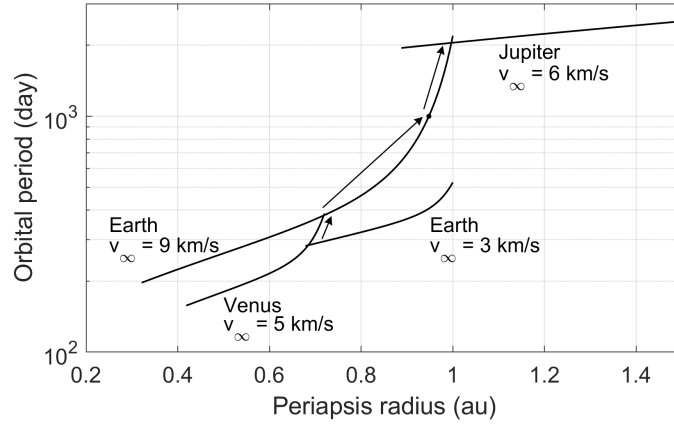
Fig. 1   TG for an Earth-to-Jupiter trajectory with intermediate Earth and Venus flybys [2].

orbiter) low-cost mission to Europa, in which the TG is used to identify tour options for both vehicles: a tour in the inner Jovian system for the orbiter and a tour of the outer, radiation-safe system for the relay. **Campagnola & Russell [9] and Strange et al. [10]** derived a new formulation for v_∞ leveraging maneuvers (VILMs) within the so-called Tisserand Leveraging Graph, used as a tool to design endgames. The new method allows rapid calculations of the minimum useful ΔV using VILMs to design resonant lunar tours at Jupiter and Saturn. **Campagnola et al.** [11] worked out a linear approximation to the solution space allowing fast sequence searches, and used the methodology to design a trajectory for an Enceladus orbiter. Then, **Campagnola & Russell [12] and Campagnola et al. [13]** extended the formulation of the TG to the circular restricted three-body problem (CR3BP). They found a trajectory encountering Callisto that inserts the S/C into a circular orbit around Europa, improving by 30% the ΔV budget of the classical patched-conics method. **Kloster et al.** [14] presented a design of a Jovian tour for an Europa orbiter mission in which the TG is used in combination with a simple radiation model to avoid hazardous exposures during the flybys. **Lantoine et al.** [15] relied on the Tisserand-Poincaré (T-P) graph, a variant of the TG for the CR3BP, to obtain initial guesses of inter-moon transfers in the Jovian system in a patched three-body model. **Hughes et al.** [16] investigated a broad collection of ballistic trajectories to Neptune using the TG for the selection of the planetary encounters. The trajectory was solved by patched conics with impulsive manoeuvres either in the form of powered gravity-assists or with VILMs, i.e., using deep-space manoeuvres to lower the launch ΔV. The authors highlighted the effects of phasing and mission constraints and estimated that only 21 out of 76 encounter sequences were feasible. **Strange et al.** [17] applied a TG-based method to the CR3BP to identify ways of capturing small asteroids around the Earth redirecting them to lunar gravity assists by means of a small (< 200 m/s) ΔV. **Colasurdo et al.** [18] employed the TG to design an efficient tour of the Galilean moons using resonant transfers **that achieved maximum surface coverage with minimum fuel consumption. It was the winning solution of** the 6th edition of the Global Trajectory Optimization Competition. **Campagnola et al.** [19] investigated three Jovian tour mission configurations (flyby-only, orbiter and lander) using the T-P graph. The solution

achieves low ΔV by means of high-altitude flybys and deep-space manoeuvres. **Maiwald** [20] adapted the TG to a low-thrust mission: the variation in the orbital energy over a thrust arc corresponds to a jump between different v_∞ contours in the TG. **Yarnoz et al.** [21] developed a systematic approach to generate multiple lunar flyby sequences for small interplanetary probes in a CR3BP, using the third-body perturbation of the Sun as a VILM equivalent. **Jones et al.** [22] employed the TG to study the triple cycler family of orbits among Earth, Mars and Venus. This type of trajectories periodically cycle between flybys of Venus, Earth, and Mars and were conceived for future manned mission to Mars. The solutions are characterized by lower ΔV requirements than traditional Earth-Mars cyclers.

When the number of planets and v_∞ levels increases (**Fig. 2**) the identification of all the possible sequences of encounters by visual inspection becomes impractical **[2]**. Besides, if the sequence of encounters is used as input of a numerical code for trajectory design and optimization, an automated examination of the TG is desirable. Automated
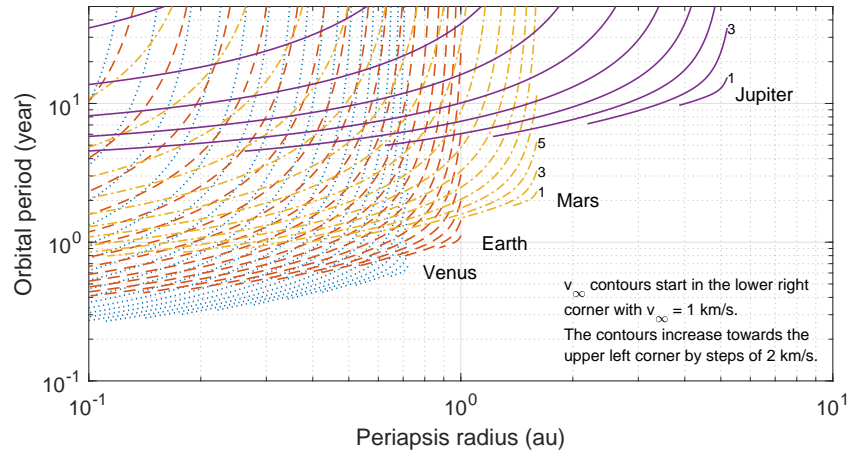

Fig. 2   TG for an Earth-to-Jupiter trajectory in which flybys with Earth, Venus and Mars are considered [2].

strategies for analysis of the TG are in widespread use, but, to the best of the authors' knowledge, they have not been documented in public literature. This contribution describes the development, implementation, validation and application of an automatic technique to explore the TG and determine all the sequences of flybys that it contains. The technique is based on a tree search method and will be referred to as the Tisserand PathFinder (TPF) algorithm.

This paper reviews the derivation of the Tisserand parameter and the procedure for constructing a TG, including simple guidelines for selecting the v_∞ levels. Next, the method to find intersections between contours, build a tree, explore it and construct sequences of encounters is described. Finally, a set of validation tests and two application examples are presented.

II. Tisserand's parameter

Tisserand's parameter is defined in a system of three bodies, two of which (**P$_1$ and P$_2$**, the primaries) are assumed in circular orbits about each other, and the third body **P$_3$** has negligible mass. This is the framework of the CR3BP. **Given the scope of our application, we shall consider only S/C trajectories contained on the plane of the primaries (zero inclination). Conservation of mechanical energy means there is a constant of motion for the third body, Jacobi's integral C_J [1]**

$$C_J = n^2 \left(x^2 + y^2\right) + 2\left(\frac{\mu_1}{r_1} + \frac{\mu_2}{r_2}\right) - \left(\dot{x}^2 + \dot{y}^2\right), \tag{1}$$

in which n is the mean motion of the primaries, μ_1 and μ_2 are their standard gravitational parameters and r_1 and r_2 are the distances from the third body to the two primaries (Fig. 3)). The reference frame is synodic (rotating, with the primaries fixed on the x-axis) and barycentric (with origin at the center of mass B of the system). In a barycentric inertial reference frame ξ, η – initially parallel to the synodic frame – Eq. 1 transforms into

$$C_J = 2n\left(\xi\dot{\eta} - \eta\dot{\xi}\right) + 2\left(\frac{\mu_1}{r_1} + \frac{\mu_2}{r_2}\right) - \left(\dot{\xi}^2 + \dot{\eta}^2\right). \tag{2}$$

Assuming that the first primary is the Sun and the second one is a planet, $(\mu_2/\mu_1)^{2/5} \ll 1$. Hence, the center of
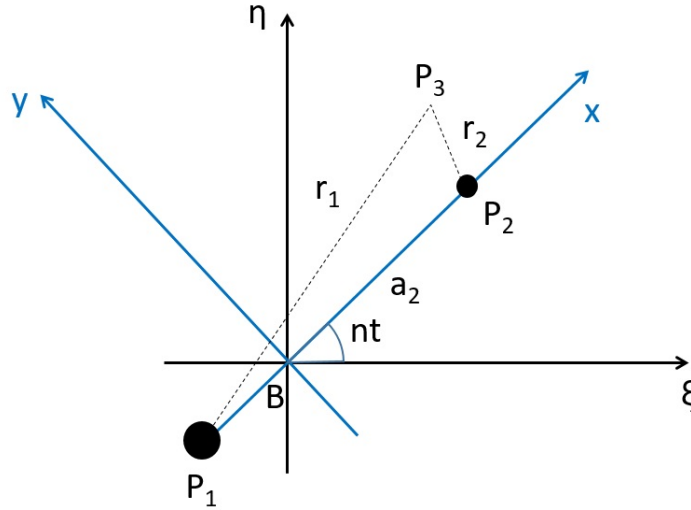

Fig. 3   The CR3BP, the barycentric synodic reference frame and the barycentric inertial reference frame.

mass of the system is very close to the Sun, and the radius of the sphere of influence of the planet is much smaller than the distance a_2 between primaries ([23] Chapter 7). Therefore, when the S/C approaches the planet, it is possible to approximate $r_1 \simeq a_2$ and $\mu_2/r_2 \simeq 0$ before entering the sphere of influence. Under these conditions, the heliocentric trajectory of the S/C is approximately Keplerian, and C_J can be rewritten in terms of the orbital elements. In particular, the last term of Eq. 2 can be approximated by the square of the Keplerian velocity v of

the S/C

$$\dot{\xi}^2 + \dot{\eta}^2 \simeq v^2 = \mu_1 \left(\frac{2}{r_1} - \frac{1}{a} \right) \simeq \mu_1 \left(\frac{2}{a_2} - \frac{1}{a} \right), \tag{3}$$

with a the S/C's semi-major axis. The first term on the right-hand side of Eq. 2 is proportional to the spacecraft's specific angular momentum h

$$\xi\dot{\eta} - \eta\dot{\xi} = \pm h = \pm\sqrt{\mu_1 a(1 - e^2)}, \tag{4}$$

where e is the orbital eccentricity. The plus sign corresponds to prograde orbits, while the minus sign must be used for retrograde trajectories.

