Global Trajectory Optimisation: Can We Prune the Solution Space when Considering Deep Space Maneuvers?

Final Report

Authors: F. Bernelli-Zazzera, M. Lavagna, R. Armellin, P. Di Lizia, F. Topputo
Affiliation: Aerospace Engineering Department, Politecnico di Milano

Authors: M. Berz
Affiliation: Department of Physics and Astronomy, Michigan State University

ESA Researcher(s): Tamás Vinkó

Date: December 2007

Contacts:

Franco Bernelli-Zazzera
Tel: +39 02 2399 8328
Fax: +39 02 2399 8334
e-mail: franco.bernelli@polimi.it

Leopold Summerer
Tel: +31(0)715655174
Fax: +31(0)715658018
e-mail: act@esa.int

Available on the ACT website
http://www.esa.int/act

Ariadna ID: 06/4101
Study Duration: 6 months
Contract Number: 2007/06/NL/HI

Contents

List of Figures

List of Tables

Abstract

It has been recently shown that the solution space of multiple gravity assist optimization problems can be pruned considerably in polynomial time. The basic idea behind this technique is to reduce the problem into a cascade of two dimensional subproblems and to prune the design space by evaluating the objective function and the constraints on a grid which samples the design space. When deep space maneuvers are introduced the complexity of the problem increases considerably due to the necessarily added dimensions, and to the larger number of local minima introduced. This study considers differential algebraic techniques as an effective tool to attach this more demanding problem. As far as differential algebra is used, the objective function and the constraints of the problem are represented by Taylor series of desired order, over boxes in which the design domain is split. Thanks to the polynomial representation of the function and the constraints, a coarse grid can be used and an efficient design space pruning can be performed. Furthermore, once the domain has been pruned, a suitable manipulation of the polynomials can ease the subsequent local optimization process, so avoiding the use of any stochastic optimiser. These two aspects, connected with an efficient management of the list of boxes in which the design space can be decomposed, make differential algebraic techniques a powerful tool for the design of multiple gravity assist transfers including deep space maneuvers, so supporting the achievement of a further step to fully automate the trajectory design problem. Additional effort is devoted to develop alternative strategies for the ultimate goal of optimizing multiple gravity assist transfers involving deep space maneuvers, and to investigate the performances of validated global optimization techniques on typical space trajectory design problems.

Key words: Global optimisation, solution space pruning, differential algebraic techniques, multiple gravity assist, deep space maneuver.

Chapter 1

Introduction

Interplanetary space trajectories are usually designed in the frame of the patched-conics technique. In this context, different conic arcs, solutions of a number of Lamberts problems, are linked together to define the whole transfer trajectory suitable for the mission considered. The patched-conics method is based on a two-body representation of the dynamics and on instantaneous velocity changes provided by chemical high thrust engines.

The patched-conics method allows the designer to define multiple gravity assist (MGA) transfers: complex trajectories made up by a sequence of planet-to-planet transfers in which the spacecraft exploits each planet encounter to achieve a velocity change. This method is well established in astrodynamics, and several past missions have used MGA trajectories to reach both inner and outer planets. It is remarkable the case of the Voyager spacecraft that flew-by Jupiter, Saturn, Uranus, and Neptune exploiting a "lucky" configuration of these planets to leave the Solar System. The first MGA trajectories were designed by hand with ad hoc methods developed for a specific mission. In these cases, it was important to find a solution to the problem, rather than to find the best solution. This was due to the high degree of complexity given by the relative motion of the planets. The more planets were encountered, the more difficult was to find a feasible solution.

In the last two decades, mission designers have exploited the benefits of approaching complex MGA problems from a global optimization standpoint. Nowadays, the aim of the trajectory design is not only to find a solution, but also to find the best solution in terms of propellant consumption, while still achieving the mission goals. In the formalism of global optimization, this means that the problem consists in looking for the optimal solution in those regions of the search space that satisfy the problem constraints. Unfortunately, the MGA problems are characterized by an objective function with a large number of clustered minima, which are prevalently associated

to the complex relative motion of the planets involved in the transfer, and to the nonlinearities governing the simple Kepler's problem. This causes local optimization gradient–based methods to converge to one of these local minima. Hence, despite their efficiency, Newton–based methods should be avoided when looking for the global minimum of a MGA problem, at least in the first stage of the search process. This means that global optimization algorithms should be used to find the best solution to a MGA problem. However, such algorithms might be computationally inefficient if used as "black box" tools due to the high dimensions of the search space, and to the landscape of the objective function cited above. Thus, the key point would be the use of a global optimization algorithm, able to exploit the structure of the search space and the nature of the MGA problem itself.

It has been shown that the solution space of a MGA optimization problem can be pruned considerably in polynomial time. This observation was successfully coded in the gravity assist space pruning (GASP) algorithm [23]. The basic idea behind this algorithm is reducing the problem into a cascade of two–dimensional sub–problems, and pruning the design space by evaluating the objective function and the constraints on a sampling grid. In this way, the search space is pre–processed, and further global optimization algorithms are employed in the reduced domain. This procedure showed better performances if compared with the standard implementation of some stochastic global optimization solvers over the entire search space. Consequently, the combination of a systematic technique (space pruning) and a stochastic global optimiser (differential evolution, multiple or simple particle swarm optimization, genetic algorithm) produces remarkable numerical burden reduction.

Further improvements are obtained by formalizing the problem in terms of epochs, so avoiding redundant ephemeris evaluations. For instance, by sampling a two–dimensional search space into k cells in each dimension only $2k$ ephemeris evaluations are required and only k^2 Lambert's problems need to be solved. Furthermore, the search space can be remarkably reduced (i.e. pruned) by approaching the problem as a cascade of two-dimensional sub-problems: inequality constraints are evaluated over the grid characterising each sub-problem, and unfeasible regions are propagated forward and backward and pruned from the whole search space. It is worth noting that this technique cannot be applied to any global optimization problem, but it has been developed exploiting the structure of MGA problems.

Unfortunately, the class of MGA transfers so formulated do not cover all the possible trajectories for a chemical propelled spacecraft. An important feature to take into account is the possibility to perform deep space maneuvers (DSM) that are usually carried out to improve the performances of a

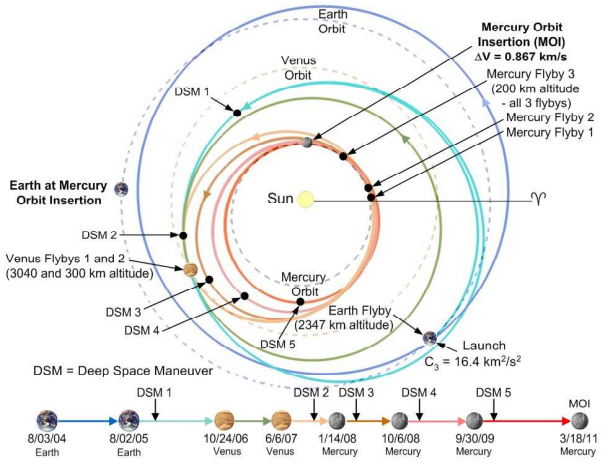


Figure 1.1: The Messenger MGADSM trajectory.

particular trajectory. When DSM are included into a MGA transfer, the resulting trajectory is usually referred to as MGADSM.

The importance of MGADSM is illustrated with an example. Figure 1.1 shows the nominal transfer trajectory for the NASA's Messenger spacecraft that is intended to insert into an orbit around Mercury in 2011 after flying-by the Earth (one time), Venus (two times), and Mercury itself (three times). Finding a feasible solution to such a problem is one of the most challenging tasks in astrodynamics. Probably, the solution would not have been possible if several DSM had not been included. As clearly shown in Figure 1.1, the nominal solution involves five DSM to be performed during the whole transfer. The aim of each DSM is the correction of the trajectory to get an optimal approach with the body to be encountered. The idea is that the gain given by this optimal flyby is worth the propellant spent in the maneuver. It is clear that the introduction of DSM, for chemical propelled spacecraft, gives some degrees of freedom that can be exploited not only to perform better gravity assists, but also to have encounters that would not be possible otherwise. A problem so formulated would cover all the possible trajectories for a chemical propelled spacecraft. Furthermore, a MGADSM trajectory can be applied as first guess solution for the design of low-thrust gravity assist transfers, where the thrust arcs can spread the impulsive velocity changes given by the DSM. In any case, if the solution space of a MGADSM problem could be pruned, the reward would be enormous. It would be possible to define a trajectory having an arbitrary number of ballistic arcs, deep space maneuvers, and low-thrust arcs (if exponential sinusoids [22] are included in

the formulation of the problem).

Nevertheless, pruning the solution space when deep space maneuvers are included is not a trivial task. First of all, the dimension of the search space increases because the generic trajectory leg is described by four or even five variables if out-of-plane maneuvers are included. Consequently, difficulties are introduced in the maneuver modelling as well as in the definition of suitable bounds for the introduced variables.

Another feature associated to the MGADSM problem is the proliferation of local minima, that makes difficult the detection of big prunable regions. It has been shown that the search space can be pruned in regions far from the local minima where the inequality constraints are not satisfied. In a MGA problem these are large regions that can be pruned in each two–dimensional problem. These regions can become even larger when forward and backward constraining are applied, reducing sensibly the final search space. When MGADSM is considered, the problem has a much greater number of local minima than a simple MGA problem and the prunable regions are expected to shrink.

For all the reasons discussed above, it is necessary to rethink the whole pruning process implemented in GASP, and to reformulate it when deep space maneuvers are included. To handle this problem, the use of differential algebraic (DA) techniques is proposed in this study.

As better described in chapter 2, differential algebra serves the purpose of automatic differentiation, i.e. the accurate computation of the derivatives of functions in a computer environment. This goal is actually achieved by replacing the classical implementation of the real algebra with the proper implementation of a new algebra based on Taylor polynomials. Given a generic function f of v variables, whose evaluation involves only algebraic operations (including transcendental functions, inversion, derivation and integration), the Taylor expansion of f up to any desired order n can be easily obtained from a computer algorithm that implements its evaluation. The related derivatives are computed with the accuracy of the floating point representation of the corresponding real numbers in the computer environment.

The main idea behind the introduction of DA techniques into the pruning process is then substituting the pointwise evaluation, typical of GASP, with the computation of the Taylor expansions of the objective and constraint functions with respect to the design variables, around suitably selected reference points of the search space. The Taylor expansions are used to approximate the functions over proper domains. In particular, simple boxes are used, and polynomial bounders are exploited to estimate the ranges of the computed functions within each box.

