AAS 21-289

COMBINED B-PLANE AND PICARD-CHEBYSHEV APPROACH FOR THE CONTINUOUS DESIGN OF PERTURBED INTERPLANETARY RESONANT TRAJECTORIES

Alessandro Masat*, Matteo Romano† and Camilla Colombo‡

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 clever framework to represent close approaches at the boundaries of the sphere of influence of the flyby planet. This representation is exploited to prune the design of perturbed resonant interplanetary trajectories in reverse cascade, without patched conics approximation. The design strategy is formulated as a multi-layer optimization problem, whose core numerically integrates the perturbed orbital motion with the Picard-Chebyshev integration method. The analytical pruning solution is also used as starting guess to ensure the fast convergence of both the numerical integration and the design algorithm. The proposed semi-analytical strategy allows to surf complex gravitational perturbing effects optimizing artificial maneuvers in a computationally efficient way, and can be extended to account for any other perturbation source. The method is applied to the design of a Solar Orbiter-like quasi-ballistic first resonant phase with Venus.

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] Ceriotti et al.[2] and Short et al.[3]) 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]). Other applications also regard pure interplanetary orbits, for instance the ESA/NASA mission Solar Orbiter[8] as the latest example: resonant trajectories with Venus are exploited to raise the orbital inclination up to almost 30 degrees[9] 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,

*PhD Candidate, DAER-Department of Aerospace Science and Technology, Politecnico di Milano, Via G. La Masa 34, 20156, Milano, Italy.

†Post-doctoral researcher, DAER-Department of Aerospace Science and Technology, Politecnico di Milano, Via G. La Masa 34, 20156, Milano, Italy.

‡Associate Professor, DAER-Department of Aerospace Science and Technology, Politecnico di Milano, Via G. La Masa 34, 20156, Milano, Italy.

accurate predictions are required for steep close approaches too: a small deviation from the nominal condition may be amplified of several orders of magnitude during the flyby, making the modeling crucial or requiring maneuvers not to deviate from the desired condition.

The b-plane formalism presents an analytic theory for the characterization of steep flybys, based on a manipulation of Öpik's variables[10] originally proposed by Carusi et al.[11] 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.[12]

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.[13] Bai and Junkins[14,15] proposed a modified version of the method, making it suitable to GPU computing platforms condensing the algorithm steps in a series of matrix operations, in this work implemented following the formalism adopted by Koblick.[16]

In the proposed approach, the Picard-Chebyshev integration method is combined with the b-plane prediction capabilities and applied to the design of multi-flyby trajectories in reverse cascade. The patched conics approximation is removed, and perturbations from the N-bodies and general relativity perturbations are included in the dynamical model. 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[17] previously developed by the authors of this work, that exploits the b-plane formalism to design a series of two body resonant orbits limitedly to the patched conics case. The newly extended version of the strategy uses the unperturbed b-plane solution is to prune the trajectory design in the perturbed environment. Starting from the Keplerian initial guesses for the patched conics arcs, a continuity link between the planetary and interplanetary legs is introduced at the boundaries of the 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 the artificial correction effort at user-specified point in the interplanetary cruise. In particular, the case of Solar Orbiter's resonant close approaches with Venus[9] is studied, achieving a quasi-ballistic transfer that surfs the chaotic perturbed environment, requiring an artificial control impulse easily achievable by today's 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[18] 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 analysed, since its connection with the b-plane formalism is straightforward[12] and the already available unperturbed design routine.[17]

As the Picard-Chebyshev method can be parallelized, the whole design strategy is well suited to be used with high performance computing facilites. In spite of this, the serial execution will be shown to be already efficient, because of the limited need to read the database for the ephemerides of the N bodies.

The first steps towards 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-life maneuver design strategy, which at last would allow to fulfil the primary needs of the analysis of actual missions.

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

CLOSE ENCOUNTERS IN THE B-PLANE

Assuming the planet to follow a circular orbit around the Sun, an intermediate frame is required before defining the b-plane flyby representation, first introduced by Carusi et al.[11] The frame is 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. 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-dimensionalisation gives in turn $|\mathbf{v}_p| = 1$ and makes the orbital period of the planet equal to 2π.

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:[10]

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

From now on 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 by Milani et al.[19]

Recalling,[11] 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.

B-plane circles

A certain post-encounter semi-major axis a' is fully determined by θ', the angle between $\mathbf{U}'$ and $\mathbf{v}_p$:[12]

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

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:[12]

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

which is equivalent to

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

with the center's ζ 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 θ', θ 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, without necessarily need to be resonant. The sole exception regards flybys that do not modify the value of a, and thus feature $\theta \equiv \theta'$, which are defined as the straight horizontal line:[12]

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

Perturbations in the b-plane

Previous results by the authors of this work[17] led to the semi-analytical definition of the b-plane circles arising from the effects of a generic perturbation source. Assuming the locus of points of common post-encounter semi-major axis a' being still well approximated by circles centered on the $\hat{\zeta}$ axis, all the perturbing effects can be condensed in three angular variations:

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

Note that, since the b-plane frame is defined at the entrance of the sphere of influence and the spherical geometry of the rotation is what determines the circles, the effect of possible variations of the magnitude of $\mathbf{U}$ is included in $\Delta\theta$. Condensing once more $\theta'^{*} = \theta' + \Delta\theta'$, finding the intersections of the perturbed circle with the $\hat{\zeta}$ axis brings the following expressions for D and R:[17]

$$\begin{aligned} D &= \frac{c(\sin\theta\cos\Delta\gamma\cos\Delta\psi - \cos\theta\sin\Delta\psi)}{\cos\theta'^{*} - \cos\theta\cos\Delta\gamma - \sin\theta\sin\Delta\gamma\cos\Delta\psi} \\ R &= \left| \frac{c\sqrt{\sin^2\theta'^{*} - \sin^2\theta\sin^2\Delta\psi}}{\cos\theta'^{*} - \cos\theta\cos\Delta\gamma - \sin\theta\sin\Delta\gamma\cos\Delta\psi} \right| \end{aligned} \tag{8}$$

In the perturbed case, the expression $\zeta_{1,2} = D \pm R$ is exact for finding the intersections with the $\hat{\zeta}$ axis of the locus of points of conserved semi-major axis as well, in fact the expressions for D and R are no more singular for $\theta \equiv \theta'$.