Changing to dimensionless variables, using μ_1 and a_2 as reference magnitudes, the mean motion and heliocentric velocity v_2 of the planet take unitary values. The expression of the normalized Jacobi's integral $\bar{C}_J$ (hereinafter, dimensionless quantities will be denoted by barred symbols) reads

$$\bar{C}_J \simeq \pm 2\sqrt{\bar{a}(1 - e^2)} + \frac{1}{\bar{a}}, \tag{5}$$

The right-hand side of Eq. 3 is called Tisserand's parameter $\bar{C}_T$ [24]. As the S/C approaches the planet, the magnitude $\bar{v}_\infty$ of the relative velocity $\bar{\mathbf{v}}_\infty$ can be obtained applying the law of cosines to the triangle formed by the heliocentric velocity of the S/C ($\bar{\mathbf{v}}$) and planet ($\bar{\mathbf{v}}_2$) (Fig. 4):

$$\bar{v}_\infty^2 = \bar{v}^2 + 1 \mp 2\bar{v}\cos\gamma, \tag{6}$$

where γ is the S/C flight path angle and the $\mp$ symbol indicates that the negative sign must be taken for prograde orbits. Furthermore,

$$\bar{v}\cos\gamma = \bar{h}. \tag{7}$$

Combining Eqs. 3, 4, 6 and 7 yields

$$\bar{v}_\infty^2 = 3 \mp 2\sqrt{\bar{a}(1 - e^2)} - \frac{1}{\bar{a}}. \tag{8}$$

According to Eq. 8, the semi-major axis and eccentricity determine the magnitude of the hyperbolic excess velocity, and the same value of v_∞ can be obtained with different combinations of these parameters.

III. The Tisserand Graph

As shown by Eq. 5, in the planar approximation C_T depends only on two orbital elements, semi-major axis and eccentricity. Figure 4 shows the geometry of a flyby. The duration of the event is assumed negligible compared to the orbital period of the planet. Hence, the heliocentric position of the S/C through the flyby is approximately constant.

As a result of the close passage, the velocity of the S/C relative to the planet changes. The net effect is a rotation of the inbound hyperbolic excess velocity $\mathbf{v}_{\infty-}$ by an angle δ, yielding the outbound hyperbolic excess velocity $\mathbf{v}_{\infty+}$ (Fig. 4). Hereinafter, the magnitude of these two vectors will be indicated with v_∞. The heliocentric velocity of the S/C changes from $\mathbf{v}_-$ to $\mathbf{v}_+$, and this change affects both the direction and the magnitude of the vector. The pump angle α_- (respectively, α_+) is defined as the angle between $\mathbf{v}_2$ and $\mathbf{v}_{\infty-}$ (respectively, $\mathbf{v}_2$ and $\mathbf{v}_{\infty+}$). We shall refer to α_- as the entry pump angle and to α_+ as the exit pump angle. Due to the symmetry of the problem, we shall limit the discussion to pump angles in the range $[0°, 180°]$*. From geometry,

$$\alpha_+ = \alpha_- + \delta, \tag{9}$$

and $\delta \in [0°, 180°]$.

The law of cosines allows to compute the magnitude of $\mathbf{v}_-$ (respectively, $\mathbf{v}_+$) from $\mathbf{v}_{\infty-}$ (respectively, $\mathbf{v}_{\infty+}$) and α_- (respectively, α_+):

$$v_-^2 = v_2^2 + v_\infty^2 + 2v_\infty \cos\alpha_-, \tag{10}$$

$$v_+^2 = v_2^2 + v_\infty^2 + 2v_\infty \cos\alpha_+ = v_2^2 + v_\infty^2 + 2v_\infty \cos(\alpha_- + \delta). \tag{11}$$

Given v_∞ and α (α_- or α_+), the above formulas yield the magnitude of the heliocentric velocity (v_- or v_+). Then, Eq. 3 gives the corresponding semimajor axis, and the orbital energy ϵ follows from

$$\epsilon = -\frac{\mu_1}{2a}. \tag{12}$$

Equation 8 can then be solved for the eccentricity. In this way, curves of constant v_∞ can be represented in a 2D map whose axes portrait the semimajor axis and the eccentricity, or any equivalent pair of orbital parameters (for example, the orbital energy versus the periapsis radius, the orbital period versus the periapsis radius or, only for elliptical orbits, the apoapsis radius versus the periapsis radius, Fig. 5). This 2D map is the TG and is employed to visualize the effect of planetary flybys on the heliocentric Keplerian orbits of the S/C. Since $\mathbf{v}_{\infty-}$ and $\mathbf{v}_{\infty+}$ have the same magnitude, they correspond to the same v_∞ contour. Note that in Fig. 5 the contours $v_\infty =$ cons. are always monotonic, irrespective of the choice of variables. **This is a consequence of assuming a prograde orbit, which means taking only the minus sign in the $\mp$ symbol of Eq. 8. This gives a unique value of e for each (v_∞, a) pair. In that case, the function $e(a)$ for fixed v_∞ is single-valued. Therefore, by virtue of being continuous, it is also monotonic. However, when both v_∞ and δ are sufficiently large, retrograde heliocentric orbits might appear and the plus sign has to be considered**

---

*Switching the sign of the pump angle has no effect on the S/C heliocentric trajectory. It simply changes the sign of the radial S/C velocity while leaving the circumferential component unchanged. Thus, the two signs correspond to a pair of symmetric points of the same conic section.

too. If both prograde and retrograde orbits are taken into account, Eq. 8 gives two eccentricities for each (v_∞, a) pair, and monotonicity is lost (Fig. 6). This is undesirable, as it makes finding contour intersections much more complex. Monotonicity can be recovered by partitioning each contour into a prograde and a retrograde branch.
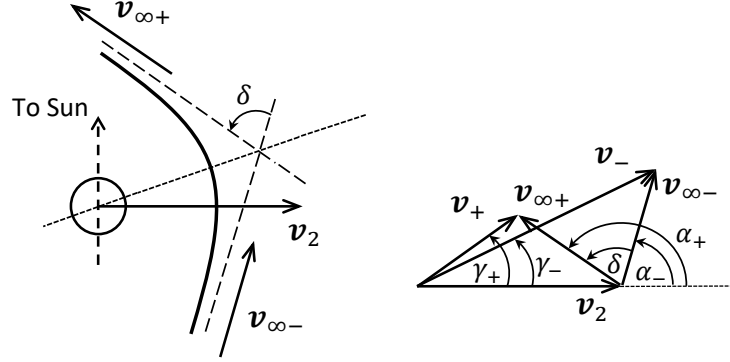

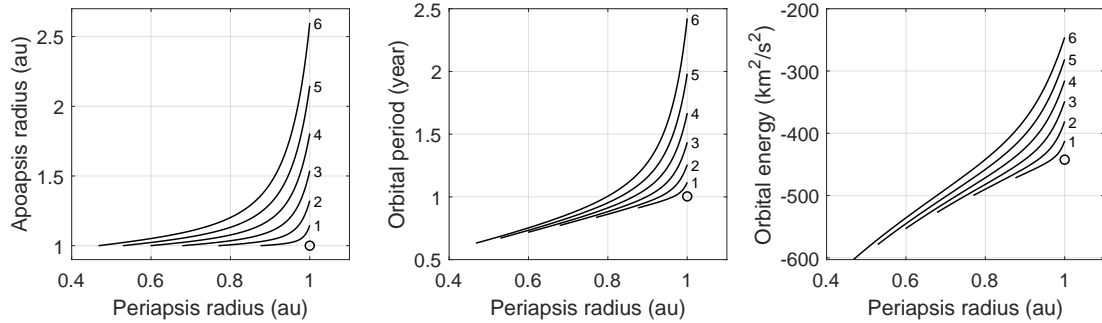**Fig. 4 Geometry of a flyby (left) and the associated velocity vector diagram (right).**

Fig. 5 Three versions of the same TG for different choices of the orbital parameters: apoapsis radius (left), orbital period (center) and orbital energy (right) versus periapsis radius. The circle represents the flyby planet, the Earth in this case, and the v_∞ contours correspond to values of the hyperbolic excess speed of 1, 2, 3, 4, 5 and 6 km/s, respectively.

