Acta Astronautica 194 (2022) 216–228

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

# B-plane and Picard–Chebyshev integration method: Surfing complex orbital perturbations in interplanetary multi-flyby trajectories

Alessandro Masat [*,1], Camilla Colombo [2]

*Politecnico di Milano, 20156, Milano, Italy*

ARTICLE INFO

*Keywords:*
b-plane
Picard–Chebyshev
Flyby design
Perturbation surf

ABSTRACT

Orbital resonances have been exploited in different contexts, with the latest interplanetary application being the ESA/NASA mission Solar Orbiter, which uses repeated flybys of Venus to change the ecliptic inclination with low fuel consumption. The b-plane formalism is a useful framework to represent close approaches at the boundaries of the sphere of influence of the flyby planet. In the presented work, this representation is exploited to prune the design of perturbed resonant interplanetary trajectories in a reverse cascade, replacing the patched conics approximation with a continuity link between flybys and interplanetary legs. The design strategy splits the flyby time and state variables in a two-layer optimization problem. Its core numerically integrates the perturbed orbital motion with the Picard–Chebyshev integration method. The analytical pruning provided by the b-plane formalism is also used as starting guess to ensure the fast convergence of both the numerical integration and the trajectory design algorithm. The proposed semi-analytical strategy allows to take advantage of complex gravitational perturbing effects optimizing artificial maneuvers in a computationally efficient way. The method is applied to the design of a Solar Orbiter-like quasi-ballistic first resonant phase with Venus.

## 1. Introduction

Orbital resonances have been exploited in several ways for mission design purposes and in many different contexts, such as the Earth–Moon case (for example in the works of Topputo et al. [1] to actually reach the moon with low fuel consumption, of Ceriotti et al. [2] for increasing the observed regions by polar orbits, and Short et al. [3] as the actual scientific orbit of the Transiting Exoplanet Survey Satellite mission) or the exploration of Jupiter's and Saturn's moon systems (for example the works of Lantoine et al. [4], Campagnola et al. [5,6] and Vaquero et al. [7]). The planned introduction of the Lunar Gateway in 2024 has drawn the attention of more recent works on the cis-lunar space. The 9:2 resonant Near Rectilinear Halo Orbits are extensively analyzed by Zimovan et al. [8], both as possible candidates for the hosting the Gateway and in terms of the transfer possibilities toward other cis-lunar orbits by McGuire et al. with and without the aid of low-thrust propulsion [9]. Singh et al. [10] investigate eclipse-aware low-thrust transfer strategies to such orbits, proposing a method whose concept resembles the one of this work, leveraging the perturbation effects through the use of high-fidelity analogues of the invariant manifolds of the Circular Restricted Three Body Problem. Other applications also regard pure interplanetary orbits, for instance the ESA/NASA mission Solar Orbiter [11] as the latest example: resonant trajectories with Venus are exploited to raise the orbital inclination up to almost 30 degrees [12] over the ecliptic, to better observe the near-polar regions of the Sun.

In this last case, the use of resonant close encounters allows to save a considerable amount of fuel because of the repeated sequential flyby maneuvers. Nonetheless, such a phenomenon remains difficult to accurately model and understand, especially at the boundaries of the planet's sphere of influence where none of the two dynamics, planetary or interplanetary, has a dominant role. This effect is amplified for shallow encounters, where either the small relative velocity with respect to the flyby planet or the high miss distance worsen the patched conics approximation. However, accurate predictions are required for steep close approaches too: a small deviation from the nominal condition may be amplified by several orders of magnitude during the flyby, requiring trajectory correction maneuvers.

The b-plane formalism presents an analytic theory for the characterization of flybys, based on a manipulation of Öpik's variables [13] originally proposed by Carusi et al. [14] further developed by Valsecchi et al. [15–17]. Fixed values of the post-encounter semi-major axis are represented as circles in the b-plane, which can therefore be targeted a priori as the link with the orbital period is well known [15].

* Corresponding author.
*E-mail address:* alessandro.masat@polimi.it (A. Masat).
[1] Ph.D. Candidate, Department of Aerospace Science and Technology.
[2] Associate Professor, Department of Aerospace Science and Technology.

https://doi.org/10.1016/j.actaastro.2022.01.045
Received 28 August 2021; Received in revised form 9 December 2021; Accepted 31 January 2022
Available online 13 February 2022

On the other side, the Picard–Chebyshev method is a semi-analytical technique to globally integrate the evolution of a generic dynamical system accounting for a generic perturbation source. Picard iterations are performed to update the coefficients of a Chebyshev polynomial interpolation of an initial solution guess. A derivation of the method can be found in the work of Fukushima [18]. Bai and Junkins [19,20] proposed a modified version of the method, making it suitable to GPU (Graphics Processing Unit) computing platforms, condensing the algorithm steps in a series of matrix operations. Nonetheless, the modified Picard–Chebyshev method has been continuously developed in the past few years, both in its formulation and implementation side and outlining possible applications for Earth orbits where it contributed to increase the efficiency of the numerical analyses. Junkins et al. [21] analyzed the performances of the method comparing the efficiency against the Runge–Kutta–Nystrom 12(10) integrator, proposing also a second order version. Later, Koblick and Shankar [22] extended the analysis to the propagation of accurate orbits testing difference force models with NASA's Java Astrodynamics toolkit. Woollands et al. [23–25] applied the method as numerical integrator for the solution of the Lambert two-point boundary value problem, assessing also the benefits of adopting the Kuustanheimo–Stiefel formulation of the dynamics and proposing a solution for the multi-revolution trajectory design. Swenson et al. [26] applied the modified Picard–Chebyshev method on the circular restricted three-body problem, using the differential correction approach. Singh et al. [27] used the method as the numerical integration scheme for their feasibility study on quasi-frozen, near polar and low altitude lunar orbits, including the N-bodies and the spherical harmonics perturbations. The fixed point nature of the method was exploited by Koblick et al. [28] to design low-thrust trajectories as an optimal control problem, discretizing the control impulses and also included the Earth's oblateness $J_2$ perturbation. Macomber et al. [29] introduced the concepts of cold, warm, ho starts of the method, addressing possible efficiency improvements by means of better initial conditions, and variable-precision force models taking advantage of the fixed-point nature of the algorithm. Woollands et al. [30]extended the optimal low-thrust design to a high-fidelity model for the non-spherical Earth, considering an arbitrary number of spherical harmonics in the perturbing acceleration. Woollands and Junkins [31] developed the Adaptive Picard–Chebyshev method, including an integral error feedback that accelerates the convergence of the Picard iterations and an empirical low to determine segment length and polynomial degree of the method, based on previous stability analyses. Atallah et al. [32] compared the method with other sequential integration techniques on different Earth-based orbital cases.

In this paper a proof of concept is proposed, where the early basic formulation of the modified Picard–Chebyshev integration method [20] is combined with the b-plane flyby prediction capabilities and applied to the design of multi-flyby trajectories in reverse cascade. The exit requirements of the current flyby are computed to meet the entrance condition of the next one. They are consequently back-integrated to obtain a new entrance condition to be targeted, within a dynamic programming-like backward recursion logic. The proposed method extends the unperturbed design algorithm [33] previously developed by the authors of this work, that exploits the b-plane formalism to design a series of two body resonant orbits in the patched conics case. The newly extended version of the strategy proposed here uses the unperturbed b-plane solution to prune the trajectory design in the perturbed environment. Starting from the Keplerian initial guesses for the patched conics interplanetary arcs, a continuity link between the planetary and interplanetary legs is introduced at the boundaries of the planet's sphere of influence. The core of the presented approach numerically integrates the full dynamics using the Picard–Chebyshev method, embedded in a multi-layer optimization problem that aims to minimize an artificial correction at a user-specified point in the interplanetary cruise. This application also tests the Picard–Chebyshev integration techniques to interplanetary orbits, where the fixed point nature of the algorithm introduces further benefits compared to the sole Earth case. In particular, the numerical propagation scheme is used to remove the patched conics approximation, and to surf the complex perturbing accelerations from the N-bodies and general relativity. In summary, the b-plane formalism is used for both the preliminary design of the patched-conics initial trajectory guess and for the description of the optimization variables. The Picard–Chebyshev integration scheme is then used at the core of the optimization, exploiting the fixed point nature for increased computational performance when including the effects of N-bodies and general relativity perturbations.

The case of Solar Orbiter's resonant close approaches with Venus [12] is studied, achieving a quasi-ballistic transfer that surfs the chaotic perturbed environment, requiring a single artificial control impulse easily achievable by low thrust propulsion technologies.

The current implementation of the design strategy only requires a generic two-body patched conics solution in the b-plane formalism, not necessarily resonant. A generic solution from the Lambert problem [34] with consequent b-plane description of the planetocentric phase would suffice for the full extension to the design of non-resonant interplanetary arcs. In this work the resonant case is analyzed, since its connection with the b-plane formalism is straightforward [15] and the already available unperturbed design routine [33].

As the Picard–Chebyshev method can be parallelized, the whole design strategy is well suited to be used with high performance computing facilities. In spite of this, the serial execution of the initial formulation of the modified Picard–Chebyshev method will be shown to be already efficient, because of the minimized need to read the database for the ephemerides of the $N$ bodies. Further performance improvements are therefore expected including the latest adaptive version of the integrator [31].

The first steps toward the development of an efficient tool for the continuous design of perturbed multi-flyby trajectories are made, with particular focus onto the resonant ones. Addressing the behavior of the natural dynamics is fundamental before implementing any real mission maneuver design strategy.

This article is outlined as follows: first, a review of the b-plane representation of flybys is given in Section 2, followed by a recap of the Picard–Chebyshev integration method in Section 3, then the concept of proposed design algorithm is presented in Section 4. Finally, the application to a Solar Orbiter-like first resonant phase with Venus is shown in Section 5.

## 2. Close encounters in the b-plane

Assuming the planet in a circular orbit around the Sun, an intermediate frame needs to be defined for the b-plane flyby representation. Such a frame was first introduced in the framework of Öpik's theory [13] by Greenberg et al. [35] and later used by Carusi et al. [14] for the characterization of close encounters, particularly aiming to find analytic expressions for post-flyby orbital parameters. Consider a frame centered on the planet's center of mass, the $(x, y, z)$ axes are directed as the heliocentric position, velocity $\mathbf{v}_p$ and angular momentum of the planet, respectively, as shown in Fig. 1. $\mathbf{U}$ and $\mathbf{U}'$ denote the pre-encounter and post-encounter planetocentric velocities, respectively.

All the involved quantities are non-dimensional, such that the planet's distance from the Sun and the Sun's gravitational parameter are both equal to 1. The non-dimensionalization gives in turn $|\mathbf{v}_p| = 1$ and makes the orbital period of the planet equal to $2\pi$. The angles $\varphi$, $\varphi'$ and $\chi$ appear in the works of Carusi and Valsecchi [14,15] for other analyses, whereas are not necessary for the purposes of the presented design algorithm. $\gamma$ identifies the flyby turn angle, $\theta$ and $\theta'$ the pre and post encounter angles between the corresponding planetocentric velocity $\mathbf{U}$ or $\mathbf{U}'$ and the planet's velocity $\mathbf{v}_p$, and $\psi$ identifies the direction for the rotation of $\mathbf{U}$ into $\mathbf{U}'$ caused by the flyby, measured counter-clockwise from the major circles identified by $\mathbf{U}$ and $\mathbf{v}_p$.