The perturbed b-plane model can be used for the computation of the post-encounter orbital elements accounting for both the close approach and any other generic perturbation effect, and it provides an improvement if compared to simulation-predicted b-plane circles. Figure 1 compares the resonant circles drawn with the unperturbed theory (left) and the new perturbed model of Equation (8) (right) with the simulated resonant semi-major axes (yellow dots) coming from the planetary protection analysis of the upper stage of the launcher of Solar Orbiter[9,20*]. The grey dots represent instead simple close approaches currently not in resonance. Note that the circles defined in Equations (6) and (8), on purpose nearly visible and drawn in light grey, have become the black

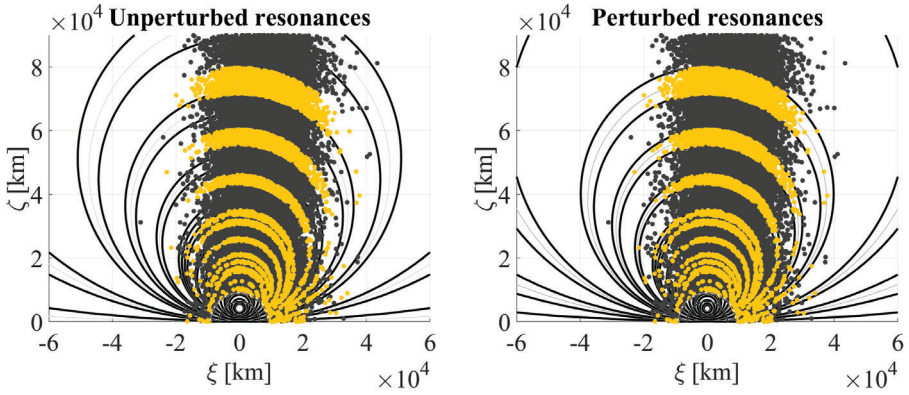


Figure 1: Perturbed b-plane resonances vs simulated resonances in the full force environment.

bounded belt shaped loci of points, because also almost perfectly phased resonant returns have been considered extending each circle over its own neighbourhood.

In Figure 1 the perturbing angles $\Delta\gamma$, $\Delta\psi$ and $\Delta\theta'$ have been computed by post-processing the results of the numerical integration of the perturbed trajectory within the sphere of influence boundaries for the barycenter of the cloud of points in Figure 1, using the Picard-Chebyhsev method. Note that $\Delta\gamma$, $\Delta\psi$ and $\Delta\theta'$ remain small in magnitude, nevertheless Figure 1 shows that 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.

PICARD-CHEBYSHEV INTEGRATION METHOD

Picard iterations[23] 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

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \qquad \mathbf{x}_0 = \mathbf{x}(t_0) \tag{9}$$

Starting from an initial approximation $\mathbf{x}^{(0)}(t)$ of the actual solution $\mathbf{x}(t)$ of the initial/boundary value problem presented in Equation (9), the i-th Picard iteration improves the previous approximation $\mathbf{x}^{(i-1)}(t)$ of $\mathbf{x}(t)$ with $\mathbf{x}^{(i)}(t)$ as[23]

$$\mathbf{x}^{(i)}(t) = \mathbf{x}^{(0)}(t) + \int_{t_0}^{t} \mathbf{f}\big(\mathbf{x}^{(i-1)}(s), s\big) ds \tag{10}$$

The method converges for a good enough initial approximation $\mathbf{x}^{(0)}(t)$ and for $i \longrightarrow +\infty$.[23]

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.[14] 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

*More detailed information about this analysis and the related validation can be found in the work of Colombo et al.,[21] Colombo et al[22] and Masat.[17]

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.[24] A possible derivation of the method follows the work of Fukushima,[13] which 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. Picard-iterate 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, thus delegating the function approximation to a finer level. 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.[23] Starting from the unperturbed Keplerian solution for the weakly perturbed two body problem, a relatively fast convergence of the method is ensured.[13]

Matrix form for vectorized and parallel computation

The method is suitable for parallel or vector implementation, indeed Fukushima also proposed a vectorized version.[25] More recent works over this technique by Bai and Junkins developed the modified Picard-Chebyshev method[15] and a CUDA implementation for NVIDIA GPUs.[14] For compactness and to better highlight the parallelisation possibilities, the method is presented using the matrix formulation by Koblick et al,[16] still based on the work of Bai and Junkins.[14,15]

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

$$t_j = \omega_2 \tau_j + \omega_1 \tag{11}$$

with

$$\tau_j = -\cos\left(\frac{j\pi}{N-1}\right), \quad \omega_1 = \frac{t_{N-1} + t_0}{2}, \quad \omega_2 = \frac{t_{N-1} - t_0}{2} \tag{12}$$