A. Construction of the contours

The TG of Fig. 7 illustrates the steps taken in the construction of a v_∞ contour. The case shown corresponds to a v_∞ of 3 km/s relative to Earth. The circular orbit of the planet is the point (1 au, 1 au) (Fig. 7a)[†]. The upper point of the contour is obtained by setting $\alpha = 0$. The algorithm outlined above gives a periapsis radius of 1 au (in this case, the S/C's orbit is tangent to Earth's orbit) and an apoapsis radius of 1.54 au (Fig. 7b). Varying α between 0 and 180 degrees yields the entire contour (Fig. 7d).

When v_∞ contours of different planets are plotted in the same TG, their intersections correspond to orbits that can be linked by flybys with these planets. For example, in the sequence of Fig. 1, the S/C departs Earth on a 3 km/s contour,

---

[†]Note that the periapsis of the heliocentric orbit of the S/C can never be higher than the orbit of the flyby planet.
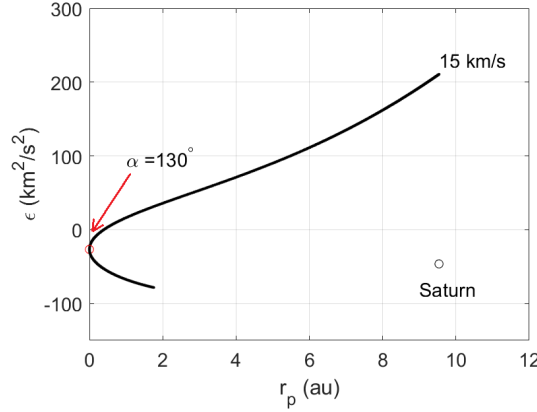
Fig. 6 A non-monotonic v_∞ contour for Saturn in which retrograde orbits (lower branch of the curve) appear for $\alpha > 130$ degrees.

performs a flyby with Venus at 5 km/s of v_∞ and passes by the Earth with a relative speed of 9 km/s twice (resonant flybys) before reaching Jupiter with a relative speed of 6 km/s.

A v_∞ contour can also be used to construct sequences of consecutive flybys with the same planet and characterized by the same hyperbolic excess speed. For example, in Fig. 7c, after tangential departure from Earth, the S/C returns after an integer number of revolutions, hence with the same encounter geometry ($\alpha_- = 0$) and heliocentric velocity. If $\alpha_+ = 45$ degrees, the new periapsis radius is 0.98 au and the apoapsis radius is 1.37 au.

The relationship ([25] Chapt. 2)

$$\sin\left(\frac{\delta}{2}\right) = \left(1 + \frac{r_\pi v_\infty^2}{\mu_2}\right)^{-1},\tag{13}$$

where r_π denotes the pericenter radius of the planetocentric hyperbola, links the minimum flyby height to the maximum achievable deflection angle δ_{max}. For example, if the minimum periapsis height above the surface of the Earth on a v_∞ contour of 3 km/s is 200 km, $\delta_{max} = 121$ degrees. δ_{max} in turn limits the maximum displacement along a v_∞ contour that can be achieved with a single flyby.

B. Selection of the v_∞ levels

Constructing a TG requires the selection of the planets and the identification of suitable v_∞ levels for each one of them. Even for the experienced orbit analyst, this task is not trivial, especially when the solution space is wide and involves several bodies. Furthermore, since the goal is to build sequences of flybys from departure to arrival, only those pairs of v_∞ contours that intersect each other are useful. Figure 8 provides some insight into this problem. Each panel refers to a specific pair of planets. The axes report v_∞ values for each planet, so that each point in the diagram can be associated with a pair of v_∞ contours. The shaded area shows the combinations of v_∞ for which intersection exist. The apex (marked with a solid circle) corresponds to a Hohmann transfer between the two planets. It is the heliocentric
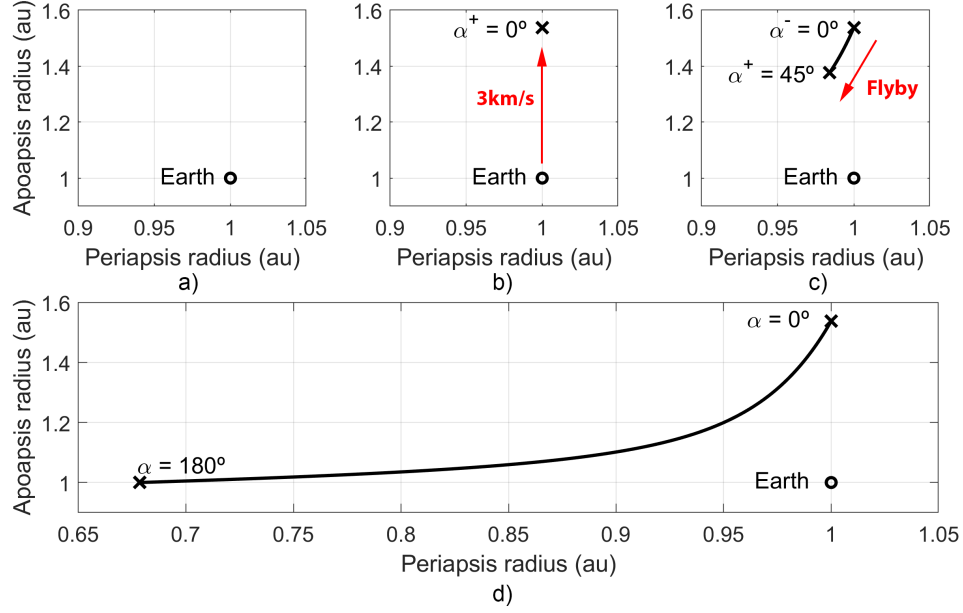
Fig. 7   Construction of a v_∞ contour.

ellipse that intersects both circular orbits with minimum hyperbolic excess velocity at departure and arrival. Therefore, the apex must lie at the bottom left corner of the shaded region. The charts can be used to narrow quickly the range of v_∞ contours that must be explored. For example, consider an Earth-Mars transfer for which the combination of launcher and payload limits the departure C_3 to 36 km^2/s^2 ($v_\infty = 6$ km/s). From Fig. 8, the useful v_∞ values for Earth lie between 3 and 6 km/s, while in the case of Mars we have to include contours from 2.7 to 10.5 km/s. Points (i.e., pairs of v_∞ values) outside the shaded area can safely be ignored, as they cannot yield an intersection. That is the case, for example, for the combination of $v_\infty = 7$ km/s at Mars and 11 km/s at Jupiter. This can be crosschecked against Fig. 2, which shows that, as expected, there is no intersection between those contours.

IV. Tisserand PathFinder

The TPF algorithm is based on the representation of the TG as a tree structure: an intersection between contours in the graph is a tree node and a transfer between intersections along a contour is a branch. The set of nodes and branches constitutes the tree. Each node has one parent node and may have one or more children nodes. A tree search algorithm has been applied to this representation in order to traverse the graph in an ordered way, collect transfers between planets and form encounter paths. Figure 9 (top) illustrates an interplanetary TG for a transfer from Earth to Mars with Earth, Venus and Mars flybys and two v_∞ contours for each planet (3 and 5 km/s). The black crosses mark contour intersections and the arrows signal transfers between planets. The construction of the tree structure from the TG is shown in the bottom part of the figure: starting from Earth with $\alpha = 0$ and $v_\infty = 3$km/s (root node), the possible paths are determined traversing the nodes and branches of the tree. The path indicated by the arrows goes through four nodes (Earth 3, Earth
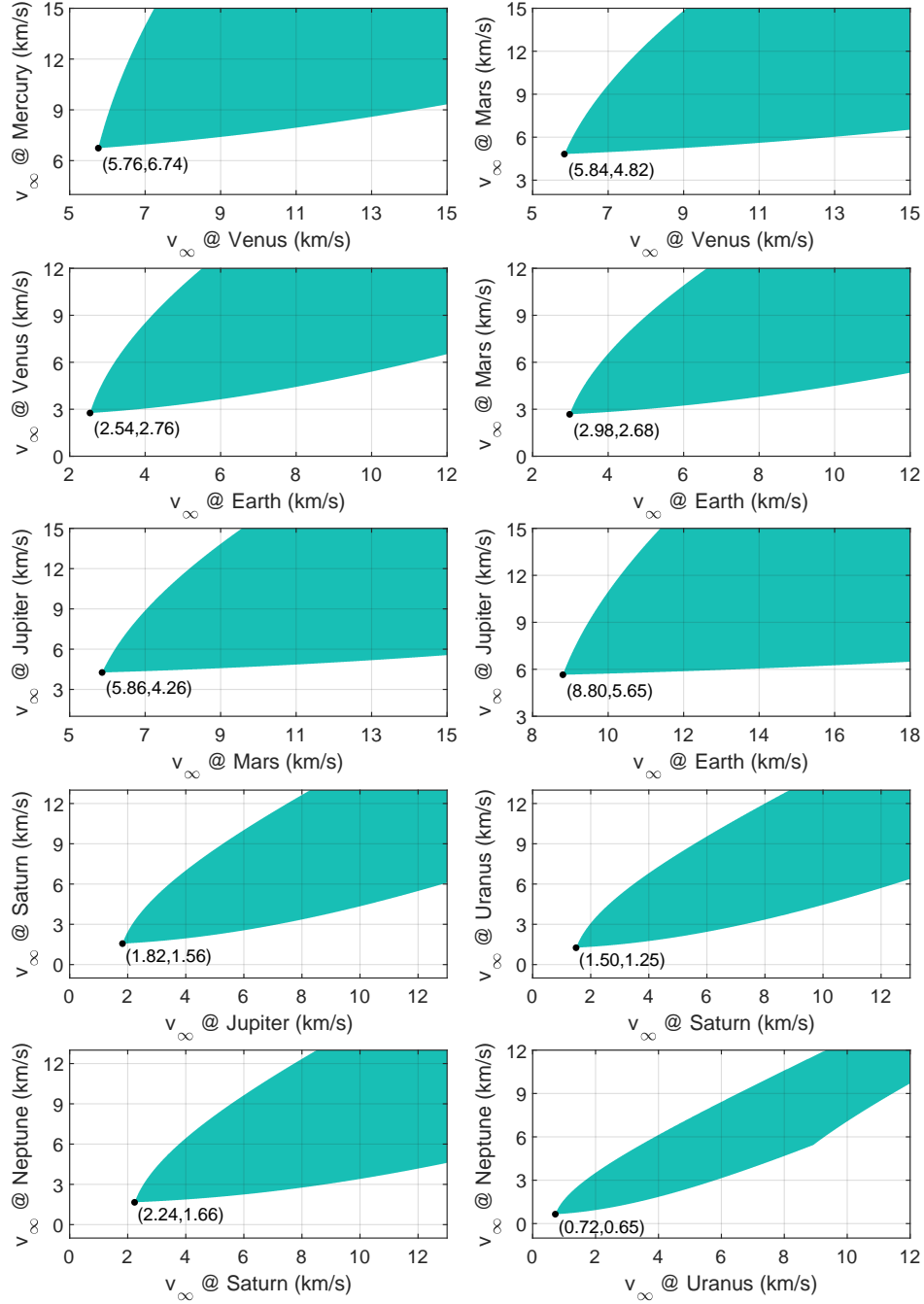
Fig. 8   Maps of intersections between v_∞ contours of different planets.