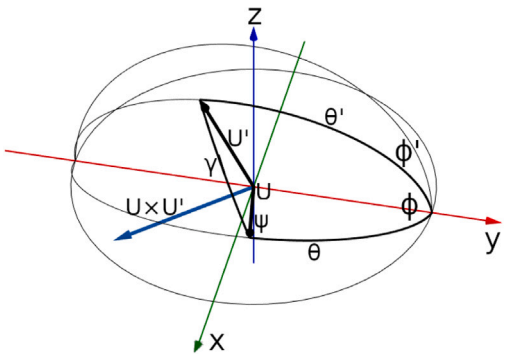



**Fig. 1.** Graphical representation of the reference frame of analysis.
*Source:* Picture re-drawn based on original from [14].

The flyby effect, interplanetary-wise in any patched conics approximation, is modeled as an instantaneous rotation of the planetocentric velocity vector $\mathbf{U}$ without magnitude change. With the above defined quantities it is possible to introduce the b-plane reference frame, whose axes $(\hat{\xi}, \hat{\eta}, \hat{\zeta})$ are defined as by Öpik [13]:

$$\hat{\eta} = \frac{\mathbf{U}}{|\mathbf{U}|}; \qquad \hat{\xi} = \frac{\mathbf{v}_p \times \mathbf{U}}{|\mathbf{U}|\ |\mathbf{v}_p|}; \qquad \hat{\zeta} = \hat{\xi} \times \hat{\eta}. \tag{1}$$

In the following, the definition *b-plane* will be used to identify the plane perpendicular to the $\hat{\eta}$ axis, because

$$\xi^2 + \zeta^2 = b^2 \tag{2}$$

with $b$ the impact parameter as in Milani et al. [36].

Recalling [14], from an interplanetary point of view the flyby can be modeled as an instantaneous rotation of $\mathbf{U}$ into $\mathbf{U}'$. The superscript $'$ shall denote the post-encounter quantities in the following lines.

### 2.1. B-plane circles

A certain post-encounter semi-major axis $a'$ is fully determined by $\theta'$ [15]:

$$\cos\theta' = \frac{1 - 1/a' - U^2}{2U} \tag{3}$$

From the b-plane properties and some spherical geometry analysis, the b-plane locus of points of a given post-encounter semi-major axis $a'$ is a circle centered on the $\hat{\zeta}$ axis [15]:

$$\xi^2 + \zeta^2 - \frac{2c\sin\theta}{\cos\theta' - \cos\theta}\zeta + \frac{c^2(\cos\theta' + \cos\theta)}{\cos\theta' - \cos\theta} = 0 \tag{4}$$

which is equivalent to

$$\xi^2 + \zeta^2 - 2D\zeta + D^2 = R^2 \tag{5}$$

with the center's $\zeta$ coordinate $D$ and the radius $R$ explicitly defined as

$$D = \frac{c\sin\theta}{\cos\theta' - \cos\theta} \qquad R = \left|\frac{c\sin\theta'}{\cos\theta' - \cos\theta}\right| \tag{6}$$

where, analogously to $\theta'$, $\theta$ is the angle between $\mathbf{U}$ and $\mathbf{v}_p$, and $c = \mu_p/|\mathbf{U}|^2$. As already mentioned, any reachable post-encounter semi-major axis can be drawn as a circle in the b-plane, and need not be resonant. The sole exception are flybys that do not modify the value of $a$, and thus feature $\theta \equiv \theta'$, which are defined as the straight horizontal line [15]:

$$\zeta = \cot\theta \tag{7}$$

### 2.2. Perturbations in the b-plane

Previous results by the authors of this work [33] led to the semi-analytical definition of the b-plane circles arising from the effects of a generic perturbation source. All the perturbing effects can be condensed in three angular variations:

- of the turn angle $\gamma$, $\Delta\gamma$;
- of the angle $\psi$ that identifies the direction of the rotation of $\mathbf{U}$ into $\mathbf{U}'$, $\Delta\psi$;
- of the post-encounter angle $\theta'$, $\Delta\theta'$.

Figs. 2(a) and 2(b) compare the resonant circles drawn with the unperturbed theory (Fig. 2(a) on the left) and the new perturbed model (Fig. 2(b) on the right) with the simulated resonant samples, highlighted in yellow, coming from the planetary protection analysis of the upper stage of the launcher of Solar Orbiter [12,37].[3] The b-plane circles, on purpose nearly visible and drawn in light gray, have become the black bounded belt shaped loci of points, because also almost perfectly phased resonant returns have been considered extending each circle over its own neighborhood.

In the case of Figs. 2(a) and 2(b) the angles $\Delta\gamma$, $\Delta\psi$ and $\Delta\theta'$ remain small in magnitude, nevertheless the difference they make in the characterization of the b-plane circles is significant. This gives a further proof to the need of precise models for the flyby phase, which is required if the desired post-encounter prediction must be accurate.

## 3. Picard–Chebyshev integration method

Picard iterations [40] are a method that can be used to obtain an approximation of the solution of initial/boundary value problems. Denoting the state of dimension $n$ with $\mathbf{x}$, the independent variable with $t$, the initial/boundary condition with $\mathbf{x}_0$ and the dynamics function with $\mathbf{f}(\mathbf{x}, t)$, the problem is defined as:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \qquad \mathbf{x}_0 = \mathbf{x}(t_0) \tag{8}$$

Starting from an initial approximation $\mathbf{x}^{(0)}(t)$ of the actual solution $\mathbf{x}(t)$ in the interval $[t_0, t]$ of the initial/boundary value problem presented in Eq. (8), the $i$th Picard iteration improves the previous approximation $\mathbf{x}^{(i-1)}(t)$ of $\mathbf{x}(t)$ with $\mathbf{x}^{(i)}(t)$ as in [40]:

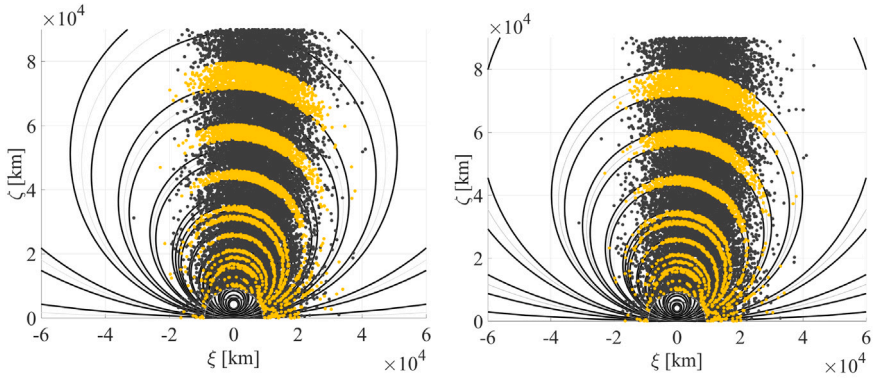
$$\mathbf{x}^{(i)}(t) = \mathbf{x}^{(0)}(t) + \int_{t_0}^{t} \mathbf{f}\left(\mathbf{x}^{(i-1)}(s), s\right) ds \tag{9}$$

The method converges for a good enough initial approximation $\mathbf{x}^{(0)}(t)$ and for $i \longrightarrow +\infty$ [40].

In the analytical Picard iteration context, performing more than one iteration is in general hard. The increasingly complex expressions for $\mathbf{x}^{(i)}(t)$ make it difficult to retrieve closed form solutions after the first 2–3 steps [19]. At the same time, numerically computing the integral functions by quadrature might not suffice in accuracy, as only the first few iterations in general improve the function approximation. In the attempt to develop parallelizable routines for the integration of the dynamical motion, the Picard–Chebyshev method was built combining the Picard iterations with the Chebyshev polynomial approximation [41]. A possible derivation of the method that follows the work of Fukushima [18] is reported in Appendix, and can be summarized in three steps:

1. Select a good enough initial guess $\mathbf{x}^{(0)}(t)$.
2. Approximate $\mathbf{f}(\mathbf{x}, t)$ and $\mathbf{x}^{(0)}(t)$ with their Chebyshev polynomial expansion.

[3] More detailed information about this analysis and the related validation can be found in the work of Colombo et al. [38], Colombo et al. [39] and Masat [33].




(a) Standard b-plane resonant circles. (b) Perturbed b-plane resonant circles.

**Fig. 2.** Visual accuracy improvement of the b-plane circle model. The analytical belts are bounded by the black circles, the yellow dots highlight the numerically detected resonances on the whole simulated cloud of initial conditions.

3. Perform a Picard iteration to update the coefficients of the interpolating Chebyshev polynomials.

The Picard iterations halt when the stopping conditions are met, based on the maximum difference between two consecutive iterations dropping below some user-specified tolerance.

The so defined method allows to easily perform several more Picard iterations than the analytical case. The involved expressions remains always of the same type, i.e. the Chebyshev polynomials. The function approximation becomes an interpolation through nodes that should be close to the true trajectory, instead of a global function whose value after the iterations still depend on the initial guess choice. Furthermore, few iterations suffice to drop below a low tolerance if the real solution $\mathbf{x}(t)$ differs from the initial guess $\mathbf{x}^{(0)}(t)$ only because of small perturbations [40]. Starting from the unperturbed Keplerian solution for the generic weakly perturbed two body problem, a relatively fast convergence of the method is ensured [18]. In the context of orbital simulations, Macomber [42] referred to this type of initial guess as warm-starting the Picard–Chebyshev iteration method, because the analytic solution of the dominant dynamics part is used to reduce the number of iterations required. Differently, the cold start was defined by simply setting all the trajectory samples as equal to the initial condition. In general, the closer the initial guess to the true trajectory, the lower the number of iterations will be. Semi-analytic initial guesses or results of propagations from simpler models are also an options, and in the case of three-body-like perturbed trajectories would be a better choice compared to the Keplerian approximation. Macomber also introduced the concept of hot start in the case of time spans covering multiple Earth planetary orbits [42], where the first orbit was used to compute the difference between the Keplerian guess and the converged trajectory. The near-periodicity of the spherical harmonics perturbation was then exploited, including this difference in the starting trajectory, achieving a further reduction of the iterations required for convergence.

### 3.1. Matrix form for vectorized and parallel computation

The method is suitable for parallel or vector implementation, indeed Fukushi-ma also proposed a vectorized version [43]. More recent works over this technique by Bai and Junkins developed the modified Picard–Chebyshev method [20] and a CUDA implementation for NVIDIA GPUs [19]. For compactness and to better highlight the parallelization possibilities, the method is presented following the matrix formulation by Koblick et al. [44].

For $N$ Chebyshev nodes and the integration interval $[t_0, t_{N-1}]$, the independent variable $t$ is sampled for $j = 0, 1, \dots, N-1$ up-front as

$$t_j = \omega_2 \tau_j + \omega_1 \tag{10}$$

with

$$\tau_j = -\cos\left(\frac{j\pi}{N-1}\right), \quad \omega_1 = \frac{t_{N-1} + t_0}{2}, \quad \omega_2 = \frac{t_{N-1} - t_0}{2} \tag{11}$$

Given the $n$-dimensional sampled states $\mathbf{y}^{(i-1)}(t_j) = \mathbf{y}_j^{(i-1)}$, $j = 0, \dots, N$ as a matrix $\mathbf{y}^{(i-1)}$ of dimension $N \times n$ computed at the Picard iteration $i-1$, the whole process can be summarized in three sequential steps to obtain the states at the iteration $i$. The first one collects the evaluations of the dynamics function $\mathbf{f}$ in the $N \times n$ force matrix $\mathbf{F}$ [44]:

$$\mathbf{F}_{j+1}^{(i)} = \omega_2 \ \mathbf{f}\left(\mathbf{y}_j^{(i-1)}, t_j\right), \quad j = 0, \dots, N-1 \tag{12}$$

Secondly, identifying with $\mathbf{A}$, $\mathbf{C}$, $\mathbf{S}$ the method's constant matrices whose definition can be found in [44], the $N \times n$ matrix $\mathbf{B}$ is obtained by rows as [44]

$$\mathbf{B}_1 = \mathbf{SAF} + 2\mathbf{y}_0, \quad \mathbf{B}_j = \mathbf{AF}, \quad j = 2, \dots, N \tag{13}$$

Third and last, the $N \times n$ matrix of the state guesses $\mathbf{y}^{(i)}$ for the $i$th Picard iteration is

$$\mathbf{y}^{(i)} = \mathbf{CB} \tag{14}$$

The iteration process stops when the maximum state difference between two consecutive Picard iterations $\mathbf{y}^{(i)}$ and $\mathbf{y}^{(i-1)}$ drops below a specified relative or absolute tolerance, upon user's choice.