The values of the Chebyshev polynomials and their derivatives at the selected N nodes do not directly enter the Picard iterations, but require some manipulation to obtain the constant matrices $\mathbf{C}$ and $\mathbf{A}$, which can be computed up-front as well. $\mathbf{C}$ is $N \times N$ and its elements are defined as[16]

$$\begin{aligned} \mathbf{C}_{j+1,1} &= \frac{1}{2} T_0(\tau_j), \quad j = 0, ..., N-1 \\ \mathbf{C}_{j+1,k+1} &= T_k(\tau_j), \quad j = 0, ..., N-1 \quad \text{and} \quad k = 1, ..., N-1 \end{aligned} \tag{13}$$

whereas $\mathbf{A}$ is $N-1 \times N$, with elements

$$\begin{aligned} \mathbf{A}_{k+1,j+1} &= \frac{T_k(\tau_j) - T_{k+2}(\tau_j)}{2(j+1)}, \quad j, k = 0, ..., N-2 \\ \mathbf{A}_{k+1,N} &= \frac{T_k(\tau_{N-1})}{2(N-1)}, \qquad\qquad k = 0, ..., N-2 \end{aligned} \tag{14}$$

where $T_k(\tau) = \cos(k \arccos(\tau))$ is the Chebyshev polynomial of degree k.[13] Another constant quantity that is involved in the iteration process is the row matrix $\mathbf{S}$ of dimension $1 \times N-1$, whose elements are[16]

$$\mathbf{S}_k = 2(-1)^{k+1}, \qquad k = 1, ..., N-1 \tag{15}$$

Finally, given the n-dimensional sampled states $\mathbf{y}^{(i-1)}(t_j) = \mathbf{y}_j^{(i-1)}, \ j = 0, ..., N$ 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 $\mathbf{F}$:[16]

$$\mathbf{F}_{j+1}^{(i)} = \omega_2 \ \mathbf{f}\big(\mathbf{y}_j^{(i-1)}, t_j\big), \quad j = 0, ..., N-1 \tag{16}$$

each of which can be performed in parallel, as well as all the upcoming matrix elementary operations.

Secondly, the $N \times n$ matrix $\mathbf{B}$ is obtained by rows as[16]

$$\mathbf{B}_1 = \mathbf{SAF} + 2\mathbf{y}_0, \quad \mathbf{B}_j = \mathbf{AF}, \quad j = 2, ..., N \tag{17}$$

Note that the boundary values $\mathbf{y}_0$ remain constant for all the Picard iterations. Third and last, the $N \times n$ matrix of the state guesses $\mathbf{y}^{(i)}$ for the i-th Picard iteration is

$$\mathbf{y}^{(i)} = \mathbf{CB} \tag{18}$$

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.

As already underlined by Fukushima[13] and Bai and Junkins,[14,15] 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. Fukushima[13] 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* until the end of the span.

The core steps of the proposed algorithm follow the presented scheme,[14,15] together with the automatic generation of the Keplerian initial guess spanning one nominal orbital period. The proposed implementation interfaces with SPICE ephemerides[26] considering them a parameter of the Picard iteration process, as will be better detailed in the following sections.

CONTINUOUS MULTI-FLYBY DESIGN IN THE RELATIVISTIC N-BODY PROBLEM

The chaotic nature of the flyby problem makes it necessary to keep a high precision level to get an accurate prediction of the trajectory after a flyby. Here the reason to account for relativistic effects, as well as to keep all the considered tolerances low. The effects of solar radiation pressure are neglected for now, nonetheless its future inclusion shall have no impact at all in the presented algorithms as only a minor (indeed just another perturbation) update of the dynamical model will be required. General relativity effects have been previously implemented by the authors of this

*The proposed implementation automatically handles either forward or backward integration.

work,[17,22] based on the Post-Newtonian model of the Einstein-Ingfeld-Hoffmann equations as presented by Seidelmann.[27] The same set of equations is used by the Jet Propulsion Laboratory (JPL) for their simulations generating ephemerides data,[26] 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. Assuming to know only some post-encounter macro-properties to be achieved 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, 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 each next flyby.

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

Patched conics b-plane solution for resonant orbits

Valsecchi et al.[28,29] 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 algorithmic and highly non-linear: 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,[17] defining an efficient optimization problem that uses the spherical geometry relations that generate the b-plane circles. Provided that the desired post-encounter condition is reachable and has the same Tisserand parameter of the pre-encounter one, the algorithm finds the point (ξ, ζ) on a specified b-plane circle associated to the rotation of $\mathbf{U}$ that gets the closest to the desired exit condition $\mathbf{U}'$.

It is also possible to include some mission constraints, such as not getting too close to the planet, as this implies only excluding certain portions of the circle/straight line, as well as including the perturbing effects according to the angular model $(\Delta\gamma, \Delta\psi, \Delta\theta')$. The values of such angles computed on a specified b-plane point were shown to significantly improve the prediction of the transformation of $\mathbf{U}$ into $\mathbf{U}'$ also in the neighbourhood of the point itself.[17]

Specifically mentioning to the case of resonances, another optimization layer was developed:[17] 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. In the unperturbed and patched conics context, 1-2 seconds only[17] were 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.[9]

When accounting for perturbing effects in the b-plane, a modification of the circles is inevitably introduced, as already shown in Figure 1. 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.[17]

Integration method adaptation and precision assessment

Koblick[16] used the modified Picard-Chebyshev method to integrate the two body problem in the planetary environment accounting for several perturbing effects. In this work the focus is brought toward an interplanetary application, with particular interest on the perturbations introduced by the gravitational field of the major bodies of the Solar System. As already briefly mentioned, the SPICE toolkit[26] is used together with JPL's ephemerides data to retrieve the states of the N bodies at any integration step. 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.[21] 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.

Using the 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:

$$\ddot{\mathbf{r}}(t) = -\sum_{i=1}^{N} \frac{\mu_i(\mathbf{r}(t) - \mathbf{r}_i(t))}{|\mathbf{r}(t) - \mathbf{r}_i(t)|^3} \tag{19}$$

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 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 data 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, together with the parallelization possibilities offered by the method, can be used to speed up numerical simulations in the interplanetary environment, provided that enough computational power is available and 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 conceptual complexity to the problem. The more complex dynamics can be simply summarized as $\ddot{\mathbf{r}}(t) = \mathbf{f}(\mathbf{r}_i(t), \dot{\mathbf{r}}_i(t))\ \ i = 1, ..., N$, 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:[13] 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.