Consequently, using DA techniques, the pointwise approach proposed in

GASP can be substituted by a sampling process relying on box samples. Thanks to the Taylor approximation, the computation of the tolerance related to the Lipschitzian constant of the GASP method is avoided by using suitable bounders of the functions and the corresponding Taylor polynomials on the considered domain boxes. Furthermore, as previously mentioned, the order of the Taylor expansion can be exploited to tune the accuracy of the approximation and the size of the grid boxes. This might result in the possibility of enlarging the grid for the domain discretization, and reducing considerably the computational burden.

Further interesting information can be drawn from DA computation. Suppose that, after the pruning process, the box bounds that possibly enclose the global optimum are identified. An optimization process within the identified domain is necessary to identify the solution [21]. Savings in computational time might be achieved by exploiting the Taylor representation of the objective function over this box. In particular, if the Taylor representation is sufficiently accurate, the further optimization process can be faster processed: the evaluation of the objective function can be substituted by a fast computation of polynomials, so exploiting the advantages of metamodel based global optimization algorithms [24].

Based on the previous observations, the first part of the report is devoted to carefully describe the introduction of DA techniques into the classical GASP algorithm, so dealing with MGA transfers without DSM. In particular, a brief survey on the theory of differential algebra is presented in chapter 2. Chapter 3 concentrates on the difficulties arising from the introduction of differential algebra into GASP. The effort spent to overcome the discontinuity problem and to expand the objective and constraint functions in Taylor series is deeply examined. The result of this process is the definition of the DA–based GASP algorithm, which will be referred to as GASP–DA in the followings. The performances of GASP–DA are then assessed by presenting the main results of an extensive test phase, relying on suitable MGA transfer test cases.

The major goal of the work is addressed in chapter 4, which is dedicated to the development of an effective pruning algorithm for MGA transfers involving DSM. The selection of the most appropriate modeling formulation is the core of this chapter, as it turned out to strongly affect the performances of the pruning process. The resulting algorithm, referred to as GASP–DSM–DA, is then submitted to a significant test phase, whose results conclude the chapter.

However, the work has not been confined to the achievement of the originally planned scopes. First of all, an alternative strategy to solve the ultimate goal of optimizing MGA transfers including DSM has been developed. The

strategy is detailed in chapter 5: after a MGA space pruning process, suitable heuristics are implemented to decide for the introduction of DSM and to define first guess solutions for subsequent optimization processes, aimed at characterizing the resulting MGA-DSM transfers. A comparison analysis of the alternative strategy with GASP–DSM–DA is reported.

Moreover, in the effort of paving the way and promoting the use of validated global optimization tools in space–related applications, the validated global optimization of two-impulse transfers is addressed in chapter 6. Taylor models [26] allow the designer to obtain validated enclosures of the objective function over a box on the search space, and the tool COSY–GO [15] can be used to get bounds of the global optimum.

Final remarks and suggestions for future developments conclude the report.

Chapter 2

Notes on Differential Algebra

The theory of differential algebra presented in this chapter has been developed by Martin Berz in the late 80s, and the short summary given in the following takes advantage of his book *Modern Map Methods in Particle Beam Physics* [14].

Differential algebraic (DA) techniques find their origin in the attempt to solve analytical problem by an algebraic approach. The DA techniques introduced by M. Berz addressed the solution of differential equations and partial differential equations, more specifically the efficient determination of Taylor expansions of the flow of differential equations in terms of initial conditions.

Historically, treatment of functions in numerics has been based on the treatment of numbers, and the classical numerical algorithms are based on the mere evaluation of functions at specific points. DA techniques are based on the observation that it is possible to extract more information on a function rather than its mere values. In particular, the Taylor coefficients of a function can be obtained up to a specified order n, along with the function evaluation, with a fixed amount of effort. The Taylor coefficients of order n for sums and product of functions, as well as scalar products with reals, can be computed from those of summands and factors; therefore, the set of equivalence classes of functions can be endowed with well-defined operations, leading to the so-called truncated power series algebra (TPSA) [3, 4].

Similarly to the algorithms for floating point arithmetic, the algorithm for functions followed, including methods to perform composition of functions, to invert them, to solve nonlinear systems explicitly, and to treat common elementary functions [7, 13]. In addition to these algebraic operations, also the analytic operations of differentiation and integration have been developed on these function spaces, defining a differential algebraic structure.

As DA represents the core of the algorithms developed in the frame of this

contract, some useful notes to get familiar with these techniques are given in the following. In particular the minimal differential algebra is explained in details, and some hints on its extension to m variables and to n-th order are then given. The chapter ends with the description of the solution of implicit parametric equations, necessary to formulate a generic MGA-DSM transfer problem.

2.1 The Minimal Differential Algebra

The simplest nontrivial differential algebra is here described. Consider all ordered pairs (q_0, q_1), with q_0 and q_1 real numbers. The addition, scalar multiplication, and vector multiplication are defined as follows:

$$\begin{aligned} (q_0, q_1) + (r_0, r_1) &= (q_0 + r_0, q_1 + r_1) \\ t \cdot (q_0, q_1) &= (t \cdot q_0, t \cdot q_1) \\ (q_0, q_1) \cdot (r_0, r_1) &= (q_0 \cdot r_0, q_0 \cdot r_1 + q_1 \cdot r_0). \end{aligned} \tag{2.1}$$

The ordered pairs with the arithmetic are called ${}_1D_1$. The first two operations are the familiar vector space structure of R^2, whereas the multiplication is similar to that in the complex numbers; except here $(0,1) \cdot (0,1)$ does not equal $(-1, 0)$, but rather $(0, 0)$. The multiplication of vectors is seen to have $(1, 0)$ as the unity element. The multiplication is commutative, associative, and distributive with respect to addition. Together, the three operations defined in (2.1) form an algebra. Furthermore, they do form an extension of real numbers, as $(r, 0) + (s, 0) = (r + s, 0)$ and $(r, 0) \cdot (s, 0) = (r \cdot s, 0)$, so that the reals can be included.

However ${}_1D_1$ is not a field, as (q_0, q_1) has a multiplicative inverse in ${}_1D_1$ if and only if $q_0 \neq 0$. If $q_0 \neq 0$ then

$$(q_0, q_1)^{-1} = \left(\frac{1}{q_0}, -\frac{q_1}{q_0^2} \right). \tag{2.2}$$

If q_0 is positive, then $(q_0, q_1) \in {}_1D_1$ has a root

$$\sqrt{(q_0, q_1)} = \left(\sqrt{q_0}, \frac{q_1}{2\sqrt{q_0}} \right), \tag{2.3}$$

as simple arithmetic shows.

One important property of this algebra is that it has an order compatible with its algebraic operations. Given two elements (q_0, q_1) and (r_0, r_1) in ${}_1D_1$,

it is defined

$$\begin{aligned}
&(q_0, q_1) < (r_0, r_1) \quad \text{if} \quad q_0 < r_0 \quad \text{or} \quad (q_0 = r_0 \text{ and } q_1 < r_1) \\
&(q_0, q_1) > (r_0, r_1) \quad \text{if} \quad (r_0, r_1) < (q_0, q_1) \\
&(q_0, q_1) = (r_0, r_1) \quad \text{if} \quad q_0 = r_0 \text{ and } q_1 = r_1.
\end{aligned} \tag{2.4}$$

As for any two elements (q_0, q_1) and (r_0, r_1) only one of the three relation holds, ${}_1D_1$ is said totally ordered. The order is compatible with the addition and multiplication; for all $(q_0, q_1), (r_0, r_1), (s_0, s_1) \in {}_1D_1$, it follows $(q_0, q_1) < (r_0, r_1) \Rightarrow (q_0, q_1) + (s_0, s_1) < (r_0, r_1) + (s_0, s_1)$; and $(s_0, s_1) > (0, 0) = 0 \Rightarrow (q_0, q_1) \cdot (s_0, s_1) < (r_0, r_1) \cdot (s_0, s_1)$.

The number $d = (0, 1)$ has the interesting property of being positive but smaller than any positive real number; indeed

$$(0, 0) < (0, 1) < (r, 0) = r. \tag{2.5}$$

For this reason d is called an infinitesimal or a differential. In fact, d is so small that its square vanishes. Since for any $(q_0, q_1) \in {}_1D_1$

$$(q_0, q_1) = (q_0, 0) + (0, q_1) = q_0 + d \cdot q_1, \tag{2.6}$$

the first component is called the real part and the second component the differential part.

The algebra in ${}_1D_1$ becomes a differential algebra by introducing a map ∂ from ${}_1D_1$ to itself, and proving that the map is a derivation. Define $\partial : {}_1D_1 \to {}_1D_1$ by

$$\partial(q_0, q_1) = (0, q_1). \tag{2.7}$$

Note that

$$\begin{aligned}
\partial\{(q_0, q_1) + (r_0, r_1)\} &= \partial(q_0 + r_0, q_1 + r_1) = (0, q_1 + r_1) \\
&= (0, q_1) + (0, r_1) = \partial(q_0, q_1) + \partial(r_0, r_1)
\end{aligned} \tag{2.8}$$

and

$$\begin{aligned}
\partial\{(q_0, q_1) \cdot (r_0, r_1)\} &= \partial(q_0 \cdot r_0, q_0 \cdot r_1 + r_0 \cdot q_1) = (0, q_0 \cdot r_1 + r_0 \cdot q_1) \\
&= (0, q_1) \cdot (r_0, r_1) + (0, r_1) \cdot (q_0, q_1) \\
&= \partial\{(q_0, q_1)\} \cdot (r_0, r_1) + (q_0, q_1) \cdot \partial\{(r_0, r_1)\}.
\end{aligned} \tag{2.9}$$

This holds for all $(q_0, q_1), (r_0, r_1) \in {}_1D_1$. Therefore ∂ is a derivation and $({}_1D_1, \partial)$ is a differential algebra.