3 - Venus 5, Venus 5 - Earth 5, Earth 5 - Mars 5) and three branches (Earth 3 to Venus 5, Venus 5 to Earth 5, Earth 5 to Mars 5).
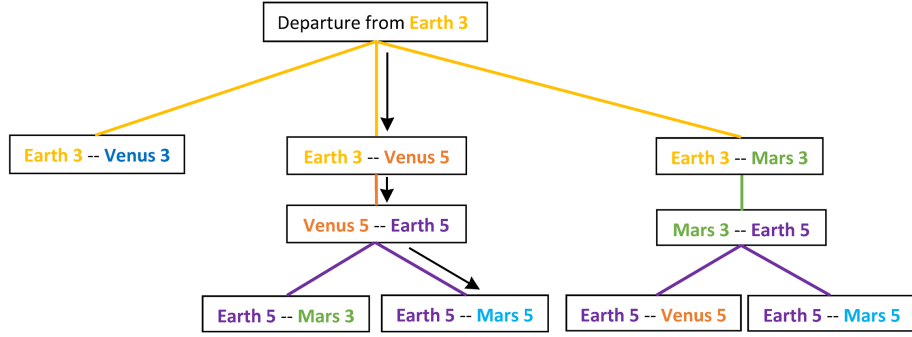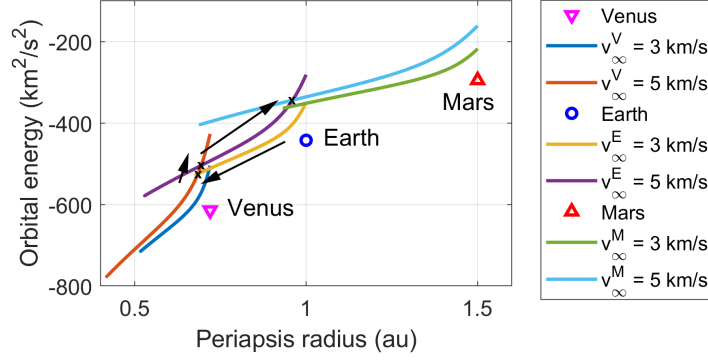
Fig. 9 TG (top) and the corresponding tree of encounter paths (bottom) for an Earth-to-Mars transfer in which flybys with Earth, Venus and Mars are considered. The arrows indicate an Earth-Venus-Earth-Mars path.

A. Determination of the intersection of two v_∞ contours

The intersections between v_∞ contours are determined finding the zeros of the function

$$f(\epsilon) = r_{p2}(\epsilon) - r_{p1}(\epsilon), \tag{14}$$

where the orbital energy is taken as the independent variable and r_{p1} and r_{p2} are the periapsis radii on the two contours. To solve Eq. 14, the regula-falsi technique [26] has been chosen due to its simplicity (no derivatives required) and robustness. The Illinois variant [27] of the algorithm is used for improved performance. At each iteration, ϵ is used to determine r_p in both contours following the procedure outlined in Fig. 10. The initial search interval is the range of ϵ common to both contours, i.e., between $\epsilon_a = max(min(\epsilon_1), min(\epsilon_2))$ and $\epsilon_b = min(\epsilon_1(\alpha = 0°), \epsilon_2(\alpha = 0°))$, as shown in the example of Fig. 11. The iterations stop when the absolute value of $f(\epsilon)$ falls below a specified tolerance. As a reference, it takes up to six iterations to reach an accuracy of 10 km.

B. Tree search

Tree search algorithms [28] are often used in optimization problems in which, given a starting point, several options must be explored to find the optimal solution. In our case, the goal is building a comprehensive set of candidate
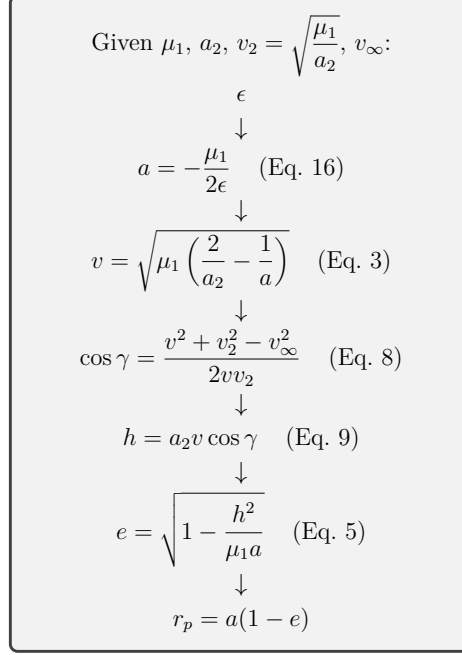
12

Given μ_1, a_2, $v_2 = \sqrt{\dfrac{\mu_1}{a_2}}$, v_∞:

$$\epsilon$$
$$\downarrow$$
$$a = -\frac{\mu_1}{2\epsilon} \quad \text{(Eq. 16)}$$
$$\downarrow$$
$$v = \sqrt{\mu_1\left(\frac{2}{a_2} - \frac{1}{a}\right)} \quad \text{(Eq. 3)}$$
$$\downarrow$$
$$\cos\gamma = \frac{v^2 + v_2^2 - v_\infty^2}{2vv_2} \quad \text{(Eq. 8)}$$
$$\downarrow$$
$$h = a_2 v \cos\gamma \quad \text{(Eq. 9)}$$
$$\downarrow$$
$$e = \sqrt{1 - \frac{h^2}{\mu_1 a}} \quad \text{(Eq. 5)}$$
$$\downarrow$$
$$r_p = a(1 - e)$$

Fig. 10 **Algorithm to obtain r_p from ϵ in a v_∞ contour relative to a planet with orbital radius a_2.**

solutions, i.e., to find all the paths in the TG connecting the departure planet with the target planet, regardless of their performance. Driven by this requirement, we chose an uninformed depth-first algorithm [29]: the uninformed tree search is suitable when there is no *a priori* knowledge of the tree, and the depth-first variant (which initially explores the nodes at the deepest levels of the tree and backtracks when it hits a dead-end) offers high speed and low memory consumption when dealing with complex trees.

The main drawback of the depth-first method is that it can get trapped in loops (in our case, repeated flybys with the same planet). To limit the number of planetary encounters and the computation time, the maximum depth of the tree is set by the user. Once the maximum depth is reached, the search does not proceed further along the current branch of the tree. This modified version of the depth-first method is referred to as the depth-limited search method. Program data is arranged in two main structures:

- Node database: It contains a list of the active nodes. Each node has a unique identifier (ID) and is associated with a parent node ID, a planet and specific values of v_∞ and ϵ. The parent node is the preceding encounter in the sequence of flybys. Once a node reaches the target destination, the parent IDs are used to rebuild the sequence of encounters.