Figure 2 shows the evolution of the relative position error with respect to JPL's data for the near-Earth asteroid 2010RF$_{12}$ from 1$^{\text{st}}$ 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. For each leg, the initial guess is the Keplerian solution, elliptical or hyperbolic depending on the current status. The flyby time is known, so three legs in total are shown*. It can

*The flyby detection routine is necessary to use the integrator as a whole, nevertheless for the design purposes of this

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*.

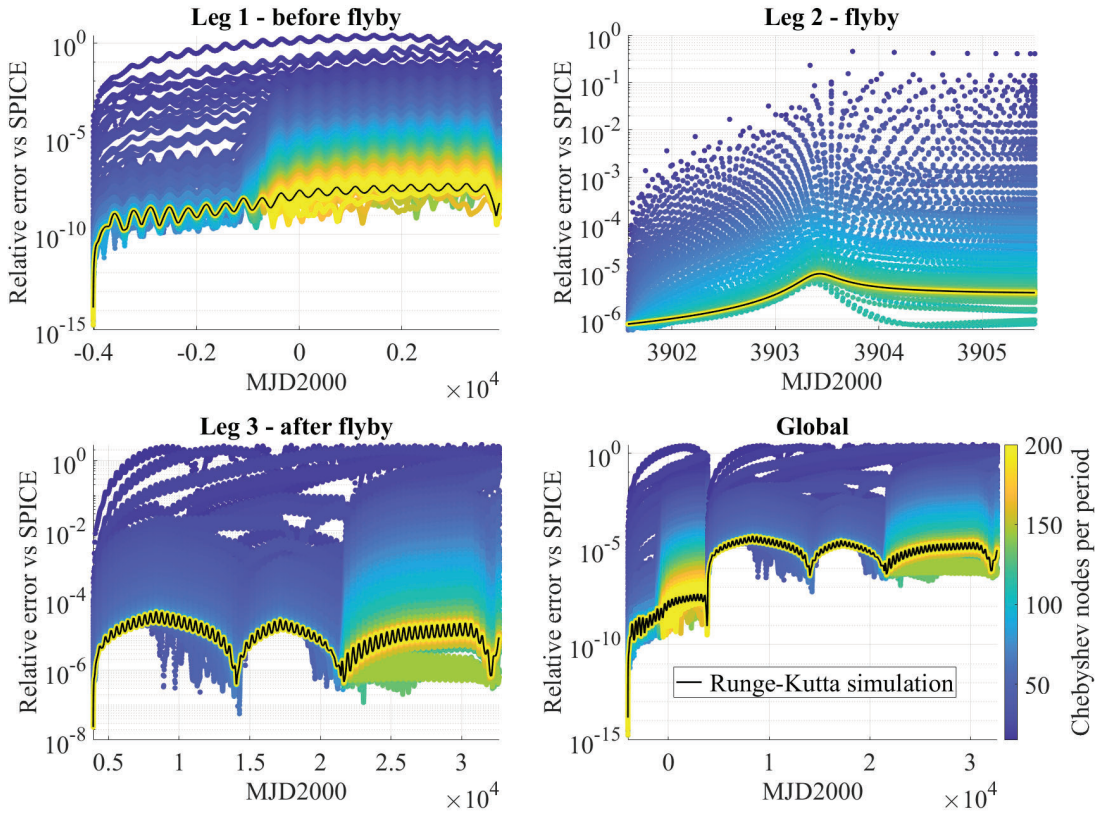


Figure 2: Picard-Chebyshev and Runge-Kutta RK78 integration errors, as relative position difference with respect to JPL's data for the asteroid 2010RF$_{12}$.

Event detection routines, such as for flybys or impacts, can complete the integrator as a whole. Their implementation will be explored in future works. Figure 3 presents instead the relative relationship between the execution time of the serial Picard-Chebyshev method and the number of Chebyshev nodes. The almost linear relationship highlights once again the possible parallelization benefits, which can be exploited to reach a high accuracy level increasing the number of nodes. A detailed analysis of the absolute computational performances, including scalability, speedup properties and implementation language influence, will also be included in future works upon completion of the whole integration strategy. To provide a first order of magnitude, a MATLAB® not parallelized implementation with a MEX® function for the dynamics† requires about 15 seconds to complete all the three legs presented in Figure 2, on a single core of a local workstation equipped

work all the legs and times are known beforehand.

*The test case was extensively discussed by Masat[17] for the validation of the implementation of relativistic effects.

†The MEX® function was generated with MATLAB®'s Code Generation Toolbox.

with an Intel® Core™ i7-7700 CPU (3.60 GHz).

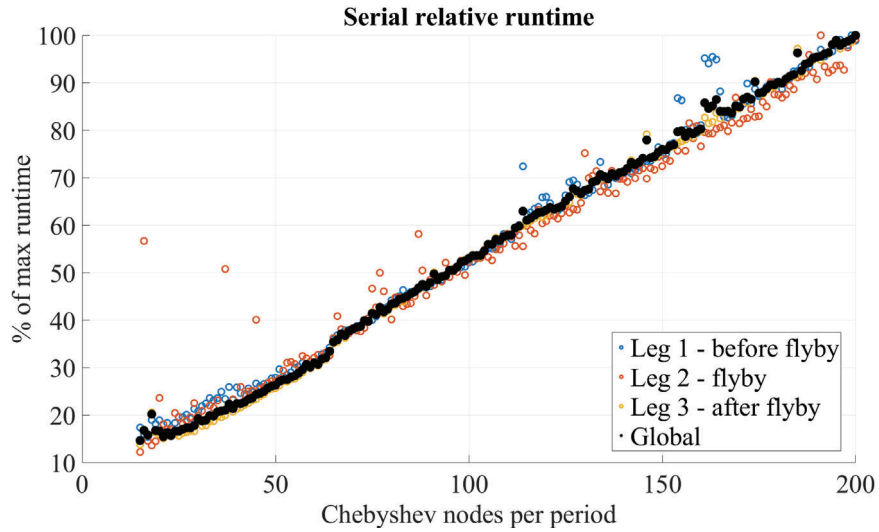


Figure 3: Picard-Chebyshev serial execution runtimes for the asteroid 2010RF$_{12}$.