The most important aspect of ${}_1D_1$ is that it allows the automatic computation of derivatives. Assuming to have two functions f and g and to put their values and their derivatives at the origin in the form $(f(0), f'(0))$ and $(g(0), g'(0))$ as two vectors in ${}_1D_1$, if the derivative of the product $f \cdot g$ is of interest, it has just to be looked at the second component of the product $(f(0), f'(0)) \cdot (g(0), g'(0))$; whereas the first component gives the value of the product of the functions. Therefore, if two vectors contain the values and the derivatives of two functions, their product contains the values and the derivatives of the product function. Defining the operation [ ] from the space of differential functions to ${}_1D_1$ via

$$[f] = (f(0), f'(0)), \tag{2.10}$$

it holds

$$\begin{aligned} [f+g] &= [f] + [g] \\ [f \cdot g] &= [f] \cdot [g] \end{aligned} \tag{2.11}$$

and

$$[1/g] = [1]/[g] = 1/[g] \tag{2.12}$$

by using 2.2. This observation can be used to compute derivatives of many kinds of functions algebraically by merely applying arithmetic rules on ${}_1D_1$, beginning from the value and the derivative of the identity function. Consider the example

$$f(x) = \frac{1}{x + \frac{1}{x}} \tag{2.13}$$

and its derivative

$$f'(x) = \frac{(1/x^2) - 1}{(x + (1/x))^2}. \tag{2.14}$$

The function value and its derivative at the point $x = 3$ are

$$f(3) = \frac{3}{10}, \quad f'(3) = -\frac{2}{25}. \tag{2.15}$$

If the function 2.13 is evaluated at the identity function $[x] = (x, 1)$ at the point 3, i.e. $(3, 1) = 3 + d$, it results

$$\begin{aligned} f((3,1)) &= \frac{1}{(3,1) + 1/(3,1)} = \frac{1}{(3,1) + (1/3, -1/9)} \\ &= \frac{1}{(10/3, 8/9)} = \left(\frac{3}{10}, -\frac{8}{9} \Big/ \frac{100}{9}\right) = \left(\frac{3}{10}, -\frac{2}{25}\right). \end{aligned} \tag{2.16}$$

As it can be seen after the evaluation of the function, the real part of the result is the value of the function at $x = 3$, whereas the differential part is the value of the derivative of the function at $x = 3$. This is simply justified by applying the relations 2.11 and 2.12

$$\begin{aligned} [f(x)] &= \left[\frac{1}{x+1/x}\right] = \frac{1}{[x+1/x]} \\ &= \frac{1}{[x]+[1/x]} = \frac{1}{[x]+1/[x]} \\ &= f([x]). \end{aligned} \tag{2.17}$$

Since, for a real x, $[x] = (x, 1) = x + d$, and $[f(x)] = (f(x), f'(x))$, apparently

$$(f(3), f'(3)) = f((3 + d)). \tag{2.18}$$

The method can be generalized to allow the treatment of common intrinsic functions, like sin, exp, by setting

$$\begin{aligned} g_i([f]) &= [g_i(f)] \quad \text{or} \\ g_i((q_0, q_1)) &= (g_i(q_0), q_1 g_i'(q_0)). \end{aligned} \tag{2.19}$$

By virtue of equations (2.1) and (2.19) any function f representable by finitely many additions, subtractions, multiplications, divisions, and intrinsic functions in ${}_1D_1$ satisfies the important relationship

$$[f(x)] = f([x]). \tag{2.20}$$

Note that $f(r+d) = f(r) + d \cdot f'(r)$ resembles $f(x+\Delta x) \approx f(x) + \Delta x \cdot f'(x)$, in which the approximation becomes increasingly more refined for smaller Δx. Here, as the Δx is infinitely small, the error turns out to be zero.

The differential algebra ${}_1D_1$ allows to compute the first derivative of every function f along with the function evaluation. This has an important consequence when the numerical integration of an ordinary differential equation is performed by means of an arbitrary integration scheme. Any integration scheme is based on algebraic operations, involving the evaluations of the ODE right hand side at several integration points; therefore the ${}_1D_1$ algebra can be exploited to compute the first order expansion of the flow of the ODE. As an example consider the first order ordinary differential equation

$$\begin{cases} \dot{x} = f(x) \\ x(0) = x_0 \end{cases} \tag{2.21}$$

and the simple first order forward Euler's scheme

$$x_k = x_{k-1} + \Delta t \cdot f(x_{k-1}). \tag{2.22}$$

If the initial value x_0 is substituted by $[x_0] = (x_0, 1) = x_0 + \delta x_0 \in {}_1D_1$ and the iteration scheme is evaluated using ${}_1D_1$ algebra, the output of the numerical integration at the k-th step is $(x_k, \partial x_k/\partial x_0)$.

By extending the algebra ${}_1D_1$ to order n and v variables, the expansions of the flow of a dynamical systems can be computed up to order n with fixed amount of effort.

2.2 The Differential Algebra ${}_nD_v$

The algebra described in this section was introduced to compute the derivatives up to an order n of functions in v variables. Similarly as before, it is based on taking the space $C^n(R^v)$, the collections of n times continuously differentiable functions on R^v. On this space an equivalence relation is introduced. For f and $g \in C^n(R^v)$, $f =_n g$ if and only if $f(0) = g(0)$ and all the partial derivatives of f and g agree at 0 up to order n. The relation $=_n$ satisfies

$$\begin{aligned} f =_n f \quad &\text{for all} \quad f \in C^n(R^v), \\ f =_n g \Rightarrow g =_n f \quad &\text{for all} \quad f, g \in C^n(R^v), \quad \text{and} \\ f =_n g \text{ and } g =_n h \Rightarrow f =_n h \quad &\text{for all} \quad f, g, h, \in C^n(R^v). \end{aligned} \tag{2.23}$$

Thus, $=_n$ is an equivalence relation. All the elements that are related to f can be grouped together in one set, the equivalence class $[f]$ of the function f. The resulting equivalence classes are often referred to as DA vectors or DA numbers. Intuitively, each of these classes is then specified by a particular collection of partial derivatives in all v variables up to order n. This class is called ${}_nD_v$.

If the values and the derivatives of two functions f and g are known, the corresponding values and derivatives of $f + g$ and $f \cdot g$ can be inferred. Therefore, the arithmetics on the classes in ${}_nD_v$ can be introduced via

$$[f] + [g] = [f + g] \tag{2.24}$$

$$t \cdot [g] = [t \cdot f] \tag{2.25}$$

$$[f] \cdot [g] = [f \cdot g]. \tag{2.26}$$

Under this operations, ${}_nD_v$ becomes an algebra. For each $k \in 1, \ldots, v$, define the map ∂_k from ${}_nD_v$ to ${}_nD_v$ for f via

$$\partial_k[f] = \left[p_k \cdot \frac{\partial f}{\partial x_k} \right], \tag{2.27}$$

where

$$p_k(x_1, \ldots, x_k) = x_k \tag{2.28}$$

projects out the k-th component of the identity function. It's easy to show that for all $k = 1, \ldots, v$ and for all $[f]$, $[g] \in {}_nD_v$

$$\partial_k([f] + [g]) = \partial_k[f] + \partial_k[g] \tag{2.29}$$

$$\partial_k([f] \cdot [g]) = [f] \cdot (\partial_k[g]) + (\partial_k[f]) \cdot [g]. \tag{2.30}$$

Therefore, ∂_k is a derivation for all k, and hence $({}_nD_v, \partial_1, \ldots, \partial_k)$ is a differential algebra.

The dimension of ${}_nD_v$ is now assessed. Define the special numbers d_k as follow:

$$d_k = [x_k]. \tag{2.31}$$

Observe that f lies in the same class as its Taylor polynomial T_f of order n around the origin; they have the same function values and derivatives up to order n. Therefore,

$$[f] = [T_f]. \tag{2.32}$$

Denoting the Taylor coefficients of the Taylor polynomial T_f of f as $c_{j_1,\ldots,j_v}$, it follows

$$T_f(x_1, \ldots, x_v) = \sum_{j_1+\cdots+j_v \leq n} c_{j_1,\ldots,j_v} \cdot x_1^{j_1} \cdots x_v^{j_v} \tag{2.33}$$

with

$$c_{j_i}, \ldots, c_{j_v} = \frac{1}{j_1! \cdots j_v!} \cdot \frac{\partial^{j_1+\cdots+j_v} f}{\partial x_1^{j_1} \cdots \partial x_v^{j_v}} \tag{2.34}$$

and thus

$$\begin{aligned} [f] = [T_f] &= \left[\sum_{j_1+\cdots+j_v \leq n} c_{j_1,\ldots,j_v} \cdot x_1^{j_1} \cdots x_v^{j_v} \right] \\ &= \sum_{j_1+\cdots+j_v \leq n} c_{j_1,\ldots,j_v} \cdot d_1^{j_1} \cdots d_v^{j_v}, \end{aligned} \tag{2.35}$$

where, in the last step, the properties $[a+b]=[a]+[b]$ and $[a\cdot b]=[a]\cdot[b]$ have been used. Therefore, the set $\{1, d_k : k=1,2,\ldots,v\}$ generates ${}_nD_v$, as any element of ${}_nD_v$ can be obtained from 1 and the d_k's via addition and multiplication. Therefore, as an algebra, ${}_nD_v$ has $(v+1)$ generators, and the terms $d_1^{j_1}\cdots d_v^{j_v}$ form a basis for the vector space ${}_nD_v$. It is shown in [14] that the number of basic elements is $(n+v)!/(n!\,v!)$, which is the dimension on ${}_nD_v$.

Similar to the structure ${}_1D_1$, ${}_nD_v$ can be ordered, and the d_k, being smaller than any real number, are infinitely small or infinitesimal. Furthermore, by applying the fixed point theorem for contracting operators on $M\subset {}_nD_v$ that map M into M, the square root, the quotient, and map inversion in ${}_nD_v$ can be obtained by iteration in a finite number of steps. Once the function composition and the elementary functions, i.e. exp, sin, and log, are introduced in the DA ${}_nD_v$, the derivatives of any function f belonging to $C^n(R^v)$ can be computed up to order n in fixed amount of effort by applying

$$[f(x_1,\ldots,x_v)]=f([x_1,\ldots,x_v])=f(x_1+d_1,\ldots,x_v+d_v). \tag{2.36}$$

The DA sketched in this section was implemented by M. Berz and K. Makino in the software COSY INFINITY. The software and all the related documentations are available free of charge for non-commercial use online at
http : //bt.pa.msu.edu/index_cosy.htm.

2.3 Solution of Parametric Implicit Equations

As it will be explained in the next chapter, the formulation of a MGA transfer optimization problem (either with or without DSM) requires the solution of parametric implicit equations. This is indeed necessary whenever we want to solve the Kepler's equation, the Lagrange's equation, and the bending angle equation for the ephemerides evaluation, the Lambert's problem and the powered gravity assist problem solution, respectively.

The algorithm developed for the solution of a scalar parametric implicit equation using DA is described; the generalization to the m dimensional case is straightforward.