Despite the proved theoretical convergence, large integration spans may lead to numerical instabilities, due to the cumulation of round-off errors even with large $N$ as multiple orbital revolutions take place [18–20]. Fukushima [18] suggests a piece-wise approach as a workaround, which has been implemented in this work and uses the modified Picard–Chebyshev method to integrate orbit by orbit in sequence[4] until the end of the span.

The core steps of the proposed algorithm follow the presented scheme [19,20], together with the automatic generation of the Keplerian initial guess spanning one nominal orbital period.

## 4. Continuous multi-flyby design in the relativistic N-body problem

Significant trajectory deflections can be achieved using flybys, however such an amplifying effect requires a high precision measure of the entrance state to the planet's sphere of influence. In fact, it is well known that even small errors on the entrance conditions can lead to completely undesired exit states, which might be disastrous for the

[4] The proposed implementation automatically handles either forward or backward integration.

forthcoming mission phases. This issue can be mitigated increasing the precision of the models used to simulate the trajectories, nonetheless the high computational complexity of some perturbation effects hinders their practical use for the mission analysis. Among those, other than their computational burden, complex gravitational fields generated by the N-body environment build an overall chaotic dynamical system. This makes it extremely difficult to search for solutions similar to each other, since such systems are characterized by diverging trajectories even for small differences on the initial conditions. This work introduces an efficient computational framework to account for such perturbing effects, taking also advantage of the chaotic force environment to minimize the artificial trajectory correction maneuvers. Being the goal the development of the design technique itself to exploit chaotic perturbations, without focusing on the particular test-case trajectory, solar radiation pressure effects are neglected. Their modeling strategy is well known, as well as its optimal exploitation through the solar sail technology. It is true that even without a sail its magnitude may be higher than some of the effects of the N-bodies, nonetheless their inclusion in the dynamical model would not be a conceptual novelty, and moreover it would not change the computational setup proposed in this work. On top of the Newtonian gravitational effects, general relativity contributions are included as well, to highlight that even perturbations with the most complex physics can be exploited by the proposed setup. General relativity effects have been previously implemented by the authors of this work [33,39], based on the post-Newtonian model of the Einstein–Infeld–Hoffmann equations as presented by Seidelmann [45]. The same set of equations is used by the Jet Propulsion Laboratory (JPL) for their simulations generating ephemerides data [46], which are also used in this work to fetch the state of the N-bodies at each sampling time $t_j$.

The b-plane theory is used to prune the optimization of a given multi-flyby trajectory. Knowledge of desired macro-properties are assumed to be known, such as semi-major axis, eccentricity, inclination and flyby planets and times, the overall algorithm can be summarized in two steps:

1. Obtaining the unperturbed patched-conics solution using the b-plane theory, as explained in Section 4.1, for the interplanetary orbits and the planetocentric details of all of the possibly multiple flybys.
2. Making the solution continuous in time and space, accounting for perturbing effects and exploiting them to minimize the corrections required to enter subsequent flybys.

The presented steps are explained in more detail in the following sections.

### 4.1. Patched conics b-plane solution for resonant orbits

Valsecchi et al. [16,17] found an analytical solution for the computation of the post-encounter orbital parameters for a given b-plane point at the entrance of the sphere of influence. They successfully identify fixed values of eccentricity and inclination that conserve the Tisserand parameter, for each point belonging to a fixed semi-major axis circle. Although analytical, the relationship is unfortunately given as a full algorithmic procedure made of highly non-linear equations: this makes it difficult to build the inverse relation, i.e. to retrieve the b-plane entrance to the sphere of influence given the full set of post-encounter orbital parameters, even in a numerical or optimization context as convexity cannot be in general ensured.

An alternative approach was developed in a previous work [33], defining an efficient optimization problem that uses the spherical geometry relations that generate the b-plane circles.

Specifically mentioning to the case of resonances, another optimization layer was developed [33]: find a set of intermediate resonant trajectories to gradually move from an initial interplanetary orbit to a final one, which is not reachable with a single flyby, for a fixed number of intermediate flybys. A set of intermediate tentative $\Delta\mathbf{v}$ targets is defined, which the algorithm tries to match while preserving the resonance condition. A block-scheme diagram of the unperturbed design algorithm is given in Fig. 3.

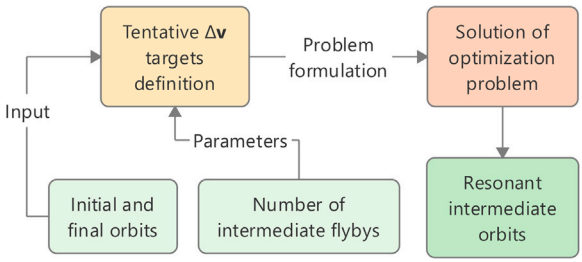



**Fig. 3.** Block-scheme diagram of the unperturbed design algorithm developed in [33].

In the unperturbed and patched conics context, 1–2 s only [33] are required by a MATLAB® implementation of this approach to design a set of resonant orbits with Venus, which are already very close to the actual optimized mission profile from Solar Orbiter's mission redbook [12].

As shown in Section 2.2, when accounting for perturbing effects in the b-plane, a modification of the circles is inevitably introduced, as already shown in Figs. 2(a) and 2(b) Nevertheless, the $\Delta\mathbf{v}$ variation due to perturbing effects is much smaller than the difference between the two set of circles, in relative terms [33].

### 4.2. Integration method adaptation and precision assessment

The SPICE toolkit [46] is used together with JPL's ephemerides data to retrieve the states of the $N$ bodies at any integration step, required for the computation of both the Newtonian and the relativistic perturbations due to the Solar System bodies. This aspect has been noted to be the most computationally expensive task in the general integration accounting for N-body effects, for instance making around 60% of the total account in the work by Colombo et al. [38]. In fact, a binary source must be scanned seeking for the closest saved samples, which must then be interpolated to fit the actual supplied time, for each step and for each of the bodies in the integration. Time steps cannot be foreseen with the standard integration methods, that continuously adapt the step size and sequentially move forward or backward from a given state, thus requiring repeated toolkit calls.

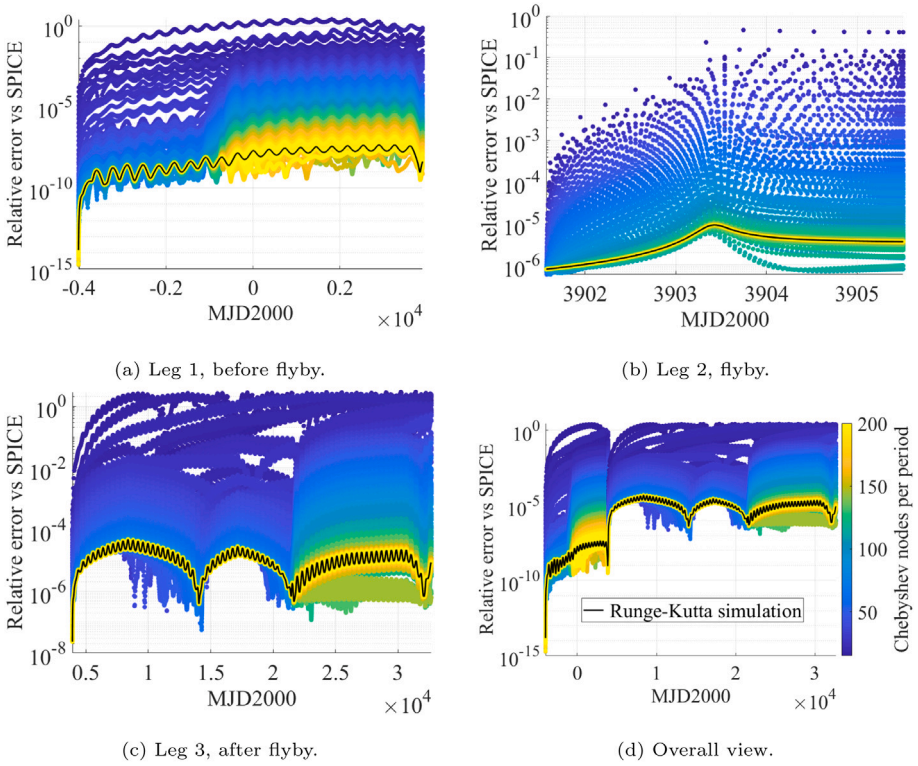
The fixed point nature of the modified Picard–Chebyshev method brings a significant advantage to this regard: consider the restricted N-body problem equation for a test particle written in barycentric Cartesian coordinates:

$$\ddot{\mathbf{r}}(t) = -\sum_{i=1}^{N} \frac{\mu_i(\mathbf{r}(t) - \mathbf{r}_i(t))}{|\mathbf{r}(t) - \mathbf{r}_i(t)|^3} \tag{15}$$

with $\mathbf{r}(t)$, $\dot{\mathbf{r}}(t)$ and $\ddot{\mathbf{r}}(t)$ position, velocity and acceleration vectors respectively.

If the time $t$ is used as the independent variable to integrate the motion of the test particle with the modified Picard–Chebyshev method, it must be sampled a-priori on the Chebyshev nodes, by the definition of the method itself. Using a dataset for the ephemerides instead of requiring a custom integration of the full N-body problem makes $\mathbf{r}_i$ sole function of the time $t$. In turn, it is possible to sample also the states of the $N$ bodies a-priori, as the sampling times are never going to change through the whole integration process, and such samples can be then given as input not only to the dynamics function evaluation, but become a parameter for all the required Picard iterations. This aspect can dramatically speed up numerical simulations in the interplanetary environment, provided that the precision achieved is satisfactory.

This section aims indeed at discussing the accuracy of the method. Note that the inclusion of general relativity effects does not add any




(a) Leg 1, before flyby.

(b) Leg 2, flyby.

(c) Leg 3, after flyby.

(d) Overall view.

**Fig. 4.** Picard–Chebyshev and Runge–Kutta RK78 integration errors, as relative position difference with respect to JPL's data for the asteroid $2010RF_{12}$, for the pre-flyby 4(a), flyby 4(b) and post-flyby legs 4(c), as well as globally for the whole integration span 4(d). The color scale is the same for all the sub-figures, and reported in 4(d). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

conceptual complexity to the problem. Its dynamics can be simply summarized as $\ddot{\mathbf{r}}(t) = \mathbf{f}(\mathbf{r}_i(t), \dot{\mathbf{r}}_i(t))\ i = 1, \dots, N$, thus also the velocity samples for all the $N$ bodies are required to keep the ephemerides data as an integration parameter. Furthermore the integration is performed piece-wise orbit-by-orbit, as suggested by Fukushima [18]: new time nodes are generated, thus new ephemerides are sampled, one orbital period by one orbital period until the end of the time span is reached, or only once for the time spent within the sphere of influence in case of flyby phases.