B-plane pruned recursive optimization: reverse cascade flyby design

A recursive strategy for multi-flyby design can be built wrapping up the concepts presented so far. The 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[17] 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.

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 manoeuvres 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 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 a Lambert's problem[18] with the related planetocentric phase and not necessarily from the already mentioned b-plane algorithm,[17] particularly as the time $t^{(j)}$, the b-plane coordinates (ξ, ζ), and the planetocentric outgoing 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*, forcing the remaining b-plane coordinate η 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 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, optimizing over a "perturbation" $\Delta t^{(j)}$ of the exit time and bounding the search to a relatively small domain.

A similar reasoning is made for the b-plane coordinates (ξ, ζ) 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 ephemerides database for the time $t_{\Delta}^{(j)}$;
2. from $(\xi + \Delta\xi, \zeta + \Delta\zeta)$ the third b-plane coordinate η 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 Equation (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 is solved numerically forward in time with the Picard-Chebyshev method, to the connection maneuver at $\tilde{t}$. It can be wrapped denoting the dynamics functions $\mathbf{r}_f(t)$ and $\mathbf{v}_f(t)$, with $t_{out}^{(j)} \leq t \leq \tilde{t}$, for position and velocity respectively, and setting $t_0 = t_{out}^{(j)}$, $\mathbf{r}(t_0) = \mathbf{r}_{out}^{(j)}$ and $\mathbf{v}(t_0) = \mathbf{v}_{out}^{(j)}$ as initial conditions. In general, $\left(\mathbf{r}_f(\tilde{t}), \mathbf{v}_f(\tilde{t})\right)$ will differ from the back-integrated state that leads to flyby $j+1$ through $\widetilde{\Delta\mathbf{r}} = \mathbf{r}_f(\tilde{t}) - \tilde{\mathbf{r}} \neq \mathbf{0}$ and $\widetilde{\Delta\mathbf{v}} = \mathbf{v}_f(\tilde{t}) - \tilde{\mathbf{v}} \neq \mathbf{0}$.

*Not reported here. See for instance Curtis[18] for more details.

The physics of the correction maneuver performed at $\tilde{t}$ 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 lives in between the function evaluation and the connection state, thus new numerical integrations would be required for the constraint alone. 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,[30] 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 $\Delta U'^{(1,2,3)}$, the general maneuver design can be written as the following optimization problem:

$$\begin{aligned} \underset{\Delta\xi,\,\Delta\zeta,\,\Delta\mathbf{U}'}{\text{minimize}} \quad & J_v\left(\tilde{t}, t_\Delta^{(j)}\right) + \alpha J_r\left(\tilde{t}, t_\Delta^{(j)}\right) \\ \text{subject to} \quad & |\Delta\xi| \leq \Delta\xi_{max}, \\ & |\Delta\zeta| \leq \Delta\zeta_{max}, \\ & |\Delta U'^{(1,2,3)}| \leq \Delta U'^{(1,2,3)}_{max} \end{aligned} \tag{20}$$

with $J_r = \left|\widetilde{\Delta\mathbf{r}}\right|$ and α sufficiently large. Finally, assuming the position constraint to be fulfilled in the optimization problem of Equation (20) whose result gives $J_v^*\left(\tilde{t}, t_\Delta^{(j)}\right) + \alpha J_r^*\left(\tilde{t}, t_\Delta^{(j)}\right)$, the outer layer optimizing the flyby time consists of breaking down $t_\Delta^{(j)}$ into $t_{out}^{(j)} + \Delta t^{(j)}$:

$$\begin{aligned} \underset{\Delta t^{(j)}}{\text{minimize}} \quad & J_v^*\left(\tilde{t}, t_{out}^{(j)} + \Delta t^{(j)}\right) \\ \text{subject to} \quad & |\Delta t^{(j)}| \leq \Delta t_{max} \end{aligned} \tag{21}$$

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 Equation (20) is a sub-problem of the optimization problem of Equation (21). This somehow enhances the flexibility of the approach, i.e. the inner layer 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 both the layers are still 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 optimization layer can be easily defined to find the best $\tilde{t}$ similarly to what done for $t_\Delta^{(j)}$ in Optimzation Problem (21), and in the presented formalism it shall straightforwardly include the innermost layer defined by the optimization problem of Equation (20).

Some observations regarding the expected computational performances of the optimization can be made based on the analysis of the Picard-Chebyshev integration method, already presented in

Figure 2 for the accuracy behavior and Figure 3 for the computational time variation with increasing number of Chebyshev nodes. Keeping on hold the parallelization possibilities, it can be said that 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 Equation (20) 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, a high acceleration of the whole solution search process is expected with properly parallelized implementations of both the integration method and the optimization algorithm. The penalty approach[30] used to define the optimization problem of Equation (20) allows for massively parallel and brute force strategies to be implemented as well, because all the remaining constraints are of boundary type.

APPLICATION: SOLAR ORBITER'S FIRST RESONANT PHASE WITH VENUS

The presented design procedure has been tested on a phase of the ongoing mission Solar Orbiter,[8] particularly taking the initial data from the trajectory profile with launch in January 2018* available in the mission redbook.[9]

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 Equation (20) is solved with the `fmincon.m` function of MATLAB®'s Optimization Toolbox, using both the Interior-point and Sequential Quadratic Programming methods,[30] based on the dimension of the search space.