We are searching for the solution of

$$f(x,p)=0 \tag{2.37}$$

for $p\in[p_l,p_u]$ with $f\in\mathcal{C}^{n+1}$. This means that $x=x(p)$ satisfying

$$f(x(p),p)=0 \tag{2.38}$$

is sought. The first step is to consider a point value of the parameter p^0 and to compute the point value of the solution x^0 by means of a classical numerical method, e.g by Newton's method. The variable x and the parameter p are then initialized as n-th order DA variables, i.e.

$$\begin{aligned} [x] &= x^0 + \delta x \\ [p] &= p^0 + \delta p, \end{aligned} \tag{2.39}$$

and the implicit equation (2.37) is expanded up to the n-th order, delivering the map

$$\delta f = \mathcal{M}_f(\delta x, \delta p). \tag{2.40}$$

Note that the map has no constant part as x^0 is the solution of the implicit equation for the nominal value of the parameter p^0. The map (2.40) is then augmented by introducing the identity map $\delta p = \mathcal{I}_p(\delta p)$, ending up with

$$\begin{pmatrix} \delta f \\ \delta p \end{pmatrix} = \begin{pmatrix} \mathcal{M}_f \\ \mathcal{I}_p \end{pmatrix} \begin{pmatrix} \delta x \\ \delta p \end{pmatrix}. \tag{2.41}$$

The n-th order map (2.41) is inverted using COSY INFINITY built-in tools, obtaining

$$\begin{pmatrix} \delta x \\ \delta p \end{pmatrix} = \begin{pmatrix} \mathcal{M}_f \\ \mathcal{I}_p \end{pmatrix}^{-1} \begin{pmatrix} \delta f \\ \delta p \end{pmatrix}. \tag{2.42}$$

As the goal is to compute the n-th order Taylor expansion of the solution manifold $x = x(p)$, the map (2.42) is evaluated for $\delta f = 0$. From the first row we have

$$\delta x = \mathcal{M}_{\delta f=0}(\delta p), \tag{2.43}$$

which is the n-th order Taylor expansion of the solution manifold, i.e.

$$\delta x = \delta x(\delta p). \tag{2.44}$$

For every value of $p \in [p_l, p_u]$ the approximate solution of $f(x, p) = 0$ can be easily computed by evaluating the Taylor polynomial (2.44).

Chapter 3

GASP-DA: Analysis and Implementation

This chapter is devoted to the accurate description of the introduction of the differential algebraic techniques into GASP algorithm. The basic idea is taking advantage of the possibility of expanding the constraint and objective functions with respect to the optimization variables. Unfortunately, this is not an easy task and several problems must be faced. This chapter takes charge of illustrating such problems and their proposed and implemented solutions.

In particular, section 3.1 describes the procedure to obtain the Taylor expansion of the constraint and objective functions by expanding the solution of the involved implicit equations. Section 3.2 highlights the presence of discontinuities on objective functions typically optimized in MGA transfers. Rationales for the presence of such discontinuities are supplied, and an extensive analysis is carried out to introduce the proposed solution to the problem. The mathematical formulation of a typical MGA transfer problem is not unique and a careful selection of the design variables has to be performed. This is the aim of section 3.3, where the reason of the selection of the so-called absolute times formulation is pointed out. Some ideas to further improve the performances of the optimization process are presented in sections 3.4 and 3.5. Finally, section 3.6 assesses its performances on test problems of various complexity.

3.1 DA-evaluation of the objective function

For the sake of a clear description of the topics addressed in this section, and without loss of generality, consider the classical problem of transferring

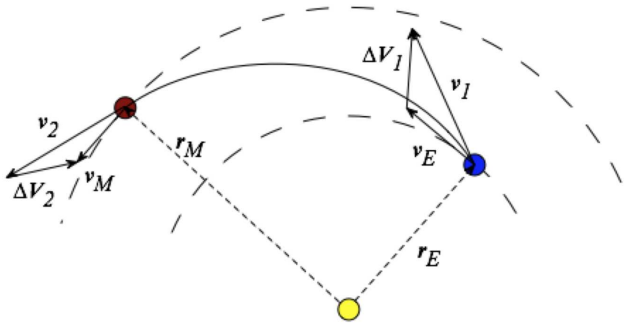


Figure 3.1: An Earth–Mars transfer.

a spacecraft from Earth to Mars by means of two impulsive maneuvers (see Figure 3.1). The typical objective function for this problem is the overall amount of ΔV that must be supplied to the spacecraft to carry out the transfer. Two design variables suffices to this aim. In particular, a possible and classical choice is selecting the departure epoch from Earth, T_E, and the time of flight from Earth to Mars, t_{EM}. Given T_E and t_{EM}, the arrival epoch at Mars, T_M, can be easily computed, and the position and velocity of Earth and Mars at the beginning and the end of the transfer are obtained through ephemerides. Then, given the initial and final positions, and the time of flight, the corresponding Lambert's problem is solved to compute the heliocentric initial velocity, $\mathbf{v}_1$, the spacecraft must be supplied with at Earth in order to reach Mars in the given time of flight, as well as the resulting heliocentric velocity at Mars, $\mathbf{v}_2$. The initial relative velocity of the spacecraft with respect to Earth, $\Delta\boldsymbol{V}_1$, and the final relative velocity with respect to Mars, $\Delta\boldsymbol{V}_2$, are readably assessed. A typical optimization problem for this simple planet-to-planet transfer involves the minimization of the overall ΔV, which is computed as the sum of the magnitude of the relative velocities at the beginning and the end of the transfer:

$$\Delta V = \|\Delta\boldsymbol{V}_1\| + \|\Delta\boldsymbol{V}_2\| = \Delta V_1 + \Delta V_2, \tag{3.1}$$

subject to usual constraints on the maximum allowed magnitudes of the relative velocities at the beginning and the end of the transfer:

$$\begin{aligned} \Delta V_1 &\leq \Delta V_{1,\max} \\ \Delta V_2 &\leq \Delta V_{2,\max} \end{aligned} \tag{3.2}$$

As already pointed out in chapter 2, behind the introduction of differential algebraic techniques is the substitution of the pointwise evaluation of

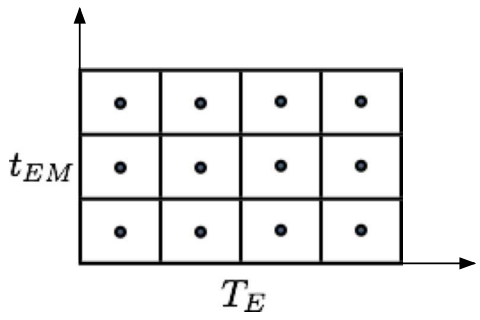


Figure 3.2: Search space sampling for the Earth–Mars transfer.

the objective function with a DA–based evaluation. Consequently, instead of dealing with classical point values of the objective function, the DA-based GASP algorithm handles its Taylor expansions around suitably selected reference points of the search space. Referring to Figure 3.2, the classical pointwise sampling of the search space implemented in GASP is here replaced by a subdivision of the whole search space in boxes. Then, differential algebraic techniques allows to compute the Taylor expansion of the constraint and objective functions within each box around a reference point, e.g. the center of the box. The Taylor expansions are bounded to estimate the range of the constraint and objective functions within the box. The range is then used in the pruning process. Before detailing the pruning algorithm for this simple planet-to-planet transfer, it is worth observing that Taylor expanding the constraint and objective functions, even for this relatively simple problem, does not lie in a mere DA-based evaluation. The solution of implicit equations is involved in the computation, which poses some additional problems.

3.1.1 Parametric implicit equations

Three implicit equations appear in the evaluation process of constraint and objective functions for typical MGA transfers. Two of them can be identified by analyzing again the previously introduced Earth–Mars transfer. The evaluation of the ephemerides of the involved planets is first required. An analytical ephemeris model is used within this work, which is based on third order polynomial fits of the orbital elements of the planets (see Figure 3.3). The fitting procedure is performed on a set of accurate values, delivered by JPL ephemeris evaluations. In particular, the analytical model is then able to supply the eccentricity of the planet orbit, e, and the mean anomaly of the planet, M, as a function of the input epoch. At this point, the Kepler's

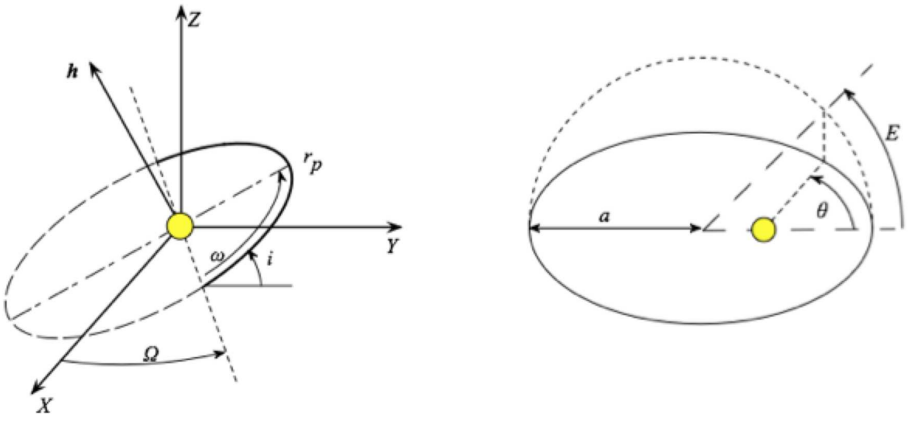


Figure 3.3: Orbital elements.

equation

$$f(E) = E - e\sin E - M = 0 \tag{3.3}$$

must be solved. This is a first implicit equation, which allows to compute the eccentric anomaly, E, from e and M. Therefore, the resulting E is used to assess the planet position and velocity by mere algebraic relations.

Given the positions of Earth and Mars, and the time of flight between the two planets, the Lambert's problem must be solved to gain the initial and final heliocentric velocities of the spacecraft. Several algorithms are available in literature, which tackle this issue from different perspectives, and offer solutions based on different numerical techniques. An unpublished algorithm developed by Izzo is used within this work, which is freely available to download in MATLAB and C++ format from the website: http://www.esa.int/gsp/ACT/inf/op/globopt.htm. The algorithm involves the solution of the so-called Lagrange's equation for the time of flight, that concisely reads

$$f(x) = A(x) - t = 0 \tag{3.4}$$

where x is related to the semi–major axis of the resulting transfer orbit, A is a function depending on x and some geometrical properties of the transfer, and t is the transfer time between the involved planets. The solution of this second implicit equation allows to compute the initial and final heliocentric velocities of the spacecraft, by means of algebraic relations and the evaluation of transcendental functions.

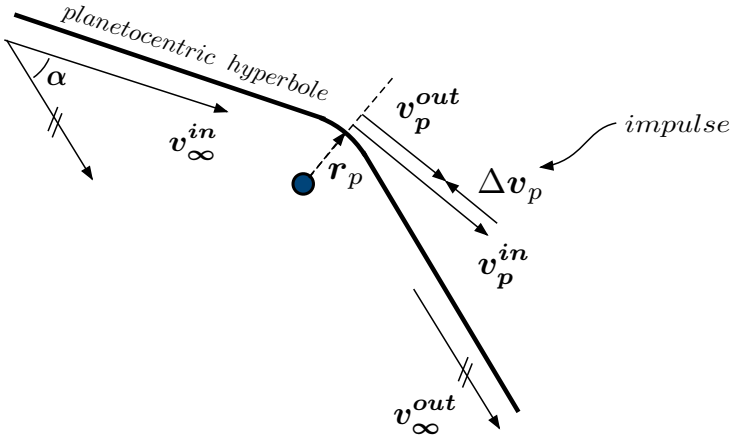


Figure 3.4: Powered gravity assist.