- Stack: This is a FIFO (First-In First-Out) heap where the identifiers of nodes in process are stored. The depth-first search method uses the FIFO strategy: as the tree is expanded vertically, the most recently created nodes – those at the lowest levels of the tree – are processed first.
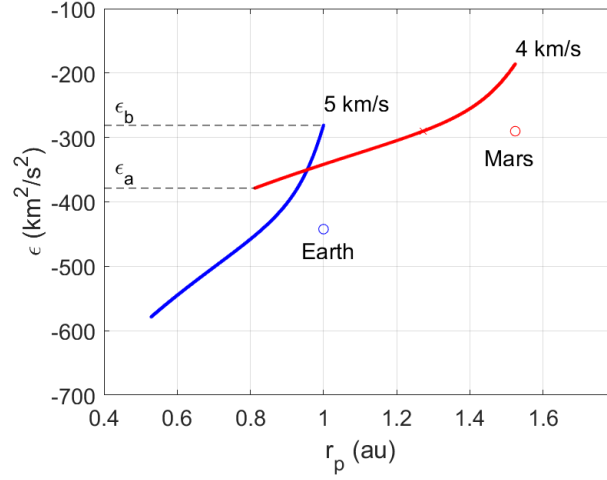
Fig. 11   Two intersecting contours for Earth and Mars with v_∞ of 5 and 4 km/s, respectively.
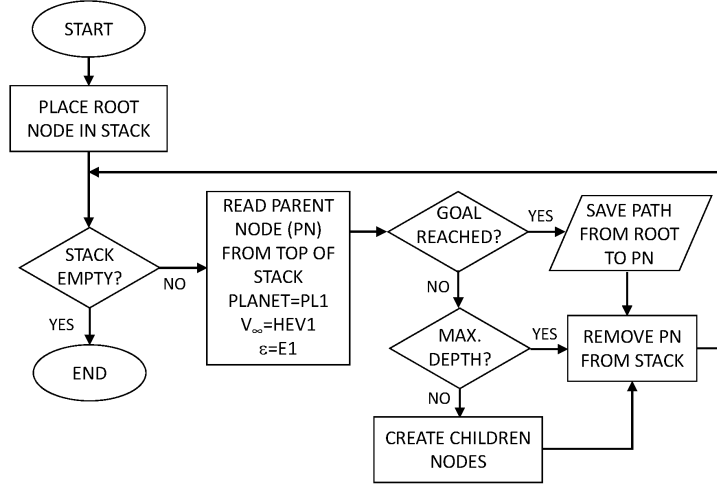

Fig. 12   Global functional diagram of TPF.

C. Search algorithm

To make the explanation easier to follow, we shall split the program structure in two distinct functional blocks:

- Processing of parent nodes (Fig. 12): These nodes are stored in the stack and were in turn created from other parents in previous iterations of the algorithm.

- Generation of children nodes (Fig. 13): These are the nodes that can be reached from the parent nodes by means of a flyby, and subsequently become new parents.

1. Processing of parent nodes

The nodes on the stack are examined and their children are generated through these steps:

1) Read input data: List of planets and v_∞ contours associated with each one, departure conditions (planet and v_∞

14

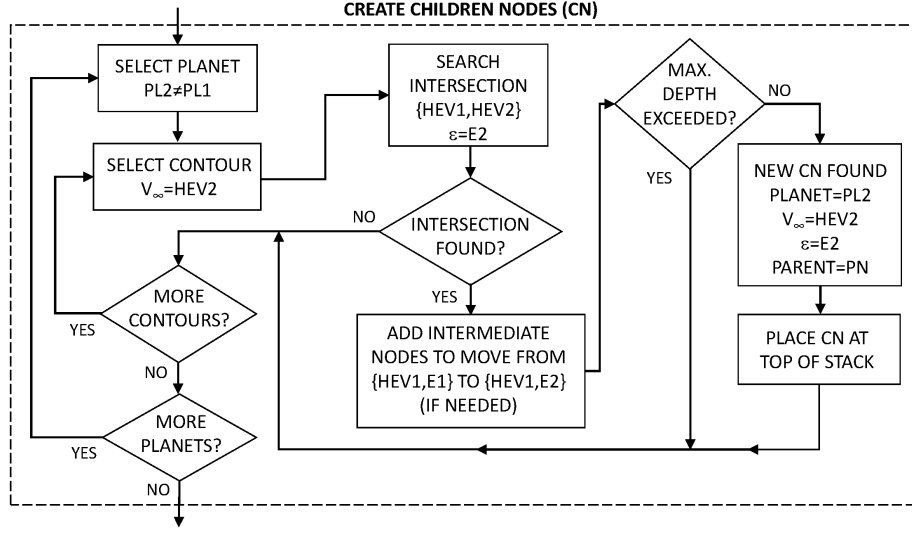**CREATE CHILDREN NODES (CN)**

Fig. 13   Flowchart of the children node generation block.

corresponding to the launch energy) and target (arrival planet and, if applicable, range of v_∞).

2) Initialize the stack with the root node (departure planet and v_∞).

3) If the stack is empty, terminate the program. Otherwise, read the node at the top of the stack. It becomes the current parent node (PN), associated with planet PL1, v_∞=HEV1 and specific orbital energy ϵ=E1.

4) Check if PN has reached the final state specified by the user (i.e., target planet and v_∞ level). If that is the case, rebuild the path from departure to destination planet using the parent node IDs, save it and proceed to step 6.

5) If PN has not reached the user-defined maximum depth, find its children nodes (see *2. Generation of children nodes*).

6) Purge PN from the stack and go back to step 3.

2. Generation of children nodes

The sequence of steps to create new Children Nodes (CN) is:

1) Select a candidate planet PL2 to explore (PL2≠PL1).

2) Select one of the v_∞ levels for PL2 (HEV2).

3) Search for an intersection between contours HEV1 of PL1 and HEV2 of PL2. Let E2 be the orbital energy of the intersection point (if it exists). If no intersection is found, proceed to step 6.

4) Add as many intermediate nodes as needed to move from E1 to E2 along the HEV1 contour. These nodes represent consecutive flybys with PL1 to achieve sufficient deflection angle without colliding with the planet. The intermediate nodes give rise to linear branches of the tree (i.e., no bifurcations) and they are never placed in the stack (because they require no further analysis). Intermediate nodes are important, however, because they

15

add to the tree depth.

5) If the maximum depth has not been exceeded, the intersection becomes a CN and is placed at the top of the stack. PL2, HEV2, E2 and PN are stored in the database entry for this CN.

6) If there are more v_∞ levels to explore for PL2, go back to step 2.

7) If there more planets to process, go back to step 1.

D. Garbage collection

Memory consumption can be an issue when evaluating large tree structures. To reduce the memory footprint of TPF, nodes that are no longer useful (e.g., nodes with no children or nodes from a branch that has been entirely explored) are deleted from the database (pruned) periodically. Because the database does not contain all the nodes analyzed, the information required to reconstruct the sequence of encounters is stored in the output file. Whenever a path to the target planet is found, the list of encounters and the corresponding nodes are stored in the output file. The list of nodes in the output data, while only a small subset of those explored, is sufficient to interpret the paths.

A Matlab script implementing the TPF algorithm has been published under LGPL license [30]. The results presented in the following sections have been computed in Matlab R2019a running under Windows 10 1803 in an Intel Core i7-6700K CPU with 4.00GHz and 32GB of RAM. In the discussion, the encounter paths are grouped into planet sequences, i.e., paths that connect the same ordered list of planets regardless of the v_∞ levels.

V. Validation

The results obtained with TPF have been compared with three solutions (named V1, V2, V3) presented in the literature.

- V1: from Earth to Mercury. The TG contained in the upper part of Fig. 14 shows two encounter paths identified by **Strange & Longuski** [2]: 1) Earth 3, Earth 3 - Venus 5, Venus 5 - Earth 7, Earth 7 - Venus 9, Venus 9 - Mercury 9; 2) Earth 3, Earth 3 - Venus 5, Venus 5 - Earth 9, Earth 9 - Mercury 11. TPF yields the same two paths (Fig. 14 bottom). Additionally, TPF indicates that the sequence of encounters for the first path is Earth-Venus-Earth-Venus-Venus-Mercury because a transfer to Mercury 9 from Earth 7 requires two consecutive Venus flybys to prevent a collision with the planet (due to the deflection angle limitation).

- V2: from Earth to Neptune. **Hughes et al.** [16] identify 72 planet sequences between Earth and Neptune. The path marked in Fig. 15 top is Earth 5, Earth 5 - Venus 7, Venus 7 - Earth 11, Earth 11 - Jupiter 7, Jupiter 7 - Neptune 3. TPF is able to find the same sequence (Fig. 15 bottom).

- V3: Venus-Earth-Mars cycler. **Jones et al.** [22] identify a triple cycler through Venus, Earth and Mars. The path highlighted in Fig. 16 top is Venus 4, Venus 4 - Earth 5, Earth 5 - Mars 3, Mars 3 - Earth 3, Earth 3 - Venus 4. TPF finds the same solution (Fig. 16 bottom).