Figs. 4(a), 4(b), 4(c), 4(d) show the evolution of the relative position error with respect to JPL's data for the near-Earth asteroid $2010RF_{12}$ from 1st January 1989, 100 years forward in time, integrating in the Sun-centered J2000 reference frame and varying the number of Chebyshev nodes per orbit from 15 to 200. Such asteroid was chosen because it performs a flyby of Earth, so that the hyperbolic phase could be tested too. The three different legs (pre-flyby in Fig. 4(a), flyby in Fig. 4(b), post-flyby in 4(c)) are shown on their own, as well as the overall global view is given in Fig. 4(d). The color scale is the same for all the sub-figures, and reported in 4(d). For each leg, the initial guess is the Keplerian solution, elliptical or hyperbolic depending on the current status, generated from the initial state (entrance to the sphere of influence in case of planetary flybys). The flyby time is known, so three legs in total are shown.[5] It can be clearly seen that the integration accuracy increases with increasing number of nodes per orbit, depicted with the same color scale in all the sub-plots, and converges to the precision of a more traditional simulation strategy plotted with the black solid line. The latter was performed using the Runge–Kutta RK78 method, the same dynamical model was adopted in both cases.[6]

Fig. 5 presents instead the relative relationship between the execution time of the sequential modified Picard–Chebyshev method and the number of Chebyshev nodes. A detailed analysis of the absolute computational performances, including scalability, speedup properties of a parallelized version and implementation language influence, will be included in future works upon completion of the whole integration strategy. To provide a first order of magnitude, a MATLAB® non-parallel implementation with a MEX® function for the dynamics[7] requires about 15 s to complete the full integration presented in Fig. 4(d), on a single core of a local workstation equipped with an Intel® Core™ i7-7700 CPU (3.60 GHz).

### 4.3. B-plane pruned recursive refinement: reverse cascade flyby design

A recursive strategy for multi-flyby design can be built wrapping up the concepts presented so far. The whole multi-flyby problem is broken down in a discrete set of orbital arcs, each being covered between two gravity assist maneuvers. When placed in a backward design, the proposed algorithm tries to give an optimal solution to the problem: *how should flyby $j$ occur, so that flyby $j+1$ happens according to some already specified features and accounting for any perturbing effect?*

The b-plane design strategy [33] provides a unique entrance (and thus exit) to the sphere of influence in the patched conics approximation. All that remains to do, conceptually, is to properly provide the interface conditions between the two legs, accounting for all the possible perturbations sources and replacing the zero/infinity link with a continuity relationship. In the following lines the subscripts *in* and *out* shall denote the specific points of entrance and exit to/from the sphere of influence for the current flyby.

[5] The flyby detection routine is necessary to use the integrator as a whole, nevertheless for the design purposes of this work all the legs and times are known beforehand.

[6] The test case was extensively discussed by Masat [33] for the validation of the implementation of relativistic effects.

[7] The MEX® function was generated with MATLAB®'s Code Generation Toolbox.

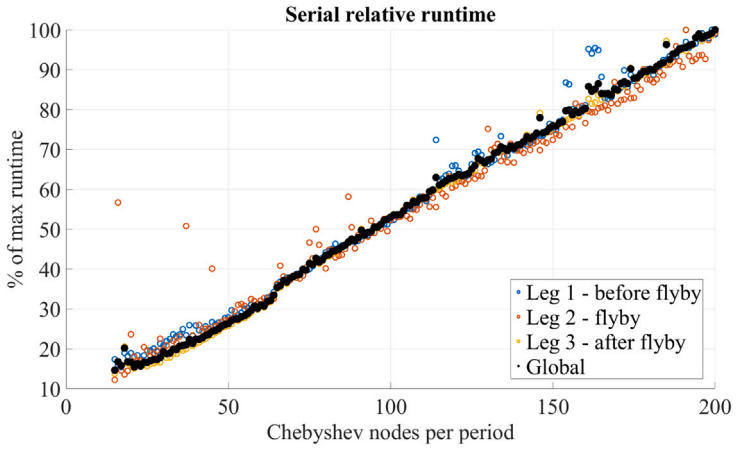



**Fig. 5.** Picard–Chebyshev serial execution runtimes for the asteroid $2010RF_{12}$, reported as relative runtime with respect to the maximum runtime obtained for the different number of Chebyshev nodes.

Consider the entrance conditions to flyby $j+1$, happening at the time $t_{in}^{(j+1)}$, as the Sun-centric position $\mathbf{r}_{in}^{(j+1)}$ and velocity $\mathbf{v}_{in}^{(j+1)}$, already fulfilling the mission requirements for $t > t_{in}^{(j+1)}$ together with possible future maneuvers already defined. Consider also a deep space correction maneuver happening at the time $\tilde{t} > t_{out}^{(j)}$. The whole entrance condition $\left(t_{in}^{(j+1)}, \mathbf{r}_{in}^{(j+1)}, \mathbf{v}_{in}^{(j+1)}\right)$ is back-integrated in the perturbed environment with the modified Picard–Chebyshev method to the time $\tilde{t} < t_{in}^{(j+1)}$, obtaining the connection state $(\tilde{\mathbf{r}}, \tilde{\mathbf{v}})$.

Assume that the unperturbed solution for flyby $j$ is expressed in the b-plane formalism, which can also mean a manipulation of the solution of the Lambert problem [34] with the related planetocentric phase and not necessarily from the already mentioned b-plane algorithm [33], particularly as:

- the outgoing time $t^{(j)}$;
- the outgoing b-plane coordinates $(\xi, \zeta)$;
- the outgoing planetocentric asymptotic velocity $\mathbf{U}'$.

Based on this, the time spent in the flyby phase $\delta t^{(j)}$ can be estimated with the time law for the hyperbolic motion,[8] forcing the remaining b-plane coordinate $\eta$ such that the distance from the flyby planet equals the radius of the sphere of influence. In turn, $\delta t^{(j)}$ can be used to get another estimate, that is the actual exit from the sphere of influence $t_{out}^{(j)} = t^{(j)} + \delta t^{(j)}/2$. The time $t_{out}^{(j)}$ is actually the outer optimization variable of the proposed algorithm. Intuitively, the time estimate arising from $t^{(j)}$ and $\delta t^{(j)}$ might not be the best possible time when to abandon the sphere of influence starting the phase toward flyby $j+1$ and performing the minimum cost correction maneuver at $\tilde{t}$, especially because of perturbing effects acting on the way. The claim that is made treats $t_{out}^{(j)}$ as a very good starting guess for an outer optimization layer, using a "perturbation" $\Delta t^{(j)}$ of the exit time as optimization variable and bounding the search to a relatively small domain. A similar reasoning is made for the b-plane coordinates $(\xi, \zeta)$ and the outgoing planetocentric velocity $\mathbf{U}'$, considering the unperturbed solution as initial optimization guess and searching over small variations thereof. Theoretical support comes in this case from the results of the perturbed b-plane circles: the relatively small difference between the selected points in the perturbed and unperturbed cases suggests to use the variations of the b-plane coordinates $(\Delta\xi, \Delta\zeta)$ as two optimization variables and to bound them again in a relatively small search space. The set of optimization variables is completed with $\Delta\mathbf{U}'$, a variation of $\mathbf{U}'$ bounded in a small domain as well. Note that the usage of the b-plane interface between flyby and interplanetary leg combined with the small bounded variation approach has also a more practical reason: despite working in a backward time recursion, a perturbed trajectory that minimizes the maneuver cost at $\tilde{t}$ may in general excessively differ from the mission requirements. The b-plane intrinsically constrains the interface to be an actual flyby, furthermore the small and bounded search space should ensure a perturbed trajectory not too different from the desired profile for $t < t_{in}^{(j)}$. Given the initial values $\left(t_{out}^{(j)}, \xi, \zeta, \mathbf{U}'\right)$ and the generic variations $\left(\Delta t^{(j)}, \Delta\xi, \Delta\zeta, \Delta\mathbf{U}'\right)$, the initial conditions $\left(\mathbf{r}_{out}^{(j)}, \mathbf{v}_{out}^{(j)}\right)$ at the time $t_{\Delta}^{(j)} = t_{out}^{(j)} + \Delta t^{(j)}$ for the forward Picard–Chebyshev integration from $t_{\Delta}^{(j)}$ to $\tilde{t}$ are uniquely defined through the following steps:

1. the flyby planet's state $\left(\mathbf{r}_p^{(j)}, \mathbf{v}_p^{(j)}\right)$ can be retrieved by reading the ephem-erides database for the time $t_{\Delta}^{(j)}$;
2. from $(\xi + \Delta\xi, \zeta + \Delta\zeta)$ the third b-plane coordinate $\eta$ is fixed by requiring the distance from the planet to equal the radius of the sphere of influence;
3. the b-plane coordinates $(\xi + \Delta\xi, \eta, \zeta + \Delta\zeta)$ can be converted into the planetocentric cartesian coordinates $\mathbf{r}_{pl}$, because the axes of the b-plane reference frame are uniquely defined as in Eq. (1) and the planetocentric velocity vector is $\mathbf{U}' + \Delta\mathbf{U}'$;
4. The Sun-centric coordinates $\left(\mathbf{r}_{out}^{(j)}, \mathbf{v}_{out}^{(j)}\right)$ are retrieved by the simple summations $\mathbf{r}_{out}^{(j)} = \mathbf{r}_{pl} + \mathbf{r}_p^{(j)}$ and $\mathbf{v}_{out}^{(j)} = (\mathbf{U}' + \Delta\mathbf{U}') + \mathbf{v}_p^{(j)}$.

The initial value problem identified by $\left(\mathbf{r}_{out}^{(j)}, \mathbf{v}_{out}^{(j)}\right)$ at the time $t_{\Delta}^{(j)} = t_{out}^{(j)} + \Delta t^{(j)}$ is solved numerically forward in time with the modified Picard–Chebyshev method, to the connection maneuver at an arbitrary time $\tilde{t}$. Using a concise notation, the initial value problem to be numerically integrated will be identified by the dynamics functions $\mathbf{r}_f(t)$ and $\mathbf{v}_f(t)$, with $t_{out}^{(j)} \le t \le \tilde{t}$, for position and velocity respectively, and setting the initial conditions:

$$t_0 = t_{out}^{(j)}, \quad \mathbf{r}(t_0) = \mathbf{r}_{out}^{(j)}, \quad \mathbf{v}(t_0) = \mathbf{v}_{out}^{(j)} \tag{16}$$

In general, the forward-integrated state $\left(\mathbf{r}_f(\tilde{t}), \mathbf{v}_f(\tilde{t})\right)$ at the correction maneuver time $\tilde{t}$ will differ from the back-integrated state that leads to flyby $j+1$ by

$$\widetilde{\Delta\mathbf{r}} = \mathbf{r}_f(\tilde{t}) - \tilde{\mathbf{r}} \neq \mathbf{0} \quad \text{and} \quad \widetilde{\Delta\mathbf{v}} = \mathbf{v}_f(\tilde{t}) - \tilde{\mathbf{v}} \neq \mathbf{0} \tag{17}$$

The physics of the correction maneuver performed at the maneuver time $\tilde{t}$ embeds the mandatory constraint of the position where it is to happen, theoretically defined as $\widetilde{\Delta\mathbf{r}} = \mathbf{0}$. Note that the dynamical motion is numerically integrated, thus leaving the maneuver position as a pure equality constraint might severely affect the computational performance of the optimization: a full Picard–Chebyshev integration would be required to evaluate the constraint function, since the optimization variables are nothing but the b-plane form of the initial state $\left(\mathbf{r}_{out}^{(j)}, \mathbf{v}_{out}^{(j)}\right)$ and their simulation to the connection time $\tilde{t}$ would always

[8] Not reported here. See for instance Vallado [34] for more details.