The third implicit equation can be identified only within interplanetary transfers involving one gravity assist at least. As highlighted in [21], in the patched-conics approximation, two subsequent heliocentric ellipses must be matched using the gravity assist maneuver. This means the pericenter radius of the hyperbolic planetocentric trajectory linking the two heliocentric arcs must be evaluated. To this aim, the classical powered gravity assist maneuver is implemented: the spacecraft is allowed to provide an impulse at the pericenter of the incoming hyperbola, tangential to the trajectory (see Figure 3.4). The planetocentric trajectory is therefore made by two arcs of hyperbola patched together. Within this model, the angle α, usually referred to as bending angle, between the incoming and the outgoing velocities, $\boldsymbol{v}_\infty^{in}$ and $\boldsymbol{v}_\infty^{out}$ respectively, is related to the pericenter radius via

$$f(r_p) = \arcsin\frac{a^-}{a^- + r_p} + \arcsin\frac{a^+}{a^+ + r_p} - \alpha = 0 \tag{3.5}$$

where $a^- = 1/(\boldsymbol{v}_\infty^{in} \cdot \boldsymbol{v}_\infty^{in})$ and $a^+ = 1/(\boldsymbol{v}_\infty^{out} \cdot \boldsymbol{v}_\infty^{out})$. Given the two heliocentric arcs to be connected by the powered gravity assist maneuver, the angle α can be easily computed through geometrical relations. The solution of the third implicit equation (3.5) delivers the pericenter radius of the planetocentric trajectory. The planetocentric velocities $\boldsymbol{v}_p^{in}$ and $\boldsymbol{v}_p^{out}$ at the pericenter, corresponding to the incoming and outgoing hyperbolic arcs respectively, are evaluated using $\boldsymbol{r}_p$, $\boldsymbol{v}_\infty^{in}$, and $\boldsymbol{v}_\infty^{out}$. Then, the required impulsive maneuver at the pericenter, $\Delta\boldsymbol{v}_p$, is the mere difference between $\boldsymbol{v}_p^{out}$ and $\boldsymbol{v}_p^{in}$.

If a pointwise evaluation of the objective and constraint functions is of interest, as in GASP algorithm, a classical numerical method for the solution

of implicit equations can be used, e.g. the Newton's method. Unfortunately, the previous implicit equations become parametric implicit equations when the Taylor expansion of the objective and constraint functions is of interest. Without loss of generality, consider the Kepler's equation. In particular, referring to the previous Earth–Mars transfer, the objective function evaluation process involves the computation of the planetary ephemerides of Earth and Mars. If the Taylor expansion of the objective function with respect to the design variables is needed, the expansion of the position and velocity of the planets has manifestly to be achieved. The Taylor expansion of the planetary orbital elements with respect to the epoch can be easily obtained by initializing the epoch as a DA variable

$$[T] = T_0 + \delta T = (T_0, 1), \tag{3.6}$$

where T_0 is a point reference epoch, and performing a DA–based evaluation of the analytical ephemeris model. Thus, the Taylor expansions $\mathcal{T}$ of the eccentricity and the mean anomaly with respect to the epoch are readily available:

$$\begin{aligned} e(\delta T) &= \mathcal{T}_e(\delta T) \\ M(\delta T) &= \mathcal{T}_M(\delta T). \end{aligned} \tag{3.7}$$

The next step is solving the Kepler's equation to compute the corresponding eccentric anomaly E. However, interest is not in a mere point value of E in this case, but rather in the Taylor expansion of the solution E with respect to the parameter T. Indeed, the explicit dependence of e and M on T must be kept and Kepler's equation reads

$$f(E, \delta T) = E - e(\delta T) \sin E - M(\delta T) = 0. \tag{3.8}$$

The solution of this parametric implicit equation is attained in terms of the Taylor expansion $E(\delta T) = \mathcal{T}_E(\delta T)$ using techniques illustrated in chapter 2. Once $E(\delta T)$ is available, the Taylor expansions of the planet position and velocity are readily obtained by carrying out the remaining algebraic manipulations in the DA framework. Clearly, the accuracy of the expansion depends on the order of the DA computation as well as on the size of the interval on the epoch, i.e. on δT. Figure 3.5 and Figure 3.6 study this accuracy referring to Mars ephemerides. In particular, the reference epoch 1456 MJD is selected, which corresponds to the arrival date of Mars Express. The Taylor expansion of Mars position and velocity around the reference epoch is computed using differential algebra. Considering an interval of 40 days around the reference epoch, for each δT, the position and velocity of Mars

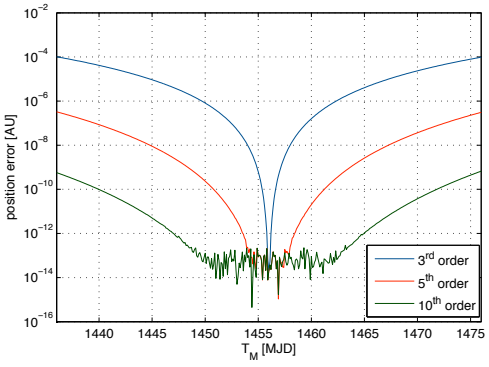


Figure 3.5: Accuracy of the Taylor expansion of the planet position.

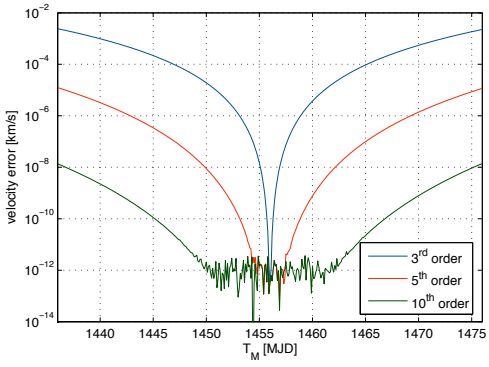


Figure 3.6: Accuracy of the Taylor expansion of the planet velocity.

are evaluated using both the Taylor expansions and the pointwise evaluations. Figure 3.5 and Figure 3.6 report the error of the Taylor expansions with respect to the pointwise evaluation, in terms of the maximum norm of the difference vectors between the corresponding positions and velocities, respectively. The figures clearly show that, although the accuracy of the Taylor expansion decreases while moving away from the reference date, it can be effectively kept to a suitable level increasing the expansion order. It is worth mentioning that the fast wiggling of the curve in the vicinity of the reference epoch is due to the tolerance set for the Newton method, which is used in the classical pointwise ephemeris evaluation.

Similar statements hold for the solution of the Lagrange's equation (3.4) and for the bending angle equation (3.5). In particular, the ephemeris evaluation and the solution of the Lambert's problem, allow to compute the overall ΔV for the simple Earth-Mars transfer. For the sake of a more complete analysis of the accuracy of the DA computations, Figure 3.8 illustrates an instance of the error of the Taylor expansion of the ΔV with respect to its point evaluation. The reference epoch 1249 MJD for the departure from Earth and the reference transfer time of 207 days are selected, which correspond to the data of Mars Express. An interval of 40 days is considered for each variable, representing a box centered on the previous reference point in the search space. Similarly to the previous analysis, the overall ΔV is evaluated by a pointwise computation as well as using a 10th order DA computation. The error is then estimated as the absolute difference. The figure shows that the 10th order DA computation is sufficiently accurate over the entire box. Figure 3.7 reports the objective function values of both the point and the DA computation: the two corresponding surfaces overlap, confirm-

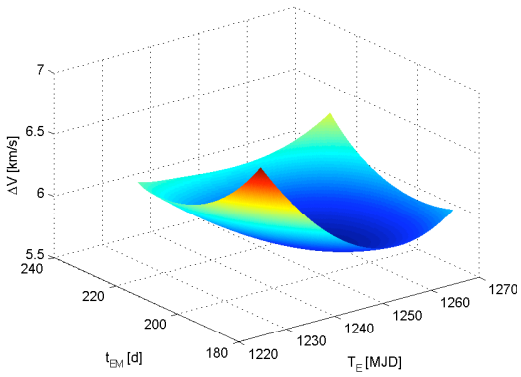


Figure 3.7: ΔV for the Earth–Mars transfer: comparison between pointwise and DA evaluation.

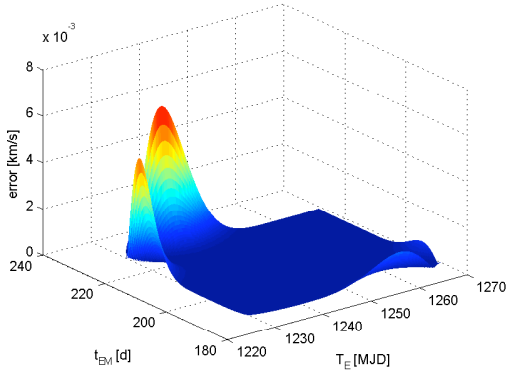


Figure 3.8: ΔV for the Earth–Mars transfer: absolute difference between pointwise and DA evaluation.

ing the great accuracy of the DA computation. Finally, as indicative of the computational time, it is worth reporting that the 10th order DA evaluation of the objective function requires about 0.016 seconds on a Pentium IV 3.06 GHz laptop.

3.2 The discontinuity problem

The previous section was devoted to describe how the Taylor expansion of the constraint and objective functions with respect to the design variables can be obtained using differential algebraic techniques. The complete extension of GASP algorithm should now be possible in a straightforward manner. This would be the case for pruning and optimization problems where regular constraint and objective functions are involved. Unfortunately, significant discontinuities characterize these functions in typical MGA transfer optimization problems, which are mainly related to geometrical considerations.