Table 1 records the total CPU time, the number of encounter paths found and the number of planet sequences identified for the three validation cases.
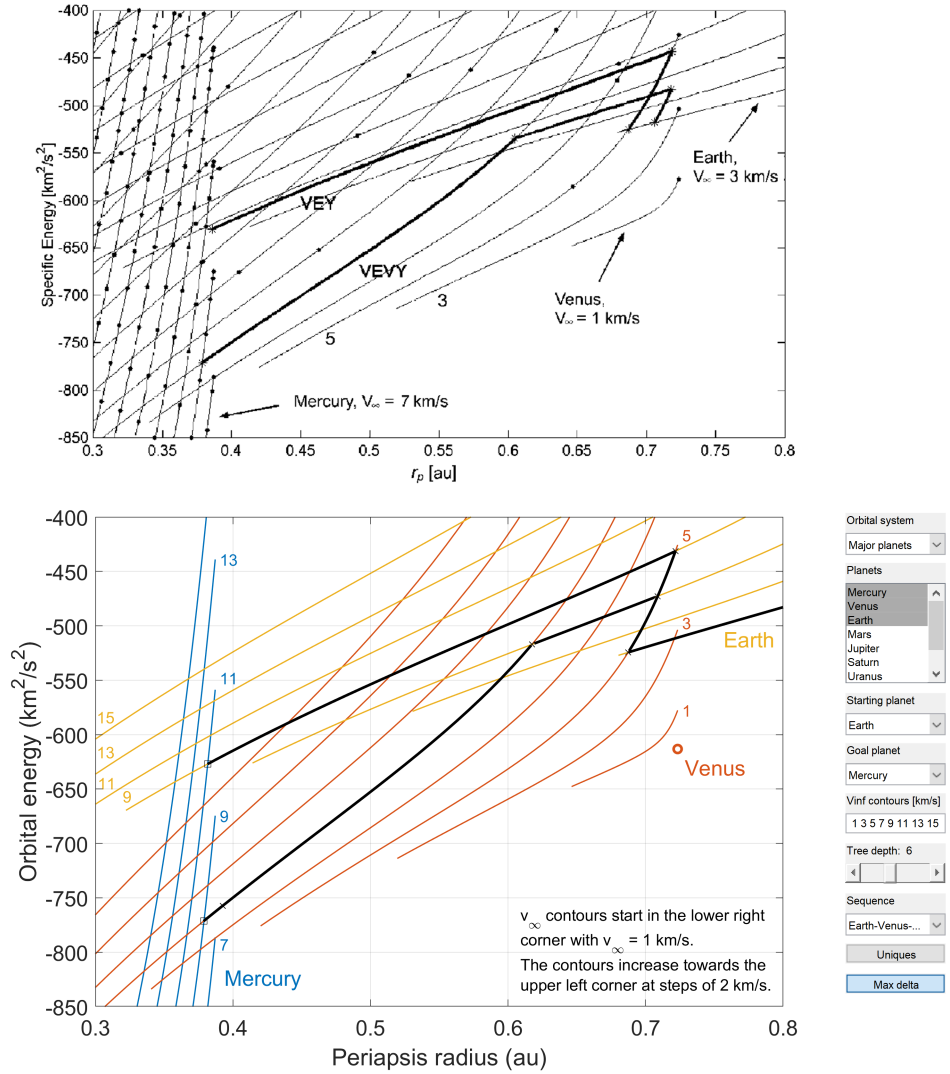

Fig. 14   Validation case V1: two Earth-to-Mercury paths found by Strange & Longuski [2] (top) and the same sequences obtained with TPF (bottom). The letters V, E and Y mean Venus, Earth and Mercury, respectively.

VI. Application to mission design

The following two mission scenarios have been designed and optimized with the aid of TPF: a transfer from Earth to Mars (M1) and the trajectory of JUICE [31] (M2). These applications illustrate the advantages of coupling TPF with a trajectory optimizer, allowing fully automated mission design and optimization. **The optimizer determines the launch conditions and impulsive maneuvers required (both in deep-space and during flybys) that connect the planets in each sequence and minimize the total propellant consumed. The user can specify an arbitrary number of mission constraints like departure window, maximum time of flight, launch energy and minimum flyby heights**

17

at each planet. **The software uses a collection of global optimization algorithms (genetic, differential evolution, particle swarm and simulated annealing) to find minima without computing the derivatives of the cost function. For this particular application the distinct paths of each planet sequence are not used, since the optimizer only requires an ordered list of planets and works by varying the encounter dates within each sequence. The TGs of the two scenarios have a maximum tree depth, including the departure and arrival planets, of seven encounters.**

- M1: the TG contains two v_∞ contours for each planet (Earth, Venus, Mars). In order to discard direct Earth-to-Mars transfers, the lowest v_∞ at Earth is set at 2.8 km/s. The path displayed in Fig. 17 is an Earth-Venus-Earth-Mars with Earth 2.8, Earth 2.8 - Venus 4, Venus 4 - Earth 4, Earth 4 - Mars 2.8. From the corresponding planet sequence, the interplanetary trajectory optimizer outputs a solution departing on 07/05/2023 with v_∞ of 2.8 km/s and arriving on 19/06/2025 with v_∞ of 3.0 km/s. The flybys with Venus and Earth (on 17/10/2023 and 08/70/2024) are powered and require velocity impulses of 913 and 155 m/s, respectively. The departure dates explored range from 01/01/2020 to 01/01/2026 and the maximum flight time between planets is set at 2 years.

- M2: the trajectory chosen for the JUICE mission to Jupiter [32] is of type Earth-Earth-Venus-Earth-Mars-Earth-Jupiter. The first Earth-to-Earth leg includes a deep-space v_∞ leveraging manoeuvre, capability not available in our trajectory optimizer. Due to this limitation, the TPF-generated sequence (obtained from three v_∞ contours at each planet) passed to the optimizer is Earth-Venus-Earth-Mars-Earth-Jupiter, as shown in Fig. 18 (Earth 6, Earth 6 - Venus 6, Venus 6 - Earth 10, Earth 10 - Mars 10, Mars 10 - Earth 12, Earth 12 - Jupiter 6). From this series, the optimizer generates a 7-year trajectory departing on 19/03/2023 with v_∞ of 4.9 km/s and performing powered flybys with Venus (28/10/2023, 84 m/s), Earth (08/08/2024, 148 m/s), Mars (14/02/2025, 1105 m/s) and Earth (16/11/2026, 394 m/s). The arrival v_∞ at Jupiter is 5.6 km/s. The range of launch dates explored is from 01/01/2023 to 31/12/2023, and the flight time between planets is limited to 2 years.

The performance of TPF for these scenarios is summarized in Table 1.

Table 1   CPU time, number of encounter paths and number of planet sequences for the three validation cases (V1, V2, V3) and the two mission scenarios (M1, M2) discussed in the text.

	CPU time (s)	Number of encounter paths	Number of planet sequences
V1	0.940	3376	12
V2	110.000	124104	101
V3	0.060	154	7
M1	0.004	4	2
M2	1.300	854	41

VII. Conclusions

We described the algorithm and underlying theory of an automated method (TPF) for extracting interplanetary paths from a Tisserand graph. The algorithm uses a depth-first tree search method, building an exhaustive collection of all the encounter sequences connecting the departure and arrival planets. The search is depth limited – there is a maximum number of flybys in the path – to keep the algorithm from creating long repetitive sequences of encounters (loops). The method identifies seamlessly those situations in which repeated encounters with the same planet are required to respect the minimum pericenter height constraint. The algorithm is completely automatic, relieving the user from the burden of visually identifying the contour intersections. The process is very fast in contemporary commodity hardware. Even the most complex scenarios can be analyzed in a matter of minutes (we presented a case with 124 000 distinct paths that completes in under 110 seconds). This efficiency gives the user additional freedom when choosing the number of excess velocity contours to inspect, because the impact on the duration of the analysis is quite limited. The automated inspection of the TG reduces the workload of the mission designer and eliminates the chance of human error, which is inherent to any tedious tasks. Furthermore, when TPF is coupled with a trajectory optimizer, the complete workflow of mission design and optimization can be streamlined.

Funding sources

References

[1] Roy, A. E., *Orbital motion*, 4th ed., Institute of Physics Publishing, Bristol, UK, 2005.

[2] Strange, N. J., and Longuski, J. M., "Graphical method for gravity-assist trajectory design," *Journal of Spacecraft and Rockets*, Vol. 39, No. 1, 2002, pp. 9–16. doi:10.2514/2.3800.

[3] Miller, J. K., and Weeks, C. J., *Application of Tisserand's criterion to the design of gravity assist trajectories*, 2002. doi:10.2514/6.2002-4717, paper AIAA 2002-4717.

[4] Heaton, A. F., Strange, N. J., Longuski, J. M., and Bonfiglio, E. P., "Automated Design of the Europa Orbiter Tour," *Journal of Spacecraft and Rockets*, Vol. 39, No. 1, 2002, pp. 17–22. doi:10.2514/2.3801.

[5] Diehl, R. E., Kaplan, D. I., and Penzo, P. A., "Satellite Tour Design for the Galileo Mission," *21st Aerospace Sciences Meeting*, 1983. doi:10.2514/6.1983-101.

[6] Heaton, A. F., and Longuski, J. M., "The feasibility of a Galileo-style tour of the Uranian satellites," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, 2003. doi:10.2514/2.3981.

[7] Okutsu, M., and Longuski, J. M., "Mars Free Returns via Gravity Assist from Venus," *Journal of Spacecraft and Rockets*, Vol. 39, No. 1, 2002, pp. 31–36. doi:10.2514/2.3778.