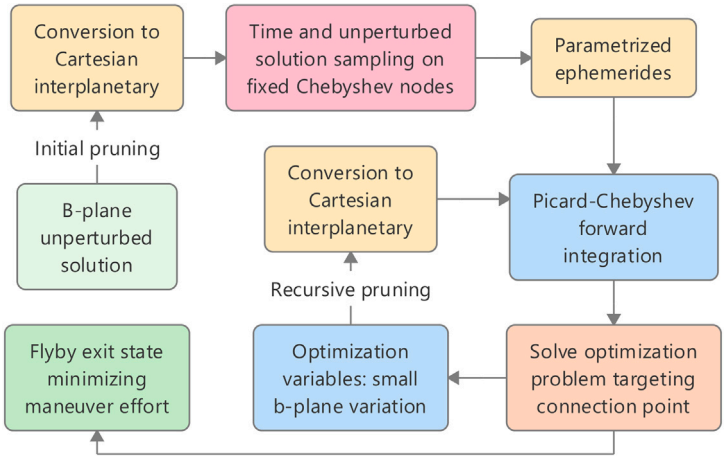



**Fig. 6.** Block-scheme diagram of features and steps embedded in the solution of the optimization problem of Eq. (18).

be needed. At the opposite side, the actual maneuver to be designed may not have any physical sense if omitted, as the continuity requirement may be lost. Nonetheless, in a numerical context an absolutely negligible value of $\widetilde{\Delta\mathbf{r}}$ suffices to satisfy the physical meaning of the correction maneuver. These observations led to the choice of explicitly implementing the position constraint with a penalty method [47], that is penalizing the objective function (the correction $|\widetilde{\Delta\mathbf{v}}|$ in this case) adding a large term direct function of the position difference $\widetilde{\Delta\mathbf{r}}$.

Therefore, defining $J_v = \left|\widetilde{\Delta\mathbf{v}}\right|$, omitting the explicit dependencies on the optimization variables for conciseness and denoting the components of $\Delta\mathbf{U}'$ with $\Delta U'^{(1,2,3)}$

$$\begin{aligned} \underset{\Delta\xi,\Delta\zeta,\Delta\mathbf{U}'}{\text{minimize}} \quad & J_v\left(\tilde{t}, t_\Delta^{(j)}\right) + \alpha J_r\left(\tilde{t}, t_\Delta^{(j)}\right) \\ \text{subject to} \quad & |\Delta\xi| \leq \Delta\xi_{max}, \\ & |\Delta\zeta| \leq \Delta\zeta_{max}, \\ & |\Delta U'^{(1,2,3)}| \leq \Delta U'^{(1,2,3)}_{max} \end{aligned} \tag{18}$$

with $J_r = \left|\widetilde{\Delta\mathbf{r}}\right|$ and the weighting factor $\alpha$ of the penalty method sufficiently large. The choice of $\alpha$ is in general arbitrary, it will be discussed in Section 5 for the presented test case.

A block-scheme diagram summarizing all the presented features and steps of the algorithmic optimization problem in Eq. (18) is given in Fig. 6. The initial unperturbed solution expressed in the b-plane formalism is converted to a Cartesian state and used to prune the optimization process. Subsequently, the fixed-point nature of the Picard–Chebyshev nature requires to sample an initial trajectory guess on fixed time nodes: as already mentioned, this feature is also exploited to perform the sampling of the N-bodies ephemerides only once, not only for the Picard–Chebyshev integration but also for the whole optimization run. Finally, the "closed loop" that can be seen in Fig. 6 is entered, where each objective function evaluation involves the Picard–Chebyshev forward integration of some coordinates, generated from the optimization variables expressed as b-plane variations. The exit conditions strongly depend on the chosen implementation, although they can follow any already existing scheme (for instance, relative state and objective function variations smaller than some user-defined tolerance in this work, as it is discussed in Section 5.2).

Finally, assuming the position constraint to be fulfilled in the optimization problem of Eq. (18) whose result gives $J^*\left(\tilde{t}, t_\Delta^{(j)}\right) = J_v^*\left(\tilde{t}, t_\Delta^{(j)}\right) + \alpha J_r^*\left(\tilde{t}, t_\Delta^{(j)}\right)$, for optimizing the flyby time it is enough to use its definition $t_\Delta^{(j)} = t_{out}^{(j)} + \Delta t^{(j)}$, with $\Delta t^{(j)}$

$$\begin{aligned} \underset{\Delta t^{(j)}}{\text{minimize}} \quad & J^*\left(\tilde{t}, t_{out}^{(j)} + \Delta t^{(j)}\right) \\ \text{subject to} \quad & |\Delta t^{(j)}| \leq \Delta t_{max} \end{aligned} \tag{19}$$

the new optimization variable:

**Table 1**
Optimization levels for the full design of the arc $j$ to $j+1$.

| | | |
|---|---|---|
| $J_v + \alpha J_r$ (Eq. (18)) | $\xrightarrow[\Delta\xi,\Delta\zeta,\Delta U']{\text{Optimize}}$ | $J^*$ (Eq. (19)) |
| $J^*$ (Eq. (19)) | $\xrightarrow[\Delta t^{(j)}]{\text{Optimize}}$ | $j$ to $j+1$, maneuver at $\tilde{t}$ |
| $j$ to $j+1$, maneuver at $\tilde{t}$ | $\xrightarrow[\tilde{t}]{\text{Optimize}}$ | $j$ to $j+1$, optimal |

Note that no choice has been made yet about the optimization algorithms, which might be sensitive to the search space size and the function relative steepness within the different regions. Moreover, it should also be tailored on the available computational resources, i.e. preferring parallelizable routines over dominantly sequential algorithms for high performance computing facilities.

The optimization problem of Eq. (18) is a sub-problem of the optimization problem of Eq. (19). This somehow enhances the flexibility of the approach, i.e. the former might be used for search space exploration purposes without the need of a finely refined solution in terms of starting time $t_\Delta^{(j)}$. Note that the optimization problem of Eq. (18) is explicitly dependent on the maneuvering point $\tilde{t}$, which in fact can and for practical applications should be optimized as well. In this work it shall remain a problem parameter, as more focus is put toward exploring the effect of small variations of the departure time $t_\Delta^{(j)}$. Completing the description, another more optimization level can be easily defined to find the best $\tilde{t}$ similarly to what done for $t_\Delta^{(j)}$ in the optimization problem of Eq. (19), and in the presented formalism it shall straightforwardly include the innermost level defined by the optimization problem of Eq. (18). A summary of the relationship among the different optimization levels is given in Table 1.

Some observations regarding the expected computational performances of the optimization can be made based on the analysis of the Picard–Chebyshev integration, already presented in Figs. 4(a), 4(b), 4(c), 4(d) for the accuracy behavior and Fig. 5 for the computational time variation with increasing number of Chebyshev nodes. For sequential executions, the higher the value of $\tilde{t}$ the higher the runtime will be, if the number of nodes per period is kept constant. Note also that the optimization problem of Eq. (18) is going to benefit from the minimal ephemerides overhead as a whole: the boundary times are fixed, thus the ephemerides dataset can be scanned only once and the related values can be considered as parameters not only within the Picard–Chebyshev integration, but also for all the iterations of the optimization algorithm. Finally, the penalty approach [47] used to define the optimization problem of Eq. (18) allows for massively parallel and brute force strategies to be implemented as well, because all the remaining constraints are of boundary type.

**Table 2**
Maneuvering point, apocenter of unperturbed initial pruning solution.

| $\mathbf{r}_{in}^{(j+1)}$ [km] | | | $\mathbf{v}_{in}^{(j+1)}$ [km/s] | | |
|---|---|---|---|---|---|
| $r_x$ | $r_y$ | $r_z$ | $v_x$ | $v_y$ | $v_z$ |
| −67030683.03 | −85738232.37 | 2563856.42 | 30.54 | −4.05 | 1.79 |

## 5. Application: Solar Orbiter's first resonant phase with venus

The presented design procedure was tested on a phase of the ongoing mission Solar Orbiter [11], particularly taking the initial data from the trajectory profile with launch in January 2017[9] available in the mission redbook [12]. For this concept validation phase, the algorithm has been entirely implemented in MATLAB®. A small computational acceleration is introduced compiling the Picard–Chebyshev iterations into a MEX® function with MATLAB® Coder™. The optimization problem of Eq. (18) is solved with the `fmincon.m` function of MATLAB®'s Optimization Toolbox, using both the Interior-point and Sequential Quadratic Programming methods [47], based on the dimension of the search space. The selected local optimization algorithms and tolerances sufficed for the test case of this work to converge, proving the methodology concept of efficiently designing trajectories that take advantage of chaotic perturbations. The use of global search approaches and/or different tolerance setups could for sure increase the robustness of the approach, at the price of possibly increasing the total computational load. In the case of practical use by mission analysts, the optimization algorithm selection should also be tailored on the hardware availability, possibly exploiting the full potential of supercomputing facilities.

Solar Orbiter's first resonant phase with Venus is reproduced accounting for perturbing effects from the $N$ bodies and general relativity. Following the notation from the mission redbook [12], the two gravity assist maneuvers are identified with V2 and V3, with V standing for the flyby planet (Venus) and the numbers 2 and 3 representing the second and third close approach with Venus from the mission launch, respectively. The interplanetary leg between the two flybys is identified with V2–V3. The goal is to design flyby V2 so that V3 can lead to a desired post-encounter trajectory almost ballistically, i.e. minimizing the correction maneuver required in the phase between V2 and V3. The maneuver is designed with maneuvering time $\tilde{t}$ at the apocenter of the first nominal orbit after V2, nevertheless as already mentioned even this aspect can and should be optimized.

### 5.1. Boundary conditions, b-plane pruning and method parameters

Generally, the required boundary condition is the state vector that allows a specified entrance to flyby $j+1$. It may come from a previous step of the presented flyby design algorithm, as the output of the back-integration of $\left(\mathbf{r}_{out}^{(j+1)}, \mathbf{v}_{out}^{(j+1)}\right)$, or simply being given, if no close approach is to happen after flyby $j+1$. Considering the Solar Orbiter-like mission, flyby V3 may be entered as the interplanetary state written in the ecliptic J2000 reference frame reported in Table 2, at the time $t_{in}^{(j+1)} = 8119.84$ MJD2000.

Solar Orbiter's first resonant phase with Venus is in a 3/4 resonance, which means that in the unperturbed and patched conics case flyby V2 is to happen 3 Venus' periods before flyby V3, and in the meantime Solar Orbiter would have traveled for 4 of its orbital periods. The output of the b-plane preliminary unperturbed design [33] enforcing the 3/4 resonance has produced the pruning quantities reported in Table 3, together with the exit time from flyby $j$ set as $t_{out}^{(j)} = 7446.52$ MJD2000.

[9] Later discarded, the actual mission left Earth on February 2020.