In order to introduce the discontinuity problem, consider the following typical pruning algorithm for the representative Earth-Mars transfer:

1. *Subdivide the search space in boxes (refer to Figure 3.2).*
2. *For each box* $[\vec{\boldsymbol{X}}] = \{[T_E], [t_{EM}]\}$*:*
    i. *initialize* $[T_E]$ *and* $[t_{EM}]$ *as DA variables and compute the Taylor expansion of* ΔV_1 *(see Figure 3.1) on* $[\vec{\boldsymbol{X}}]$*;*

ii. *bound the polynomial expansion of* ΔV_1 *on* $[\vec{\boldsymbol{X}}]$*, i.e. estimate its minimum* $\underline{\Delta V_1}$ *and maximum* $\overline{\Delta V_1}$ *on* $[\vec{\boldsymbol{X}}]$*;*

iii. *if* $\underline{\Delta V_1} > \Delta V_{1,max} \Rightarrow$ *discard the current box* $[\vec{\boldsymbol{X}}]$ *and analyze the next in the list;*

iv. *compute the Taylor expansion of* ΔV_2 *(see Figure 3.1) on* $[\vec{\boldsymbol{X}}]$*;*

v. *bound the polynomial expansion of* ΔV_2 *on* $[\vec{\boldsymbol{X}}]$*, i.e. estimate its minimum* $\underline{\Delta V_2}$ *and maximum* $\overline{\Delta V_2}$ *on* $[\vec{\boldsymbol{X}}]$*;*

vi. *if* $\underline{\Delta V_2} > \Delta V_{2,max} \Rightarrow$ *discard the current box* $[\vec{\boldsymbol{X}}]$ *and analyze the next in the list;*

vii. *keep* $[\vec{\boldsymbol{X}}]$ *in the list.*

It is worth mentioning that bounding the Taylor expansions, as required in steps 2.ii and 2.v of the previous algorithm, is not a trivial task. Although a non-validate quadratic estimation process is suggested in section 3.5, the basic tool used within this work, and for the example reported in this section, is the Linear Dominated Bounder described by Makino in the Ph.D. dissertation [26].

The previous algorithm has been implemented in COSY–Infinity and applied to the Earth–Mars transfer. In particular, as summarized in Table 3.1, a search space of 5000 days on the departure epoch and 500 days on the transfer time is selected. Figure 3.9 reports the overall ΔV over the defined search space. Quasi-periodicities can be identified on the figure, especially on the departure date, which can be easily related to the synodic period of the Earth–Mars system. In fact, based on the mean planetary orbital radius and in the hypothesis of coplanar circular orbits, the synodical period of the Earth-Mars system is assessed to be about 2.14 years, which can be clearly identified as the period of the oscillations in Figure 3.9. After one synodical period, the relative geometrical configuration of Earth and Mars

variable	lower bound	upper bound	amplitude	units
T_E	1000	6000	50	MJD
t_{EM}	100	600	50	MJD

Table 3.1: Bounds and box-size for the design variables of the Earth–Mars transfer.

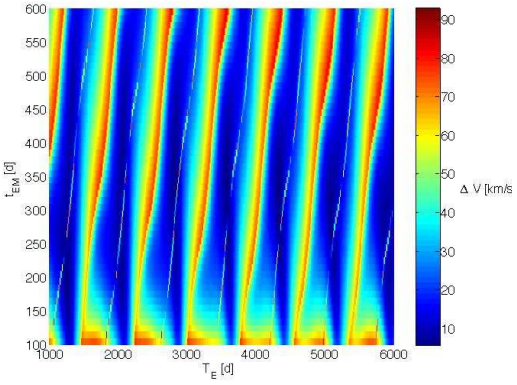


Figure 3.9: ΔV for the Earth–Mars transfer.

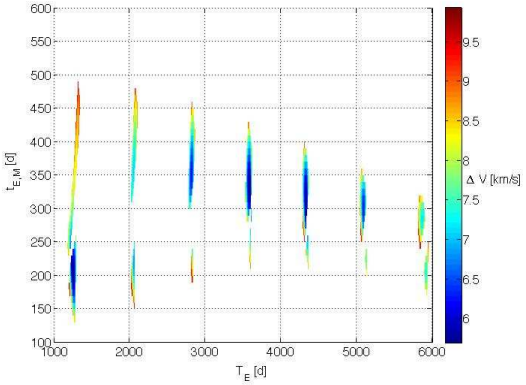


Figure 3.10: ΔV for the Earth–Mars transfer after pruning.

approximately recurs, so that the objective function values repeat, leading to a quasi-periodic function. Figure 3.10 reports instead the search space remaining after inferring the following pruning constraints on ΔV_1 and ΔV_2:

$$\begin{aligned} \Delta V_1 &\leq 5 \text{ km/s} \\ \Delta V_2 &\leq 5 \text{ km/s}. \end{aligned} \tag{3.9}$$

Let now the pruning algorithm illustrated above be applied to this relatively simple problem. In particular, as reported in the third column of Table 3.1, the search space is uniformly subdivided in boxes, using 50×50 days boxes on the two design variables. The pruning process is then performed using the constraints (3.9). The boxes remaining after pruning are reported in Figure 3.11 on the pruned search space of Figure 3.10, which is aimed to be enclosed by the algorithm. Figure 3.11 clearly shows that the accuracy of the attained enclosure is not satisfactory. Specifically, even if the remaining boxes enclose the desired portion of the search space, some boxes remain after pruning, which should have been pruned away. To better achieve rationales on this behavior, Figure 3.12 reports the same remaining boxes on the objective function plot over the overall search space. As can be clearly seen, these unsought remaining boxes tend to lie on lines over the search space, which can be related to well known discontinuities of this problem. Such discontinuities correspond to the so-called transitions from the "short–way" to the "long–way" solutions of the Lambert's problem, and viceversa (see Figure 3.13). A geometrical overview of the problem is reported in Figure 3.14 for a sample transition from the short–way to the long–way solution. Transfer trajectories for a planet-to-planet transfer are plotted, moving from the left

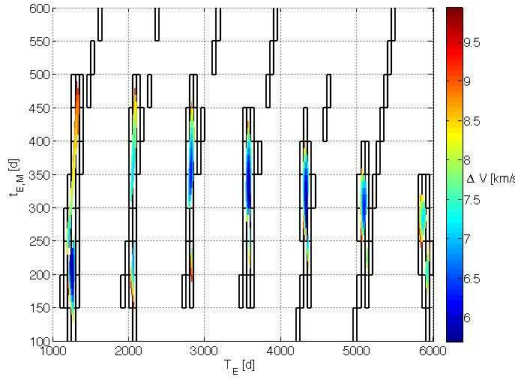


Figure 3.11: Enclosure of the pruned search space for the Earth–Mars transfer.

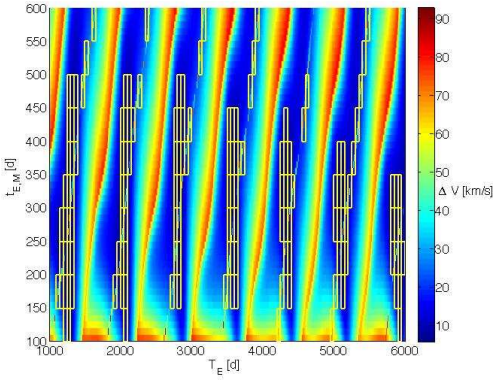


Figure 3.12: Boxes of Figure 3.11 reported on the overall search space of the Earth–Mars transfer.

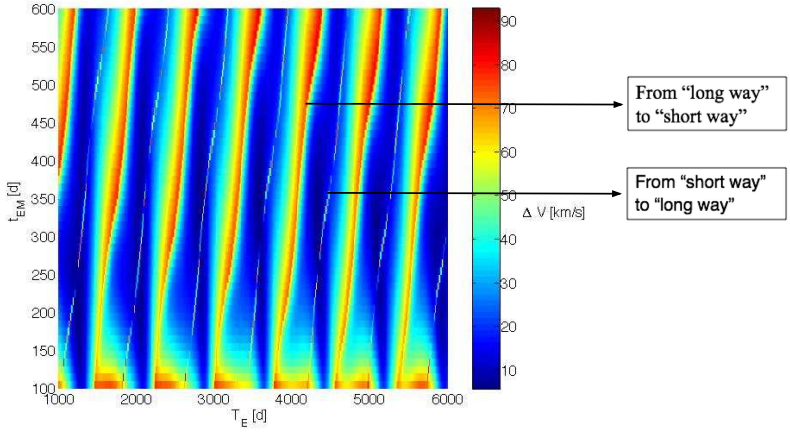


Figure 3.13: Discontinuities on the ΔV for the Earth–Mars transfer.

side of the discontinuity to the right side. On the left side of the discontinuity the short–way solutions are selected by the Lambert's solver. Moving toward the right side, the orbital plane inclination of the transfer trajectories tends to increase. The discontinuity occurs when the transfer trajectory is exactly perpendicular to the ecliptic. Just after the occurrence of the discontinuity, in order to keep dealing with prograde solutions of the Lambert's problem, the long–way solution is suddenly selected. Corresponding to the previous transition, a plot of the overall ΔV with respect to the departure epoch is reported in Figure 3.15: ΔV goes up close to the discontinuity, where the

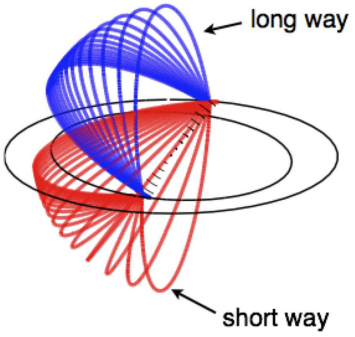


Figure 3.14: Geometrical overview of the transition from the short–way to the long–way solution.

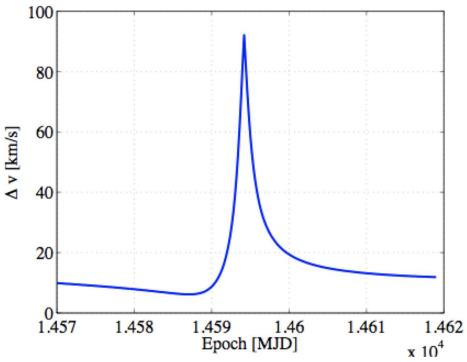


Figure 3.15: ΔV w.r.t. T_E: transition from the short–way to the long–way solution.

difference between the inclinations of the planetary orbital planes and of the transfer trajectory increases; a small discontinuity occur exactly at the pick, which can not be detected on the figure. Well known theoretical arguments show that Taylor polynomial expansions fail when discontinuities on the processed function occur. This can be deemed the cause of the presence of undesired boxes after pruning: Taylor expansions within boxes lying on the discontinuity do not accurately approximate constraint functions; consequently bounds of the corresponding ranges are wrongly estimated, and the boxes tend to be kept in the list of admissible solutions. Intensive work has been devoted to overcome the discontinuity problem, and to improve the accuracy of the enclosure of the pruned search space. A first attempt was based on the use of box-reshaping and box-splitting techniques, which are briefly illustrated in section 3.2.1. The actual solution finally implemented in GASP–DA is instead illustrated in section 3.2.2.