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,[9] 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.

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 following interplanetary state:

$$\mathbf{r}_{in}^{(j+1)} = \begin{Bmatrix} -67036625.74 \\ -85735252.56 \\ 2558691.33 \end{Bmatrix} \text{km}, \quad \mathbf{v}_{in}^{(j+1)} = \begin{Bmatrix} 30.54 \\ -4.05 \\ 1.79 \end{Bmatrix} \text{km/s}, \quad t_{in}^{(j+1)} = 8119.84 \text{ MJD2000} \tag{22}$$

*Later discarded, the actual mission left Earth on February 2020.

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 travelled for 4 of its orbital periods. The output of the b-plane preliminary optimization layer[17] enforcing the 3/4 resonance has given the following pruning quantities:

$$\begin{Bmatrix}\xi\\ \zeta\end{Bmatrix} = \begin{Bmatrix}8484.11\\ 6432.19\end{Bmatrix}\text{km}, \quad \mathbf{U}' = \begin{Bmatrix}3.08\\ 17.78\\ 3.66\end{Bmatrix}\text{km/s}, \quad t_{out}^{(j)} = 7446.50 \ \text{MJD2000} \tag{23}$$

The maneuvering time is set as a parameter, particularly at the nominal* apocenter of the first interplanetary resonant orbit. For the starting guess:

$$\tilde{\mathbf{r}} = \begin{Bmatrix}-133539124.84\\ -31992755.47\\ -4424590.99\end{Bmatrix}\text{km}, \quad \tilde{\mathbf{v}} = \begin{Bmatrix}5.08\\ -20.43\\ 1.65\end{Bmatrix}\text{km/s}, \quad \tilde{t} = 7570.89 \ \text{MJD2000} \tag{24}$$

The maximum values where to bound $\left(\Delta\xi_{max}, \Delta\zeta_{max}, \Delta U_{max}^{\prime(1,2,3)}\right)$ have been set as 1% of the impact parameter[19] $b = \sqrt{\xi^2 + \zeta^2}$ and of $|\mathbf{U}'|$ for the b-plane coordinates and the velocity components respectively. The boundary value for the exit time variation Δt_{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 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 Equation (20) 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.

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 Equation (20) is solved several times in cascade, 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 probelm, up to an absolute minimum of 10^{-8} starting from the already introduced $\pm$ 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};

*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.

- the penalty factor α 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 α by a factor 10 each time, up to 10^9;
- the the "interior point" algorithm in `fmincon.m` is selected for the first half optimization problems, whereas "sqp" is used in the last ones because of the smaller search space;
- 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*.

the optimization problem of Equation (21) 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 Δt_{max} by a factor 10 for 5 times, from the initial grid size equal to $\pm$ 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.

Note that both the function `fmincon.m` and the algorithm implemented for the outer layer can run in parallel. Direct search methods are particularly suited for parallelization, the possible implementation in high performance computing facilities might abandon descent algorithms like "interior point" and "sqp" in favour of a finer and parallel domain scanning strategy.

Results

Solving the optimization problem of Equation (20) with the above described implementation took about 2-3 minutes on a single core of a local workstation equipped with an Intel® Core™ i7-7700 CPU (3.60 GHz).

The 200 nodes algorithm† converged to the following residual $\Delta\mathbf{r}^*$ and impulsive action $\Delta\mathbf{v}^*$ for the required maneuver:

$$\Delta\mathbf{r}^* = \begin{Bmatrix} -1.92 \\ -1.23 \\ 1.81 \end{Bmatrix} \text{m}, \quad \Delta\mathbf{v}^* = \begin{Bmatrix} -0.88 \\ 0.85 \\ 1.59 \end{Bmatrix} \text{m/s}, \tag{25}$$

One should note both how small the correction effort is, despite the execution point $\tilde{t}$ is yet to be optimized, and the fulfilment 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:

$$\begin{aligned} |\Delta\mathbf{v}^*| &= 2.01 \ \text{m/s}, \quad \frac{|\Delta\mathbf{v}^*|}{|\tilde{\mathbf{v}}|} = 9.49 \times 10^{-5} \\ |\Delta\mathbf{r}^*| &= 2.91 \ \text{m}, \quad \frac{|\Delta\mathbf{r}^*|}{|\tilde{\mathbf{r}}|} = 2.12 \times 10^{-11} \end{aligned} \tag{26}$$

*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.

†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.

Most of the computational time was required to fulfil 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 $t_{out}^{(j)*}$ was found to be slightly higher than the initial guess $t_{out}^{(j)}$, and together with the interplanetary optimal starting state the initial condition is

$$\mathbf{r}_{out}^{(j)*} = \begin{Bmatrix} -64960081.70 \\ -85998322.99 \\ 2681916.07 \end{Bmatrix} \text{km}, \quad \mathbf{v}_{out}^{(j)*} = \begin{Bmatrix} 31.01 \\ -3.44 \\ 1.78 \end{Bmatrix} \text{km/s}, \quad t_{out}^{(j)*} = 7446.52 \text{ MJD2000} \tag{27}$$

Retrieving the b-plane coordinates (ξ^*, ζ^*) and the planetocentric velocity $\mathbf{U}'^*$ from $\left(\mathbf{r}_{out}^{(j)*}, \mathbf{v}_{out}^{(j)*}\right)$ and Venus' position at $t_{out}^{(j)*}$ proves the optimal pruning brought by the b-plane prediction

$$\begin{Bmatrix} \xi^* \\ \zeta^* \end{Bmatrix} = \begin{Bmatrix} 8484.69 \\ 6542.04 \end{Bmatrix} \text{km}, \quad \mathbf{U}'^* = \begin{Bmatrix} 3.26 \\ 17.76 \\ 3.67 \end{Bmatrix} \text{km/s} \tag{28}$$

The value of ζ^* looks slightly (0.02%) 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.