**Table 3**
Retrieved optimal b-plane coordinates $(\xi^*, \zeta^*)$ and planetocentric velocity $\mathbf{U}^*$.

| $\xi$ [km] | $\zeta$ [km] | $U_x$ [km/s] | $U_y$ [km/s] | $U_z$ [km/s] |
|---|---|---|---|---|
| −8057.07 | −5497.19 | 3.08 | 17.78 | 3.66 |

**Table 4**
Maneuvering point, apocenter of unperturbed initial pruning solution.

| $\tilde{\mathbf{r}}$ [km] | | | $\tilde{\mathbf{v}}$ [km/s] | | |
|---|---|---|---|---|---|
| $r_x$ | $r_y$ | $r_z$ | $v_x$ | $v_y$ | $v_z$ |
| −133524954.60 | −32036518.08 | −4418791.75 | 5.09 | −20.43 | 1.65 |

The maneuvering time is set as a parameter, particularly at the nominal[10] apocenter of the first interplanetary resonant orbit, at the time $\tilde{t} = 7570.92$ MJD2000, with the correspondent state $(\tilde{\mathbf{r}}, \tilde{\mathbf{v}})$ reported in Table 4.

The maximum values where to bound $\left(\Delta\xi_{max}, \Delta\zeta_{max}, \Delta U'^{(1,2,3)}_{max}\right)$ have been set as 1% of the impact parameter [36] $b = \sqrt{\xi^2 + \zeta^2}$ and of $|\mathbf{U}'|$ for the b-plane coordinates and the velocity components respectively. The boundary value for the exit time variation $\Delta t_{max}$ is set to 1% of Venus' orbital period. Trivially, the optimization starts with all the variables $\left(\Delta\xi, \Delta\zeta, \Delta\mathbf{U}', \Delta t^{(j)}\right)$ set equal to zero. The cost functions $J_r$ and $J_v$ are in all the cases computed as the relative values $|\Delta\mathbf{v}|/|\tilde{\mathbf{v}}|$ and $|\Delta\mathbf{r}|/|\tilde{\mathbf{r}}|$ with respect to the known maneuvering point, to remove the possible dimension sensitivity.

Specifically for the modified Picard–Chebyshev method, 160 nodes per period are used and the iterations are stopped when the maximum of the relative difference between two consecutive state updates drops below $10^{-14}$. The first arc to be designed, i.e. the one defining the optimal exit and the maneuver, spans less than one orbital period, thus proportional nodes to the defined 160 per period based on its total time length are set, according to the fixed nodes per period logic. The optimal time found is then used for a single run of the optimization problem of Eq. (18) with 200 Chebyshev nodes, assessing the influence of the number of nodes in the design precision, comparing both the node cases against a relativistic simulation.

### 5.2. Optimization implementation

Despite the narrow region where the optimization variables are set to vary, it has been observed that even the smallest variations have a relevant impact in the convergence of the algorithm, especially if the position constraint is made strict. For this reason and to keep a robust approach in the concept validation phase, the optimization problem of Eq. (18) is solved several times in a continuation procedure, using the result of the previous step as the new starting guess. Particularly:

- the search space dimension is reduced by 10 times for each optimization problem, up to an absolute minimum of $10^{-8}$ starting from the already introduced ± 1% for each variable;
- within the optimization solver, the initial minimum relative step size between two iterations is of $10^{-6}$, reduced by a factor 10 each time up to $10^{-15}$;
- the penalty factor $\alpha$ is initially set to $10^5$ to improve the convergence also for the $J_v$ contribution, although the position constraint is then made stricter by raising the value of $\alpha$ by a factor 10 each time, up to $10^9$;
- the Interior-point algorithm in `fmincon.m` is selected for the first half optimization problems, whereas Sequential Quadratic Programming is used in the last ones because of the smaller search space;

[10] It is set as if the orbital parameters were exactly equal to the desired trajectory after the maneuver, assuming the difference between pre and post maneuver orbits to be small.

**Table 5**
Optimization results, in terms of position difference residual $\Delta\mathbf{r}^*$ and correction effort $\Delta\mathbf{v}^*$ at the maneuvering time.

| $\Delta\mathbf{r}^*$ [m] | | | $\Delta\mathbf{v}^*$ [m/s] | | |
|---|---|---|---|---|---|
| $r_x$ | $r_y$ | $r_z$ | $v_x$ | $v_y$ | $v_z$ |
| −0.52 | −0.52 | −1.19 | −1.28 | 1.57 | 0.22 |

**Table 6**
Optimization results, in terms of initial position residual and $\Delta\mathbf{v}$ magnitude.

| $\lvert\Delta\mathbf{v}^*\rvert$ [m/s] | $\lvert\Delta\mathbf{v}^*\rvert/\lvert\tilde{\mathbf{v}}\rvert$ [–] | $\lvert\Delta\mathbf{r}^*\rvert$ [m] | $\lvert\Delta\mathbf{r}^*\rvert/\lvert\tilde{\mathbf{r}}\rvert$ [–] |
|---|---|---|---|
| 2.04 | $9.66\times10^{-5}$ | 1.39 | $1.02\times10^{-11}$ |

**Table 7**
Optimization results, in terms of initial optimal state $\left(\mathbf{r}_{out}^{(j)*}, \mathbf{v}_{out}^{(j)*}\right)$.

| $\mathbf{r}_{out}^{(j)*}$ [km] | | | $\mathbf{v}_{out}^{(j)*}$ [km/s] | | |
|---|---|---|---|---|---|
| $r_x$ | $r_y$ | $r_z$ | $v_x$ | $v_y$ | $v_z$ |
| −64960957.28 | −85998225.22 | 2682290.24 | 31.00 | −3.45 | 1.7 |

- MATLAB®'s `globalsearch` algorithm solves the current optimization problem if the previous step has returned the starting guess without improvements, searching for a global minimizer.[11]

The optimization problem of Eq. (19) is solved with a grid search approach. The time span is always sampled with the initial supplied value plus 40 evenly spaced values of $\Delta t^{(j)}$, reducing $\Delta t_{max}$ by a factor 10 for 5 times, from the initial grid size equal to ± 1% of Venus' orbital period. The best value from the previous search is used as starting point for the new one. This approach resembles the algorithm used in MATLAB®'s `patternsearch.m` function, implemented manually in this work to keep a low number of trial $\Delta t^{(j)}$ in this concept validation phase.

### 5.3. Results

Solving the optimization problem of Eq. (18) with the above described implementation took about 2–3 min on a single core of a local workstation equipped with an Intel® Core™ i7-7700 CPU (3.60 GHz). The 200 nodes algorithm[12] converged to the residual $\Delta\mathbf{r}^*$ and impulsive action $\Delta\mathbf{v}^*$ for the required maneuver presented in Table 5.

One should note both how small the correction effort is, despite the execution point $\tilde{t}$ is yet to be optimized, and the fulfillment of the position constraint. The presented maneuver is modeled as a single impulse, nevertheless given its magnitude it can be easily achieved by the current low thrust propulsion technologies, as shown in Table 6

Most of the computational time was required to fulfill the position constraints. If the presented algorithm was to be used with a wider but still good position tolerance, it is likely to run significantly faster even before its parallel implementation.

The best starting time obtained was $t_{out}^{(j)*} = 7446.52$ MJD2000, slightly higher than the initial guess $t_{out}^{(j)}$. The interplanetary optimal starting state the initial condition is given in Table 7.

The b-plane coordinates $(\xi^*, \zeta^*)$ and the planetocentric velocity $\mathbf{U}'^*$ retrieved from $\left(\mathbf{r}_{out}^{(j)*}, \mathbf{v}_{out}^{(j)*}\right)$ and Venus' position at $t_{out}^{(j)*}$ are presented in Table 8, proving the optimal pruning brought by the b-plane prediction.

[11] For performance reasons a maximum of half of the iterations can run the global search, in any case the presented test case at most two were experienced out of all the ten steps.

[12] The difference with the $\Delta\mathbf{v}$ resulting from the 160 nodes run is negligible, the position constraint is slightly worse fulfilled but in the same order of magnitude.

**Table 8**
Retrieved optimal b-plane coordinates $(\xi^*, \zeta^*)$ and planetocentric velocity $\mathbf{U}^*$.

| $\xi^*$ [km] | $\zeta^*$ [km] | $U_x^*$ [km/s] | $U_y^*$ [km/s] | $U_z^*$ [km/s] |
|---|---|---|---|---|
| −8057.22 | −5700.49 | 3.25 | 17.76 | 3.67 |

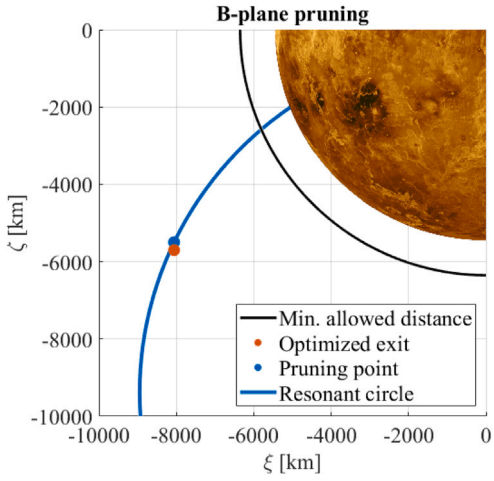



**Fig. 7.** Visual representation of the b-plane pruning strategy. The pruning point corresponds to the blue dot, whereas the optimized point is depicted in dark orange.

The value of $\zeta^*$ looks slightly (2%) out of the initial bounds despite the constraint, which may have two different explanations. First, the optimization variables are updated concurrently: variations on $\mathbf{U}'$ also change the orientation of the b-plane axes, which in turn result on different b-plane coordinates for a given fixed position in space. Secondly, the domain reduction sequential procedure may find a minimum close to the initial boundaries, centering there the next narrower search. The difference is in any case rather small in magnitude, as it can be also seen in the real-scale difference shown in Fig. 7: with the pruning point identified by the blue dot, whereas the optimized one is plotted in dark orange.

Fig. 8 shows the difference between the designed trajectory with respect to a relativistic simulation of the same case, both featuring the optimized maneuver at $\tilde{t}$. It can be observed that the two trajectories basically coincide even if using the lower number of Chebyshev nodes, for a relative difference that remains in the order of $10^{-8}$ and as expected from what already seen in Figs. 4(a), 4(b), 4(c), 4(d). Again as expected the higher number of nodes brought a more accurate solution, with the difference from the relativistic simulation reduced by more than 10 times. Even if small, the error inevitably cumulates and gets amplified if multiple gravity assists are present, thus a higher number of nodes should be kept the more precise the design needs to be. The periodic "hills" visible in Fig. 8 happen far from the domain boundaries, and are located where the Chebyshev nodes become more sparse.[13] In this particular case, they correspond to the neighborhood of the pericenter of the pre-maneuver arc. The periodic error increase likely due to the faster orbital dynamics nearby the pericenter, not followed by the Chebyshev nodes thickness, as the domain boundaries are located at the correction maneuver (at the apocenter). This effect could be mitigated by adapting the Picard–Chebyshev integration intervals so that the node distribution becomes denser nearby the pericenter, for instance splitting the optimization horizon into two sub-intervals, the first from the flyby exit to the pericenter, the second from the pericenter to the connection maneuver point, and finally following the same concept for the fixed post-maneuver arc. If the full trajectory was required with

[13] Because of the definition of Chebyshev nodes in Eqs. (10) and (11).