3.2.1 Box-reshaping and box-splitting

A first attempt to solve the discontinuity problem is based on the observation that the discontinuity lines tend to follow a straight path on the search space (see Figure 3.16). In particular, it can be easily assessed that, if Earth and Mars followed circular and coplanar orbits, the discontinuity lines would be exactly straight line, with a common slope α, which could be related to the planetary orbital periods by

$$\tan\alpha = \frac{P_M}{P_E} - 1, \tag{3.10}$$

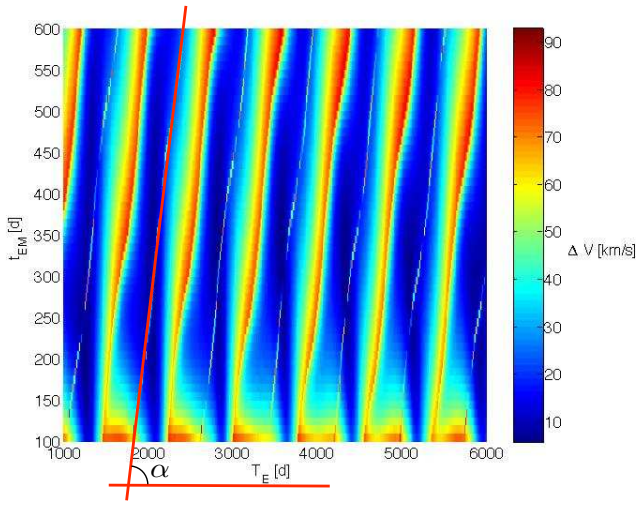


Figure 3.16: The approximate slope of the discontinuities.

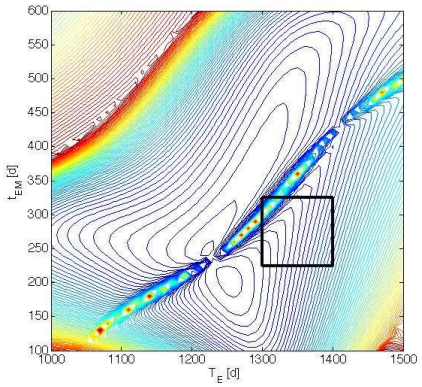


Figure 3.17: Sample box lying on the discontinuity before reshaping.

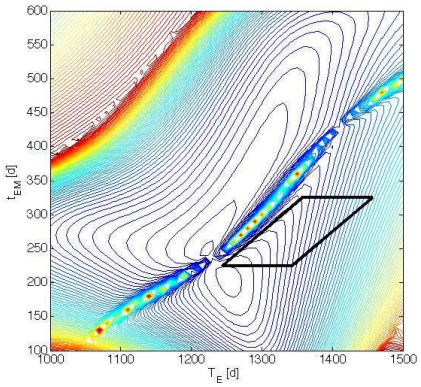


Figure 3.18: Sample box after reshaping.

where P_M and P_E are the orbital periods of Mars and Earth, respectively.

Based on the previous observation, boxes could be suitably reshaped in order to reduce the number of boxes lying on the discontinuity, remaining after the pruning process. Referring to Figure 3.17, suppose the reported box lying on the discontinuity is being processed. As already pointed out, the box is represented by the vector of DA numbers

$$[\vec{\boldsymbol{X}}] = \{[T_E], [t_{EM}]\} = \{T_E + \delta T_E, t_{EM} + \delta t_{EM}\}. \tag{3.11}$$

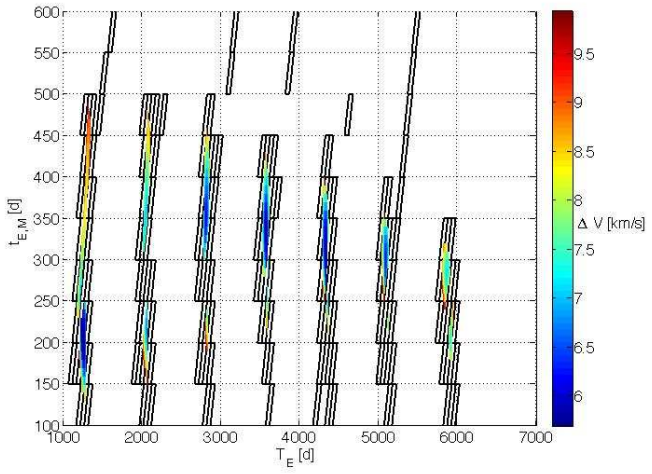


Figure 3.19: Enclosure of the pruned search space for the Earth–Mars transfer using reshaped boxes.

Using the approximate slope α of equation (3.10), the new box

$$[\vec{\boldsymbol{X}}] = \{T_E + \delta T_E + (1/\tan\alpha) \cdot \delta t_{EM}, t_{EM} + \delta t_{EM}\} \tag{3.12}$$

can be gleaned out, which matches up the reshaping procedure depicted in Figure 3.18: two sides of the original box are made approximately parallel to the discontinuity line. As highlighted in the figure, the sample box would not lie on the discontinuity after reshaping, so avoiding the problem of the associated Taylor expansions.

As could be easily objected, the expedient of box-reshaping alone can not solve the discontinuity problem, as boxes lying on the discontinuity line still occur even after reshaping. This is illustrated in Figure 3.19, where the pruning algorithm is applied to the previous Earth–Mars transfer problem using 50×50 days reshaped boxes: even if the number of boxes lying on the discontinuities decreases if compared with those of Figure 3.11, the accuracy of the enclosure of the pruned search space is still inadequate.

A box-splitting process has then been added to the previous reshaping technique, which is schematically presented in Figure 3.20 and 3.21. Suppose that, after reshaping, the box reported in Figure 3.20 is being processed. The algorithm for box-splitting is based on the following steps:

1. Moving on a horizontal line, passing through the center of the box (red line), identify a point lying on the discontinuity.

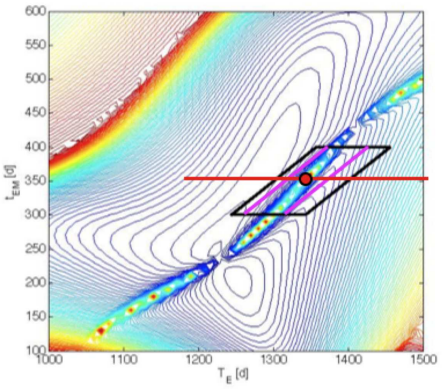


Figure 3.20: Identification of the splitting lines.

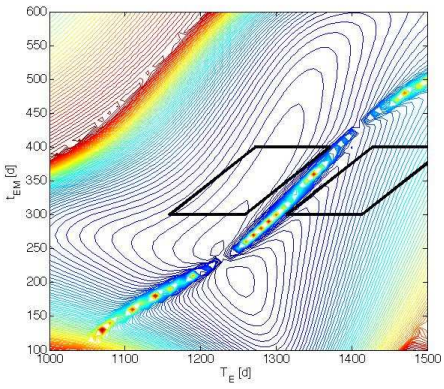


Figure 3.21: Reshaped boxes after the splitting procedure.

2. Enclose the discontinuity within a strip (magenta lines).

3. Identify two discontinuity-free boxes (see Figure 3.21).

4. Replace the original box with the two identified boxes.

Unfortunately, despite the simplicity and the theoretical effectiveness of this technique, some critical issues can be identified.

First of all, for the sake of computational time containment, the identification of the point lying on the discontinuity involved in step 1 is performed by processing the Taylor expansions of the related geometrical quantities. However, as already pointed out, Taylor expansions do not accurately represent such quantities within these regions. Consequently, the identification of the desired point turns out not to be accurate enough for the splitting purposes.

Secondly, the definition of the enclosing strip of step 2 requires some heuristics for the suitable assessment of the width of the region to be excluded. Evidently, the corresponding tolerances depend on the planetary system under study, since the deviation of the discontinuity lines from the approximating straight line strongly depends on the planetary orbital elements. A sharp enclosure of the discontinuity is however necessary, since good local minima and the global optimum lie close to the discontinuities from the short–way to the long–way solution of the Lambert's problem in this planet-to-planet transfer. Consequently, good regions are often thrown away because of bad estimates of the tolerances.

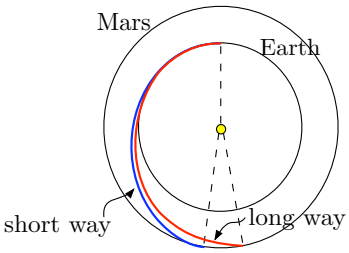


Figure 3.22: The discontinuity from the short–way to the long–way solution disappears in a planar model.

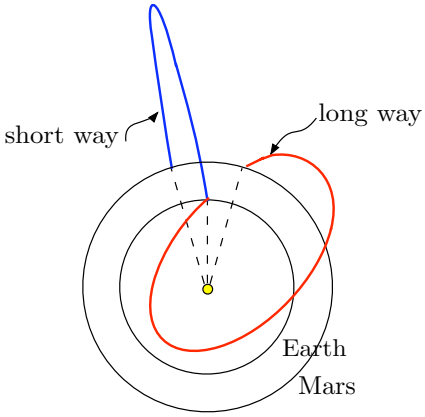


Figure 3.23: The discontinuity from the long–way to the short–way solution remains in a planar model.

Finally, a computational time increase follows. Besides the additional operation required by the previous algorithm, an evident cause can be detected in step 4. Indeed, every box lying on the discontinuities is replaced by two boxes, both of which must be processed again.

The previous considerations led to the decision of adopting the alternative strategy to solve the problem, which is discussed in next section.

3.2.2 Planar planetary model

The implemented solution for the discontinuity problem is based on the observation that the unfavorable discontinuity lines, i.e. the lines close to good local minima, correspond to the transition from the short-way to the long-way solution of the Lambert's problem. The previous discontinuity do not occur if a planar planetary model is used instead of the actual three-dimensional model associated to the ephemeris evaluator. This can be easily recognized by analyzing again Figure 3.14. The discontinuity is related to the ambiguity on the inclination of the orbital plane, when a perpendicular transfer occurs. In this situation, the definition of prograde and retrograde transfers is singular, and the inclination of the corresponding transfer orbit is characterized by sign ambiguity, i.e. ± 90 deg. The previous ambiguity vanishes if a planar planetary model is used: the orbital plane of the connecting Lambert's arc is uniquely determined as coinciding with the ecliptic, and the transition from the short–way to the long–way solution is continuous. It is worth observing that this is not the case for the transition from the long–way

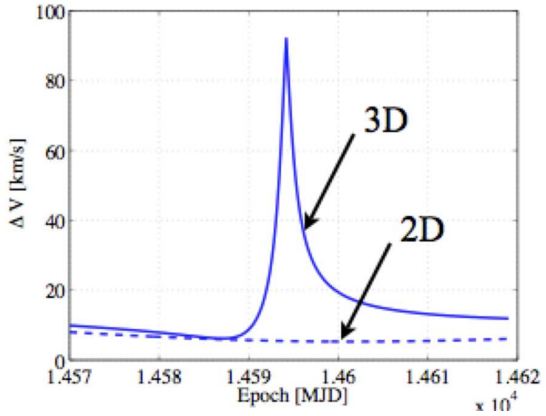


Figure 3.24: ΔV w.r.t. T_E: transition from the short–way to the long–way solution. Comparison between the three-dimensional and the two-dimensional planetary models.