Figure 4 shows the difference between the designed trajectory with respect to the 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 Figure 2. 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 an higher number of nodes should be kept the more precise the design needs to be. The periodic "hills" visible in Figure 4 happen far from the domain boundaries where the nodes are thicker*. They are likely due to the interpolation of the Picard-Chebyshev solution over the simulation time steps, necessary to compute 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, Figures 5a and 5b show the continuous trajectory that embeds the planetocentric phases for both the flybys V2 and V3, togehter 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 (Figure 5a), and zooming over the flyby regions the new continuity feature can be recognized (Figure 5b).

*Because of the definition of Chebyshev nodes in Equations (11) and (12).

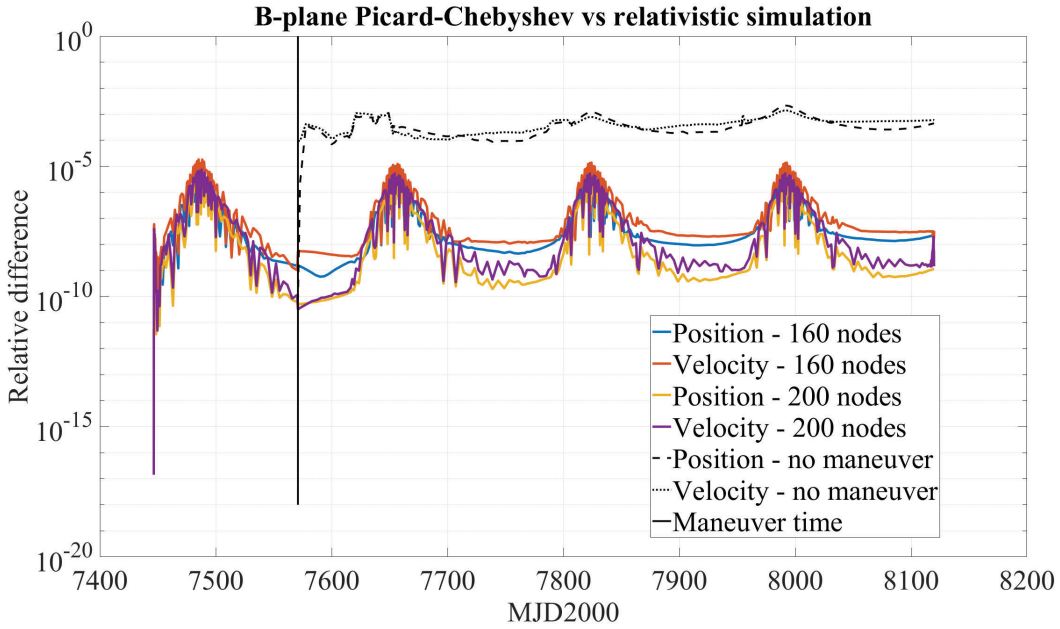
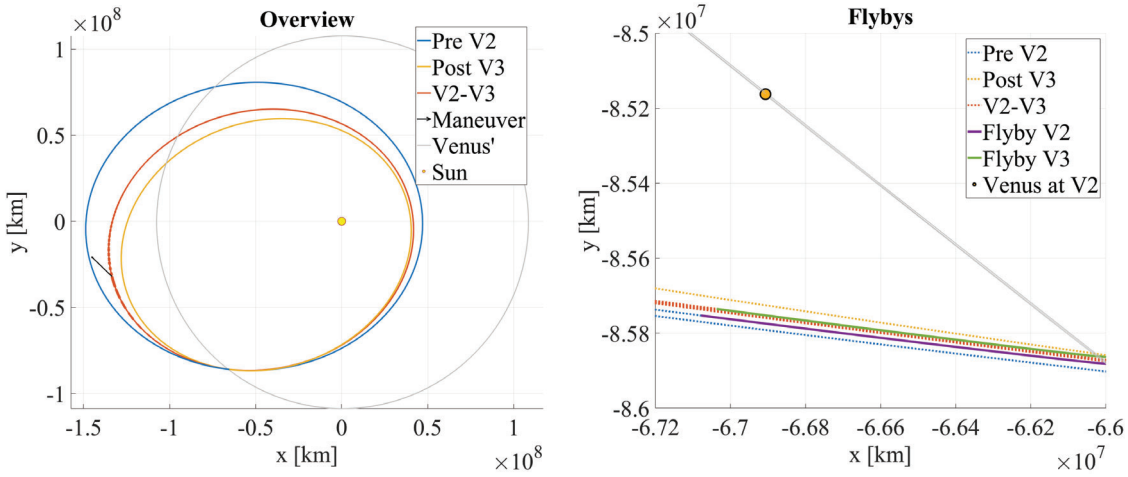


Figure 4: Design difference with respect to relativistic simulation between V2 and V3 for the two node cases, and without manoeuvre.



(a) Deep space correction maneuver and overview.

(b) Zoom over Venus' flybys V2 and V3.

Figure 5: Solar Orbiter's continuous first resonant phase with Venus.

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 for-

malism 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[8] needs. This aspect might be tackled with the proposed strategy before the design of any manevuer, 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 serial 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 sequential multi-step solution of the optimization problem of Equation (20) 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 Picard-Chebyshev method and the solution of the optimization problem.

Finally, 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 towards 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[31] could benefit from this design strategy, since moving from the interplanetary to the jovian system would not require major changes at all.

ACKNOWLEDGMENT

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)*.

REFERENCES

[1] F. Topputo, M. Vasile, and F. Bernelli-Zazzera, "Earth-to-moon low energy transfers targeting L1 hyperbolic transit orbits," *Annals of the New York Academy of Sciences*, 2005, 10.1196/annals.1370.025.