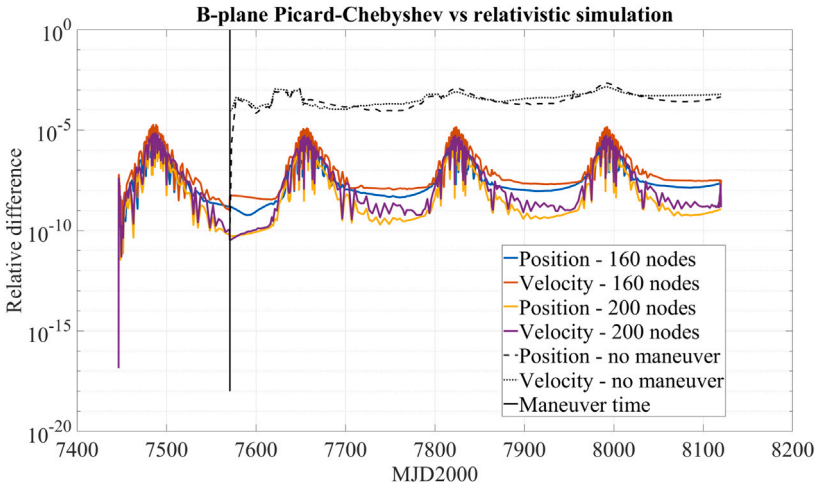
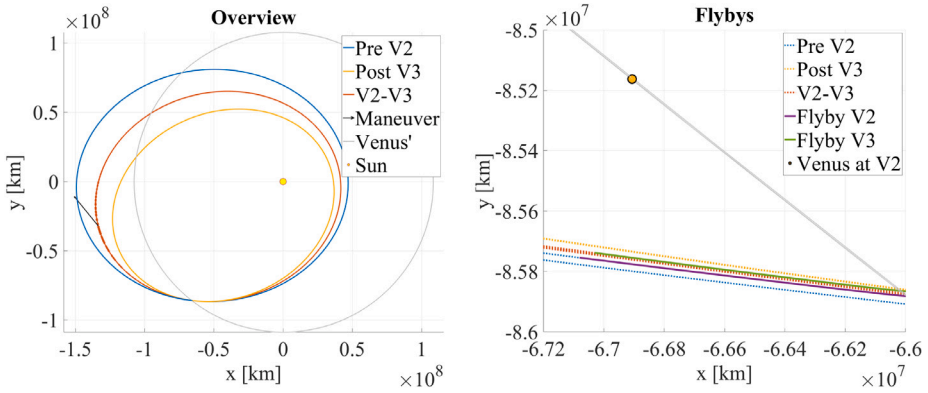



**Fig. 8.** Design difference with respect to relativistic simulation between V2 and V3 for the two node cases, and without maneuver.




(a) Deep space correction maneuver and overview. (b) Zoom over Venus' flybys V2 and V3.

**Fig. 9.** Solar Orbiter's continuous first resonant phase with Venus.

as high precision as possible this should be considered, anyway, even with the tested setup, the long-term error evolution remains low, and already allows for a precise design at the event points (flybys and correction maneuver) compared against standard simulation techniques. The "noise" over those hills is explained by interpolation of the Picard–Chebyshev solution over the standard simulation time steps, necessary to visualize the presented difference measurement. The impact of the correction maneuver, despite small, can also be assessed: at the time of the close approach V3 the position would differ of thousands of kilometers from the desired condition, definitely destroying the correct occurrence of the flyby.

Finally, Figs. 9(a) and 9(b) show the continuous trajectory that embeds the planetocentric phases for both the flybys V2 and V3, together with the pre-V2 and post-V3 solutions and all generated with the Picard–Chebyshev approach. As expected they are all very similar to the original mission profile (Fig. 9(a)), and zooming over the flyby regions the new continuity feature can be recognized (Fig. 9(b)).

## 6. Conclusion

The complexity of the multi-flyby design problem in the continuous environment was successfully broken down into a backward recursive approach that designs each of the flybys in cascade, considering the next encounter as the target condition for how to perform the current one. Given the results of an unperturbed patched conics analysis, the b-plane was proven to be a powerful formalism to enforce a continuity condition with, and particularly well suited for pruning purposes, making the dimension of the optimization search space minimal.

A first possible development direction is the inclusion of tighter mission constraints, such as a minimum pericenter distance as Solar Orbiter [11] needs. This aspect might be tackled with the proposed strategy before the design of any maneuver, seeking for quasi-ballistic solutions that surf the effects of orbital perturbations, even if chaotic, aiming to minimize the required artificial corrections.

Despite it might be already satisfactory, the computational performance of the method is for sure what can be improved the most by future works. First of all, the newest adaptive version of the modified Picard–Chebyshev method can be adopted and the sequential execution can be accelerated by a complete implementation in a compiled programming language, instead of the MATLAB® platform as proposed in this work. Furthermore, although the multi-step solution of the optimization problem of Eq. (18) proposed in the presented application is extremely robust, it could be better tailored by prior analysis of the search space or made more lightweight already scanning with finer tolerances. Most importantly, the whole approach can and is built to be parallelized, both for the modified Picard–Chebyshev method and the solution of the optimization problem.

In conclusion, a systematic framework about how to surf a complex perturbation environment such as the relativistic N-body problem is proposed. Provided the model to be sufficiently accurate, the available technology might become the new bottleneck for practical purposes: some uncertainty is inevitably introduced by the execution of the control maneuvers, as well as the connected orbital determination measurements. Future works might also deepen this aspect using models of real life equipment, studying in turn what consequences non-precise

measurements or thruster firings might have on the high-fidelity designed trajectory, together with the possible required mission planning actions.

Looking to possible applications in other environments, the presented approach may be used in the design of moon tour missions toward the giant planets, which feature the available fuel as a major constraint. Quasi-ballistic solutions are always sought for, to swing by the numerous bodies multiple times maximizing the exploration outcome. For instance the currently being planned JUICE [48] could benefit from this design strategy, since moving from the interplanetary to the jovian system would not require major changes at all.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Funding sources

The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon2020 research and innovation programme as part of project COMPASS (Grant agreement No 679086), www.compass.polimi.it.

## Appendix

The expansions of $\mathbf{f}(\mathbf{y},t)$ and $\mathbf{y}(t)$ in terms of the Chebyshev polynomials of the first kind $T_j(\tau)$ [41] are:

$$\begin{aligned}\mathbf{f}(\mathbf{y},t) &= \sum_{j=0}^{+\infty} \mathbf{F}_j T_j(\tau)\\ \mathbf{y}(t) &= \sum_{j=0}^{+\infty} \mathbf{Y}_j T_j(\tau)\end{aligned} \tag{20}$$

where $T_j(\tau) = \cos(j \arccos \tau)$. Denoting with $t_0$ and $t_f$ initial and final integration times respectively and introducing $H = b - a$ with $a \leq t_0$ and $b \geq t_f$ for generality:

$$\tau = \frac{2t - (a+b)}{H}, \qquad -1 \leq \tau \leq 1 \tag{21}$$

Fukushima [18] presented the method for scalar functions, whereas in this work $\mathbf{F}_j$ and $\mathbf{Y}_j$ are vectors of the same dimension of the state $\mathbf{y}$, representing the coefficients of the $j$th term in the Chebyshev expansion.

Truncating the expansion to the order $N$ for the state and to order $N-1$ for the dynamics function, and highlighting the $i$th Picard iteration yields [18]

$$\begin{aligned}\mathbf{f}^{(i)}(\mathbf{y}^{(i)},t) &= \sum_{j=0}^{N^{(i)}-1} \mathbf{F}_j^{(i)} T_j(\tau)\\ \mathbf{y}^{(i)}(t) &= \sum_{j=0}^{N^{(i)}} \mathbf{Y}_j^{(i)} T_j(\tau)\end{aligned} \tag{22}$$

Note that $N$ is not necessarily fixed for all the iterations. The physical time is discretized by selecting the zeros of the polynomial $T_N$ as the points of evaluation of $\mathbf{f}(\mathbf{y},t)$ and $\mathbf{y}(t)$, to obtain [18,41]:

$$t_k^{(i)} = \frac{a+b}{2} - \frac{H}{2}\cos\left(\frac{(2k-1)\pi}{2N^{(i)}}\right), \quad k = 1,\dots,N^{(i)} \tag{23}$$

Applying the Picard iteration brings the approximation $\mathbf{y}_k^{(i)}$:

$$\mathbf{y}_k^{(i)} = \mathbf{y}^{(i-1)}\left(t_k^{(i)}\right) = \sum_{j=0}^{N^{(i-1)}} c_{jk}^{(i)} \mathbf{Y}_j^{(i-1)} \tag{24}$$

where, since $T_j(\cos\theta) = \cos j\theta$ [41]:

$$c_{jk}^{(i)} = \cos\left(\frac{j(2k-1)\pi}{2N^{(i)}}\right) \tag{25}$$

$\mathbf{f}$ can now be evaluated at the points $\mathbf{y}_k^{(i)}$:

$$\mathbf{f}_k^{(i)} = \mathbf{f}\left(\mathbf{y}_k^{(i)}, t_k^{(i)}\right) \tag{26}$$

After a re-arrangement of the terms and exploiting the orthogonality of the Chebyshev polynomials [41], the following recursions for the coefficients $\mathbf{Y}_j^{(i)}$ are obtained:

$$\begin{aligned}\mathbf{Y}_j^{(i)} &= \frac{H}{jN^{(i)}} \sum_{k=1}^{N^{(i)}} s_{jk}^{(i)} \mathbf{g}_k^{(i)}, \qquad j = 1,\dots,N^{(i)}\\ \mathbf{Y}_0^{(i)} &= \mathbf{y}_0 - \sum_{j=1}^{N^{(i)}} \mathbf{Y}_j^{(i)} T_j(t_0')\end{aligned} \tag{27}$$

with

$$\begin{aligned}s_{jk}^{(i)} &= \sin\left(\frac{j(2k-1)\pi}{2N^{(i)}}\right)\\ \mathbf{g}_k^{(i)} &= \mathbf{f}_k^{(i)} s_{1k}^{(i)}\end{aligned} \tag{28}$$

Note that the expression for $\mathbf{Y}_0^{(i)}$ in Eq. (27) simply means $\mathbf{y}^{(i)}(t_0) = \mathbf{y}_0$ [18].

The recursion can be summarized as:

$$\mathbf{Y}_j^{(i-1)} \xrightarrow{\text{Eq. (24)}} \mathbf{y}_k^{(i)} \xrightarrow{\text{Eq. (26)}} \mathbf{f}_k^{(i)} \xrightarrow{\text{Eq. (27)}} \mathbf{Y}_j^{(i)} \tag{29}$$

In the above lines the two different subscripts $j$ and $k$ are used to distinguish the quantities where the function evaluation is performed and the ones that build the actual Chebyshev coefficients.

## References

[1] F. Topputo, M. Vasile, F. Bernelli-Zazzera, Earth-to-Moon low energy transfers targeting L1 hyperbolic transit orbits, Ann. New York Acad. Sci. 1065 (1) (2005) 55–76, http://dx.doi.org/10.1196/annals.1370.025.

[2] M. Ceriotti, C.R. McInnes, Design of ballistic three-body trajectories for continuous polar earth observation in the Earth-Moon system, Acta Astronaut. 102 (2014) 178–189, URL https://www.sciencedirect.com/science/article/pii/S0094576514001982.

[3] C. Short, K. Howell, A. Haapala, D. Dichmann, Mode analysis for long-term behavior in a resonant Earth-Moon trajectory, J. Astronaut. Sci. 64 (2) (2017) 156–187, http://dx.doi.org/10.1007/s40295-016-0098-9.

[4] G. Lantoine, R.P. Russell, S. Campagnola, Optimization of low-energy resonant hopping transfers between planetary moons, Acta Astronaut. 68 (7) (2011) 1361–1378, URL https://www.sciencedirect.com/science/article/pii/S0094576510003711.