[8] Khan, M., Campagnola, S., and Croon, M., *End-to-End Mission Analysis for a Low-Cost, Two-Spacecraft Mission to Europa*, 2004. Paper AAS 041-132.

[9] Campagnola, S., and Russell, R. P., "The Endgame problem Part 1: V-Infinity Leveraging technique and the Leveraging graph," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 463–475. doi:10.2514/1.44258.

[10] Strange, N. J., Campagnola, S., and Russell, R. P., *Leveraging flybys of low mass moons to enable an Enceladus orbiter*, 2010. Paper AAS 09-435.

[11] Campagnola, S., Strange, N. J., and Russell, R. P., "A fast tour design method using non-tangent V-Infinity Leveraging Transfers," *Celestial Mechanics and Dynamical Astronomy*, Vol. 108, 2010, pp. 165–186. doi:10.1007/s10569-010-9295-1.

[12] Campagnola, S., and Russell, R. P., "Endgame Problem Part 2: Multibody Technique and the Tisserand-Poincare Graph," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 476–486. doi:10.2514/1.44290.

[13] Campagnola, S., Skerritt, P., and Russell, R. P., "Flybys in the planar, circular, restricted, three-body problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 113, 2012, pp. 343–368. doi:10.1007/s10569-012-9427-x.

[14] Kloster, K. W., Petropoulos, A. E., and Longuski, J. M., "Europa orbiter tour design with Io gravity assists," *Acta Astronautica*, Vol. 68, No. 7-8, 2011, pp. 931–946. doi:10.1016/j.actaastro.2010.08.041.

[15] Lantoine, G., Russell, R. P., and Campagnola, S., "Optimization of low-energy resonant hopping transfers between planetary moons," *Acta Astronautica*, Vol. 68, No. 7-8, 2011, pp. 1361–1378. doi:10.1016/j.actaastro.2010.09.021.

[16] Hughes, K. M., Moore, J. W., and Longuski, J. M., *Preliminary Analysis of Ballistic Trajectories to Neptune via Gravity Assists from Venus, Earth, Mars, Jupiter, Saturn, and Uranus*, 2013. Paper AAS 13-805.

[17] Strange, N., Landau, D., Chodas, P. W., and Longuski, J. M., *Identification of Retrievable Asteroids with the Tisserand Criterion*, 2014. doi:10.2514/6.2014-4458, paper AIAA 2014-4458.

[18] Colasurdo, G., Zavoli, A., Longo, A., Casalino, L., and Simeoni, F., "Tour of Jupiter Galilean moons: Winning solution of GTOC6," *Acta Astronautica*, Vol. 102, 2014, pp. 190–199. doi:10.1016/j.actaastro.2014.06.003.

[19] Campagnola, S., Buffington, B. B., and Petropoulos, A. E., "Jovian tour design for orbiter and lander missions to Europa," *Acta Astronautica*, Vol. 100, No. 1, 2014, pp. 68–81. doi:10.1016/j.actaastro.2014.02.005.

[20] Maiwald, V., "About Combining Tisserand Graph Gravity-Assist Sequencing with Low-Thrust Trajectory Optimization," *6th International Conference on Astrodynamics Tools and Techniques*, 2016.

[21] Yárnoz, D. G., Yam, H., Campagnola, S., and Kawakatsu, Y., "Extended Tisserand-Poincare Graph and Multiple Lunar Swingby Design with Sun Perturbation," *6th International Conference on Astrodynamics Tools and Techniques*, 2016.

[22] Jones, D. R., Hernández, S., and Jesick, M., *Low Excess Speed Triple Cyclers of Venus, Earth, and Mars*, 2017. Paper AAS 17-577.

[23] Kaplan, M. H., *Modern Spacecraft Dynamics and Control*, John Wiley & Sons, 1976.

[24] Murray, C. D., and Dermott, S. F., *Solar System Dynamics*, Cambridge University Press, Cambridge, UK, 2000. doi: 10.1017/CBO9781139174817.

[25] Curtis, H. D., *Orbital Mechanics for Engineering Students*, Butterworth-Heinemann, Oxford, UK, 2014.

[26] Dahlquist, G., and Bjorck, A., *Numerical Methods*, Dover Publications, Inc., USA, 2003.

[27] Dowell, M., and Jarratt, P., "A modified regula falsi method for computing the root of an equation," *Bit*, Vol. 11, No. 2, 1971, pp. 168–174. doi:10.1007/BF01934364.

[28] Cormen, T. H., Leiserson, C. E., Rivest, R., and Stein, C. ., *Introduction to Algorithms*, 2nd ed., MIT Press, 1990. doi: 10.1017/CBO9781107415324.004.

[29] Korf, R. E., "Depth-first iterative-deepening: An optimal admissible tree search," *Artificial intelligence*, Vol. 27, 1985, pp. 97–109.

[30] De La Torre, D., Fantino, E., Flores, R., García, C., and Calvente, O., "Tisserand PathFinder Algorithm," `https://gitlab.upc.edu/juno/tpf`, 2020.

[31] The JUICE Science Working Team, "Jupiter Icy moons Explorer Exploring the emergence of habitable worlds around gas giants," Tech. rep., European Space Agency, 2014. ESA/SRE(2014)1.