[2] M. Ceriotti and C. R. McInnes, "Design of ballistic three-body trajectories for continuous polar earth observation in the Earth-Moon system," *Acta Astronautica*, 2014, 10.1016/j.actaastro.2014.06.001.

[3] C. Short, K. Howell, A. Haapala, and D. Dichmann, "Mode Analysis for Long-Term Behavior in a Resonant Earth–Moon Trajectory," *Journal of the Astronautical Sciences*, 2017, 10.1007/s40295-016-0098-9.

*www.compass.polimi.it

[4] G. Lantoine, R. P. Russell, and S. Campagnola, "Optimization of low-energy resonant hopping transfers between planetary moons," *Acta Astronautica*, 2011, 10.1016/j.actaastro.2010.09.021.

[5] S. Campagnola and Y. Kawakatsu, "Three-dimensional resonant hopping strategies and the Jupiter magnetospheric orbiter," *Journal of Guidance, Control, and Dynamics*, 2012, 10.2514/1.53334.

[6] S. Campagnola, A. Boutonnet, J. Schoenmaekers, D. J. Grebow, A. E. Petropoulos, and R. P. Russell, "Tisserand-leveraging transfers," *Journal of Guidance, Control, and Dynamics*, 2014, 10.2514/1.62369.

[7] M. Vaquero and K. C. Howell, "Transfer design exploiting resonant orbits and manifolds in the Saturn-Titan system," *Journal of Spacecraft and Rockets*, 2013, 10.2514/1.A32412.

[8] EADS-Astrium, "Solar Orbiter," *Journal of Physics: Conference Series*, 2011, 10.1088/1742-6596/271/1/011004.

[9] E. S. Agency, "Solar Orbiter Definition Study Report (Red Book)," Tech. Rep. July, 2011, 10.1088/1742-6596/271/1/011004.

[10] E. J. Opik, "Interplanetary Encounters: Close-Range Gravitational Interactions," Vol. 2, 1976.

[11] A. Carusi, G. B. Valsechi, and R. Greenberg, "Planetary close encounters: geometry of approach and post-encounter orbital parameters," *Celestial Mechanics and Dynamical Astronomy*, 1990, 10.1007/BF00050709.

[12] G. B. Valsecchi, A. Milani, G. F. Gronchi, and S. R. Chesley, "Resonant returns to close approaches: Analytical theory," *Astronomy & Astrophysics*, 2003, 10.1051/0004-6361:20031039.

[13] T. Fukushima, "Picard Iteration method, Chebyshev Polynomial Approximation, and Global Numerical Integration of Dynamical Motions," *The Astronomical Journal*, 1997, 10.1086/118404.

[14] X. Bai and J. L. Junkins, "Solving initial value problems by the Picard-Chebyshev method with NVIDIA GPUS," *Advances in the Astronautical Sciences*, 2010.

[15] X. Bai and J. L. Junkins, "Modified Chebyshev-Picard iteration methods for solution of initial value problems," *Advances in the Astronautical Sciences*, 2011.

[16] D. Koblick, M. Poole, and P. Shankar, "Parallel high-precision orbit propagation using the modified picard-chebyshev method," *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 2012, 10.1115/IMECE2012-87878.

[17] A. Masat, M. Romano, and C. Colombo, "B-plane orbital resonance analysis and applications," master's thesis, Politecnico di Milano, 2019.

[18] H. D. Curtis, *Orbital Mechanics for Engineering Students*. 2013, 10.1016/C2011-0-69685-1.

[19] A. Milani, S. R. Chesley, P. W. Chodas, and G. B. Valsecchi, "Asteroid Close Approaches: Analysis and Potential Impact Detection," *Asteroids III*, 2002.

[20] M. Romano, M. Losacco, C. Colombo, and P. Di Lizia, "Impact probability computation of Near-Earth Objects using Monte Carlo Line Sampling and Subset Simulation," *Celestial Mechanics and Dynamical Astronomy*.

[21] C. Colombo, F. Letizia, and J. Van Der Eynde, "SNAPPshot ESA planetary protection compliance verification software Final report, Technical Report ESA-IPL-POM-MB-LE-2015-315," tech. rep., University of Southampton, 2016.

[22] C. Colombo, M. Romano, and 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.

[23] E. Hairer, G. Wanner, and S. P. Nørsett, *Solving Ordinary Differential Equations I*. Springer Berlin, 2nd ed., 1993, 10.1007/978-3-540-78862-1.

[24] T. J. Rivlin, "The Chebyshev Polynomials," *Mathematics of Computation*, 1976, 10.2307/2005983.

[25] T. Fukushima, "Vector Integration of Dynamical Motions by the Picard-Chebyshev Method," *The Astronomical Journal*, 1997, 10.1086/118443.

[26] C. H. Acton, "Ancillary data services of NASA's navigation and Ancillary Information Facility," *Planetary and Space Science*, 1996, 10.1016/0032-0633(95)00107-7.

[27] P. K. Seidelmann, *Explanatory Supplement To The Astronomical Almanac*. University Science Books, 1992.

[28] G. B. Valsecchi, "Geometric conditions for quasi-collisions in Öpik's theory," *Lecture Notes in Physics*, 2006, 10.1007/3-540-32455-0-6.

[29] G. B. Valsecchi, E. M. Alessi, and A. Rossi, "An analytical solution for the swing-by problem," *Celestial Mechanics and Dynamical Astronomy*, 2015, 10.1007/s10569-015-9631-6.

[30] I. Griva, S. G. Nash, and A. Sofer, *Linear and Nonlinear Optimization*. 2009, 10.1137/1.9780898717730.

[31] European Space Agency (ESA), "Jupiter ICy moons Explorer Exploring the emergence of habitable worlds around gas giants. Definition Study Report.," Tech. Rep. 1.0, 2014.