[5] S. Campagnola, Y. Kawakatsu, Three-dimensional resonant hopping strategies and and the Jupiter magnetospheric orbiter, J. Guid. Control Dyn. 35 (1) (2012) 340–344, http://dx.doi.org/10.2514/1.53334.

[6] S. Campagnola, A. Boutonnet, J. Schoenmaekers, D.J. Grebow, A.E. Petropoulos, R.P. Russell, Tisserand-leveraging transfers, J. Guid. Control Dyn. 37 (4) (2014) 1202–1210, http://dx.doi.org/10.2514/1.62369.

[7] M. Vaquero, K.C. Howell, Transfer design exploiting resonant orbits and manifolds in the Saturn-Titan system, J. Spacecr. Rockets 50 (5) (2013) 1069–1085, http://dx.doi.org/10.2514/1.A32412.

[8] E.M. Zimovan, K.C. Howell, D.C. Davis, Near rectilinear halo orbits and their application in cis-lunar space, in: 3rd IAA Conference On Dynamics And Control Of Space Systems, Vol. 20, Moscow, Russia, 2017.

[9] M. McGuire, L. Burke, S. McCarty, K. Hack, R. Whitley, D. Davis, C. Ocampo, Low thrust cis-lunar transfers using a 40 kW-class solar electric propulsion spacecraft, in: AAS/AIAA Astrodynamics Specialist Conference, Stevenson (WA), USA, 2017, pp. 1–21.

[10] S. Singh, J. Junkins, B. Anderson, E. Taheri, Eclipse-conscious transfer to lunar gateway using ephemeris-driven Terminal Coast arcs, J. Guid. Control Dyn. 44 (11) (2021) 1972–1988, http://dx.doi.org/10.2514/1.G005920.

[11] EADS-Astrium, Solar orbiter, J. Phys. Conf. Ser. 271 (2011) 011004, http://dx.doi.org/10.1088/1742-6596/271/1/011004.

[12] European Space Agency (ESA), Solar Orbiter Definition Study Report (Red Book), Tech. Rep., 2011, URL https://sci.esa.int/s/w7yO4P8.

[13] E.J. Opik, Interplanetary Encounters: Close-Range Gravitational Interactions, Vol. 2, Elsevier Scientific Pub. Co, Amsterdam, 1976.

[14] A. Carusi, G.B. Valsechi, R. Greenberg, Planetary close encounters: geometry of approach and post-encounter orbital parameters, Celest. Mech. Dyn. Astron. 49 (2) (1990) 111–131, http://dx.doi.org/10.1007/BF00050709.
[15] G.B. Valsecchi, A. Milani, G.F. Gronchi, S.R. Chesley, Resonant returns to close approaches: Analytical theory, Astron. Astrophys. 408 (3) (2003) 1179–1196, http://dx.doi.org/10.1051/0004-6361:20031039.
[16] G.B. Valsecchi, Geometric conditions for quasi-collisions in Öpik's theory, in: J. Souchay (Ed.), Dynamics Of Extended Celestial Bodies And Rings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 145–158, http://dx.doi.org/10.1007/3-540-32455-0_6.
[17] G.B. Valsecchi, E.M. Alessi, A. Rossi, An analytical solution for the swing-by problem, Celest. Mech. Dyn. Astron. 123 (2) (2015) 151–166, http://dx.doi.org/10.1007/s10569-015-9631-6.
[18] T. Fukushima, Picard iteration method, Chebyshev polynomial approximation, and global numerical integration of dynamical motions, Astron. J. 113 (1997) 1909–1914, http://dx.doi.org/10.1086/118404.
[19] X. Bai, J.L. Junkins, Solving initial value problems by the picard-Chebyshev method with NVIDIA GPUs, in: Advances In The Astronautical Sciences, 2010.
[20] X. Bai, J.L. Junkins, Modified Chebyshev-Picard iteration methods for orbit propagation, J. Astronaut. Sci. 58 (4) (2011) 583–613, http://dx.doi.org/10.1007/BF03321533.
[21] J.L. Junkins, A. Bani Younes, R.M. Woollands, X. Bai, Picard iteration, Chebyshev polynomials and Chebyshev-Picard methods: Application in astrodynamics, J. Astronaut. Sci. 60 (3) (2013) 623–653, http://dx.doi.org/10.1007/s40295-015-0061-1.
[22] D. Koblick, P. Shankar, Evaluation of the modified Picard-Chebyshev method for high-precision orbit propagation, J. Aerosp. Eng. 28 (5) (2015) 04014125, http://dx.doi.org/10.1061/(ASCE)AS.1943-5525.0000463.
[23] R.M. Woollands, A. Bani Younes, J.L. Junkins, New solutions for the perturbed Lambert problem using regularization and Picard iteration, J. Guid. Control Dyn. 38 (9) (2015) 1548–1562, http://dx.doi.org/10.2514/1.G001028.
[24] R.M. Woollands, J.L. Read, A.B. Probe, J.L. Junkins, Multiple revolution solutions for the perturbed Lambert problem using the method of particular solutions and Picard iteration, J. Astronaut. Sci. 64 (4) (2017) 361–378, http://dx.doi.org/10.1007/s40295-017-0116-6.
[25] R.M. Woollands, J. Read, K. Hernandez, A. Probe, J.L. Junkins, Unified Lambert tool for massively parallel applications in space situational awareness, J. Astronaut. Sci. 65 (1) (2018) 29–45, http://dx.doi.org/10.1007/s40295-017-0118-4.
[26] T. Swenson, R. Woollands, J. Junkins, M. Lo, Application of modified Chebyshev picard iteration to differential correction for improved robustness and computation time, J. Astronaut. Sci. 64 (3) (2017) 267–284, http://dx.doi.org/10.1007/s40295-016-0110-4.
[27] S.K. Singh, R. Woollands, E. Taheri, J. Junkins, Feasibility of quasi-frozen, near-polar and extremely low-altitude lunar orbits, Acta Astronaut. 166 (2020) 450–468, http://dx.doi.org/10.1016/j.actaastro.2019.10.037, URL https://www.sciencedirect.com/science/article/pii/S0094576519313657.
[28] D. Koblick, S. Xu, J. Fogel, P. Shankar, Low thrust minimum time orbit transfer nonlinear optimization using impulse discretization via the modified picard-Chebyshev method, Comput. Model. Eng. Sci. 111 (1) (2016) http://dx.doi.org/10.3970/cmes.2016.111.001.
[29] B. Macomber, A.B. Probe, R. Woollands, J. Read, J.L. Junkins, Enhancements to modified Chebyshev-Picard iteration efficiency for perturbed orbit propagation, Comput. Model. Eng. Sci. 111 (1) (2016) 29–64, http://dx.doi.org/10.3970/cmes.2016.111.029, URL http://www.techscience.com/CMES/v111n1/27313.
[30] R. Woollands, E. Taheri, J.L. Junkins, Efficient computation of optimal low thrust gravity perturbed orbit transfers, J. Astronaut. Sci. 67 (2) (2020) 458–484, http://dx.doi.org/10.1007/s40295-019-00152-9.
[31] R. Woollands, J.L. Junkins, Nonlinear differential equation solvers via adaptive Picard-Chebyshev iteration: Applications in astrodynamics, J. Guid. Control Dyn. 42 (5) (2019) 1007–1022, http://dx.doi.org/10.2514/1.G003318.
[32] A.M. Atallah, R.M. Woollands, T.A. Elgohary, J.L. Junkins, Accuracy and efficiency comparison of six numerical integrators for propagating perturbed orbits, J. Astronaut. Sci. 67 (2) (2020) 511–538, http://dx.doi.org/10.1007/s40295-019-00167-2.
[33] A. Masat, M. Romano, C. Colombo, B-Plane Orbital Resonance Analysis and Applications, Politecnico di Milano, 2019, URL https://www.diva-portal.org/smash/get/diva2:1440101/FULLTEXT01.pdf.
[34] W. McClain, D. Vallado, Fundamentals Of Astrodynamics And Applications, in: Space Technology Library, Springer Netherlands, 2001, URL https://books.google.it/books?id=PJLlWzMBKjkC.
[35] R. Greenberg, A. Carusi, G. Valsecchi, Outcomes of planetary close encounters: A systematic comparison of methodologies, Icarus 75 (1) (1988) 1–29, http://dx.doi.org/10.1016/0019-1035(88)90125-X, URL https://www.sciencedirect.com/science/article/pii/001910358890125X.
[36] A. Milani, S.R. Chesley, P.W. Chodas, G.B. Valsecchi, Asteroid close approaches: Analysis and potential impact detection, in: Asteroids III, 2002, pp. 55–69.
[37] M. Romano, M. Losacco, C. Colombo, P. Di Lizia, Impact probability computation of near-earth objects using Monte Carlo line sampling and subset simulation, Celest. Mech. Dyn. Astron. 132 (8) (2020) 42, http://dx.doi.org/10.1007/s10569-020-09981-5.
[38] C. Colombo, F. Letizia, J. Van Der Eynde, SNAPPshot ESA Planetary Protection Compliance Verification Software Final Report V1.0, Technical Report ESA-IPL-POM-MB-LE-2015-315, Tech. Rep., University of Southampton, 2016.
[39] C. Colombo, M. Romano, A. Masat, SNAPPshot ESA Planetary Protection Compliance Verification Software Final Report V 2.0, Technical Report ESA-IPL-POM-MB-LE-2015-315, Tech. Rep., Politecnico di Milano, 2020.
[40] E. Hairer, G. Wanner, S.P. Nørsett, Solving Ordinary Differential Equations I, second ed., Springer Berlin, 1993, http://dx.doi.org/10.1007/978-3-540-78862-1.
[41] T.J. Rivlin, The Chebyshev polynomials, Math. Comput. 30 (1976) http://dx.doi.org/10.2307/2005983.
[42] B.D. Macomber, Enhancements to Chebyshev-Picard Iteration Efficiency for Generally Perturbed Orbits and Constrained Dynamical Systems (Ph.D. thesis), Texas A & M University, 2015, Supervisor: Junkins, John L., URL https://hdl.handle.net/1969.1/155745.
[43] T. Fukushima, Vector integration of dynamical motions by the picard-Chebyshev method, Astron. J. 113 (1997) 2325, http://dx.doi.org/10.1086/118443.
[44] D. Koblick, M. Poole, P. Shankar, Parallel high-precision orbit propagation using the modified picard-Chebyshev method, in: ASME International Mechanical Engineering Congress And Exposition, Proceedings, IMECE, 2012, http://dx.doi.org/10.1115/IMECE2012-87878.
[45] P.K. Seidelmann, Explanatory Supplement to The Astronomical Almanac, University Science Books, 1992.
[46] C.H. Acton, Ancillary data services of NASA's navigation and ancillary information facility, Planet. Space Sci. 44 (1) (1996) 65–70, http://dx.doi.org/10.1016/0032-0633(95)00107-7, URL https://www.sciencedirect.com/science/article/pii/0032063395001077.
[47] I. Griva, S.G. Nash, A. Sofer, Linear And Nonlinear Optimization, 2009, http://dx.doi.org/10.1137/1.9780898717730.
[48] European Space Agency (ESA), Jupiter Icy Moons Explorer Exploring the Emergence of Habitable Worlds Around Gas Giants. definition Study Report, ESA Mission Reports, Tech. Rep., (1.0) 2014, p. 128, URL https://sci.esa.int/web/juice/-/54994-juice-definition-study-report.