[32] Grasset, O., Dougherty, M., Coustenis, A., and et al., "JUpiter ICy moons Explorer (JUICE): An ESA mission to orbit Ganymede and to characterise the Jupiter system," *Planetary and Space Science*, Vol. 78, 2013, pp. 1 – 21. doi:10.1016/j.pss.2012.12.002.
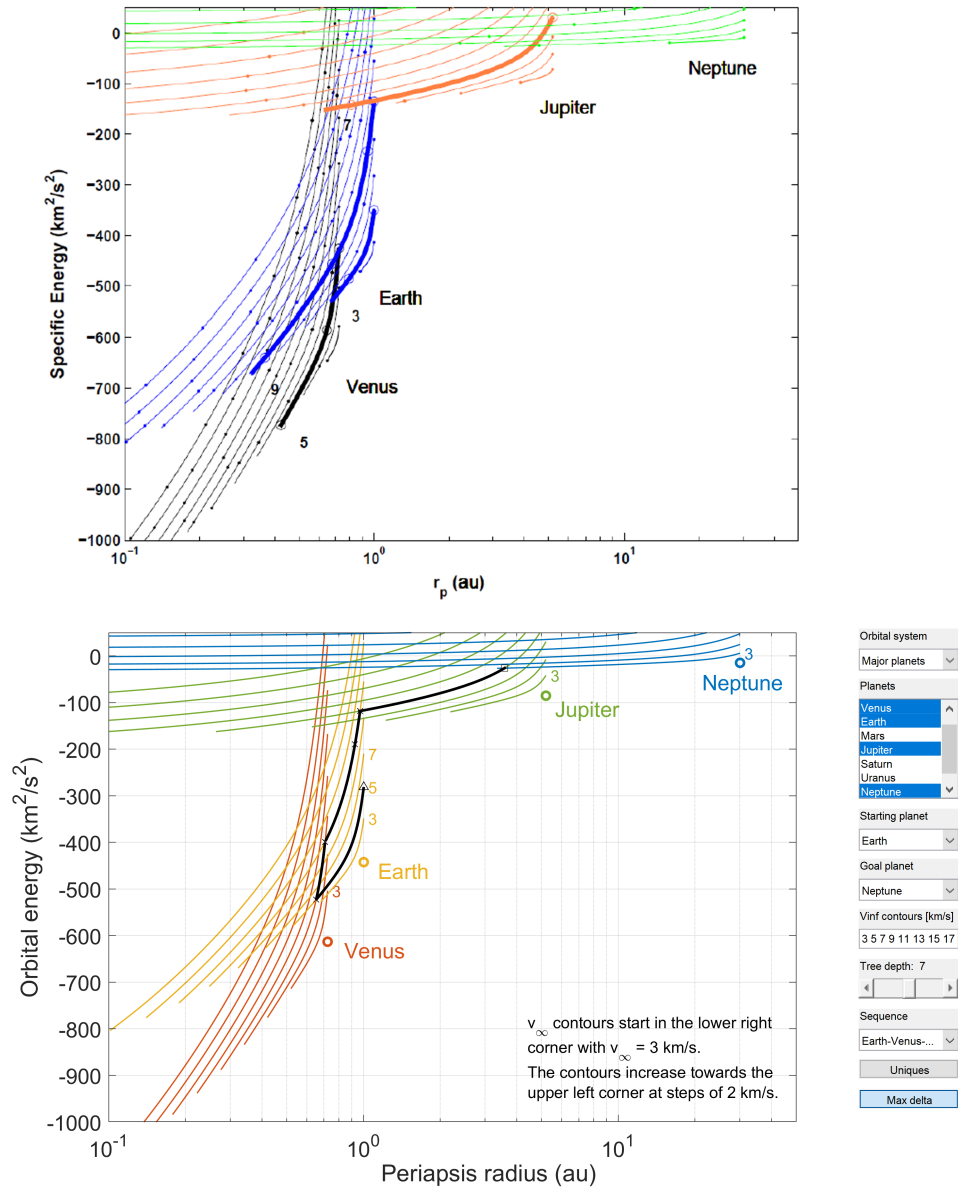
Fig. 15   Validation case V2: match between the Earth-to-Neptune path of Hughes et al. [16] (top) and that computed by TPF (bottom).
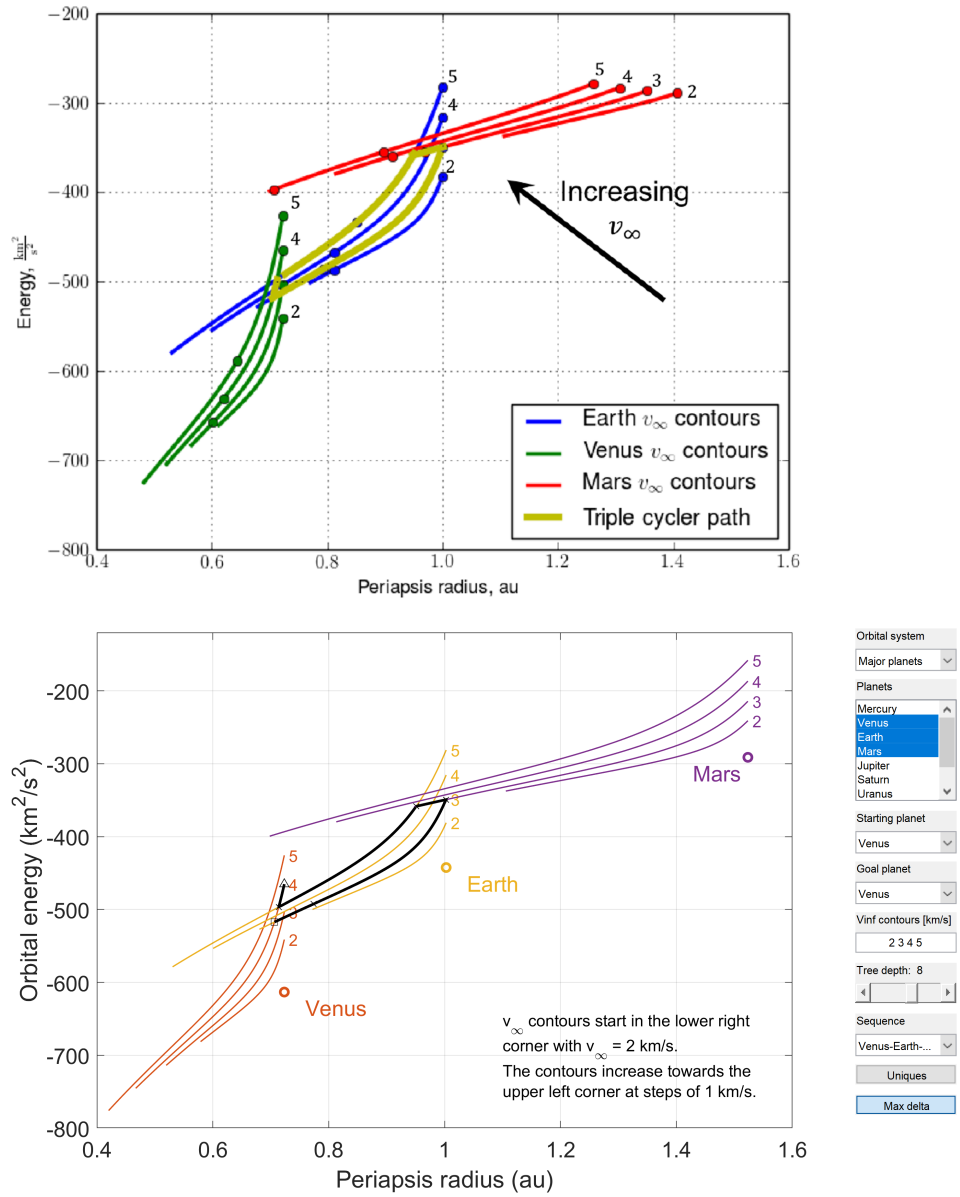
Fig. 16 Validation case V3: match between the Venus-Earth-Mars cycler path of Jones et al. [22] (top) and that computed by TPF (bottom).
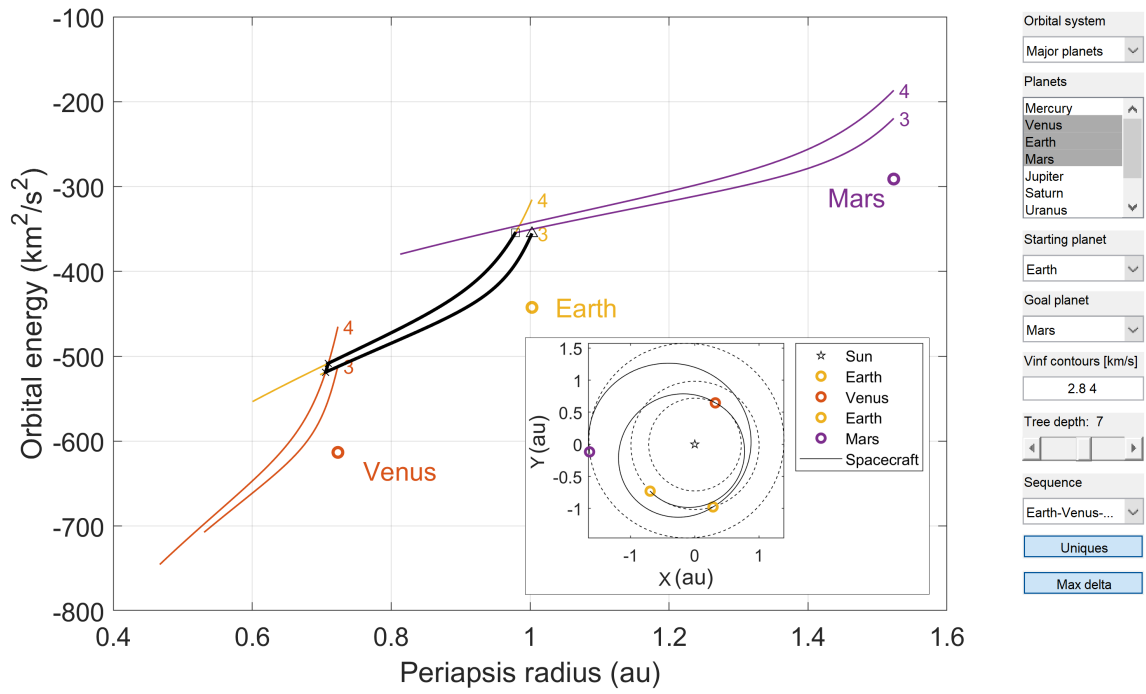
23

Fig. 17   Scenario M1: TG for an Earth-to-Mars transfer and the optimized interplanetary trajectory obtained from the sequence highlighted in the diagram.
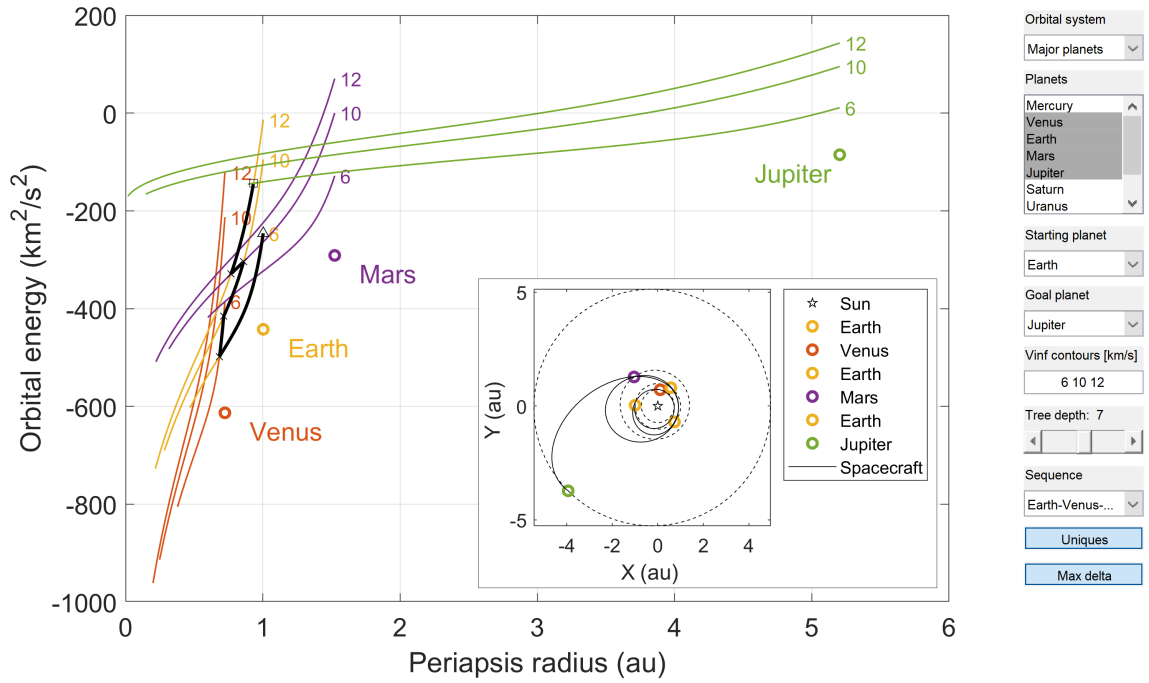

Fig. 18   Scenario M2: TG for a JUICE-like transfer and optimized trajectory.