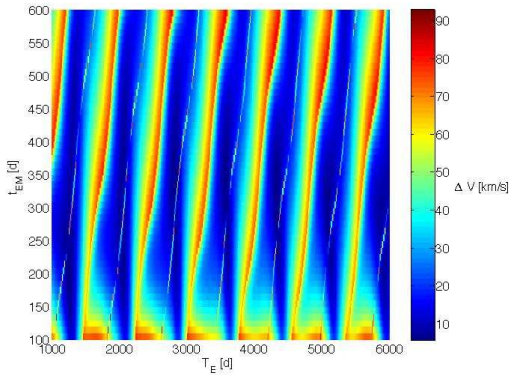


Figure 3.25: ΔV for the Earth–Mars transfer in the three-dimensional planetary model.

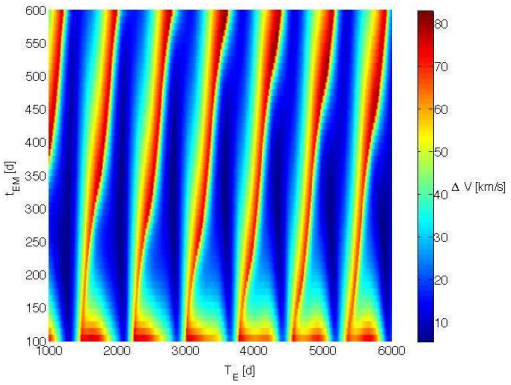


Figure 3.26: ΔV for the Earth–Mars transfer in the two-dimensional planetary model.

to the short–way solution (see Figure 3.22 and Figure 3.23): the ambiguity on the transfer plane vanishes, but a geometrical discontinuity remains. The disappearance of the first discontinuity is clearly confirmed in Figure 3.24, where the overall ΔV reported in Figure 3.15 is compared with the same plot in case the planar planetary model is used. Figures 3.25 and 3.26 address a similar comparison, by plotting the ΔV over the whole search space in the two planetary models. The discontinuities corresponding to the transition from the short-way to the long-way solutions disappear, whereas the other

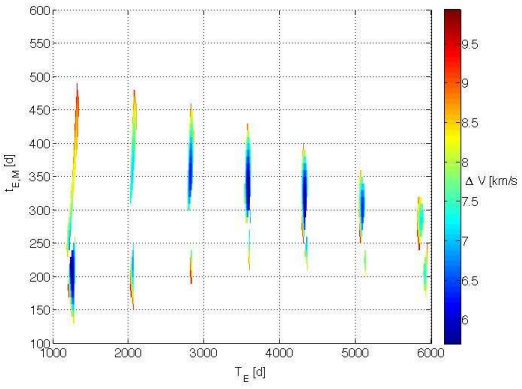


Figure 3.27: Pruned search space for the Earth–Mars transfer in the three-dimensional planetary model.

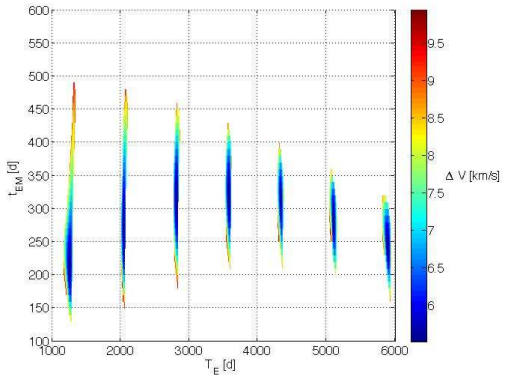


Figure 3.28: Pruned search space for the Earth–Mars transfer in the two-dimensional planetary model.

discontinuities still occur.

A major observation can be stated, which is the main driver for the following decisions. A careful analysis of the difference between the ΔV in the three-dimensional, ΔV_{3D}, and the two-dimensional, ΔV_{2D}, planetary models over the entire search space has been performed. It turns out that the error $\Delta V_{err} = \Delta V_{3D} - \Delta V_{2D} \geq 0$ everywhere, i.e.

$$\Delta V_{3D} \geq \Delta V_{2D} \tag{3.13}$$

on the whole search space. This is confirmed by the comparison analysis in Figure 3.24, and the ΔV_{3D} and ΔV_{2D} values in Figure 3.25 and 3.26. The main consequence of relation (3.13) is that, given the same constraints on ΔV_1 and ΔV_2, the pruned search space in the two–dimensional planetary model encloses the pruned search space in the three–dimensional planetary model. This is illustrated in Figure 3.27 and Figure 3.28, where the pruned search spaces corresponding to the two planetary models are compared. Consequently, if the pruning process is performed in the two-dimensional planetary model, no branches of the feasible domain in the real three-dimensional planetary model are lost.

Given the previous considerations, a planar planetary model is adopted in the DA-based GASP algorithm to perform the pruning process. No mathematical proof is supplied about the validity of this conservative hypothesis for a general transfer. The low inclination of all planetary orbits, and geometrical considerations, led to the decision of conjecturing its validity for interplanetary transfers in the Solar System. Although more rigorous mathematical considerations should be sought in future works, the fairness of the

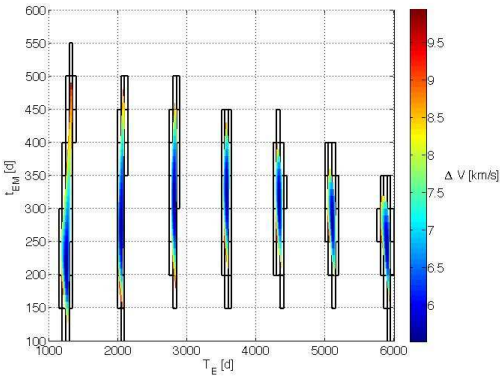


Figure 3.29: GASP–DA on the two-dimensional model.

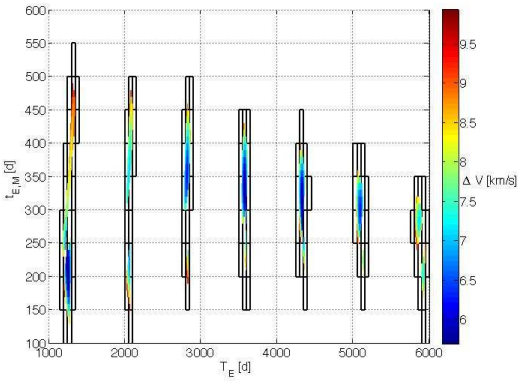


Figure 3.30: GASP–DA on the three-dimensional model.

hypothesis has been confirmed by the test phase illustrated in section 3.6: all the best-known solutions of typical MGA transfer problems have been identified to lie in the boxes remaining after the pruning process. It is worth anticipating that the previous approximation is only used within the pruning process, whereas the subsequent necessary optimization process is performed within the actual three-dimensional planetary model. As a further proof of the validity of the approximation, the performances of the pruning algorithm for the case of the Earth–Mars transfer in the two–dimensional planetary model are analyzed in Figure 3.29 and Figure 3.30. The boxes remaining after the pruning process sharply enclose the pruned search space of both the two-dimensional and the three-dimensional models. A plain improvement in the enclosure accuracy can be detected in the three-dimensional model by comparing Figure 3.30 with Figure 3.11.

3.3 The dependency problem

The previous considerations are based on analyses performed within the framework of a planet-to-planet transfer, i.e. an Earth-Mars transfer, where the departure epoch from Earth and the transfer time from Earth to Mars are selected as design variables. However, an alternative problem formulation is preferable. In particular, substituting the arrival epoch at Mars to the transfer time from Earth to Mars in the set of the design variables has already shown important advantages from a computational point of view [21]. The use of this second formulation turns out to significantly reduce the number of ephemeris evaluations required by the pruning algorithm, allowing the

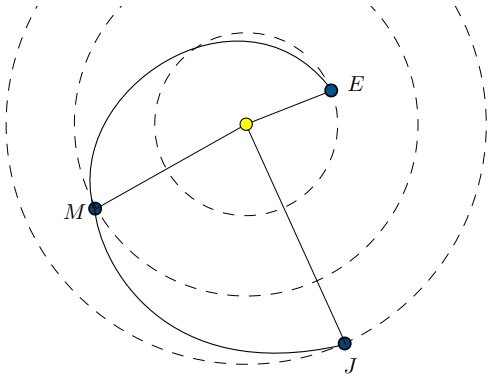


Figure 3.31: Scheme of an Earth–Mars–Jupiter transfer.

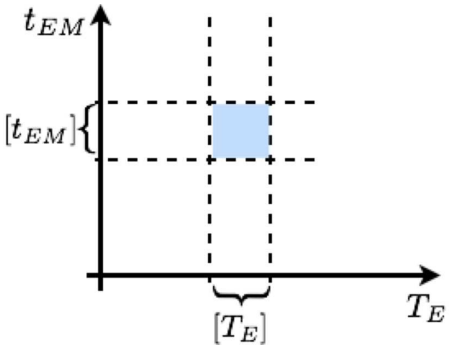


Figure 3.32: Design space for the first arc of the Earth–Mars–Jupiter transfer.

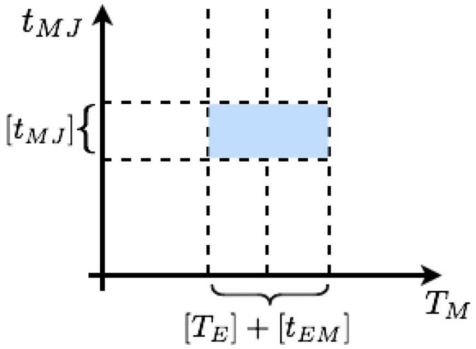


Figure 3.33: Design space for the second arc of the Earth–Mars–Jupiter transfer.

whole process to gain a polynomial complexity. Further reasons of selecting this second formulation can be outlined, which are important alike for the DA-based GASP, especially if actual MGA transfers are studied.

Without loss of generality, consider the scheme of an Earth–Mars–Jupiter transfer, reported in Figure 3.31. The set of design variables usually selected for this MGA transfer is composed of the departure epoch from Earth, T_E, the transfer time from Earth to Mars, t_{EM}, and the transfer time from Mars to Jupiter, t_{MJ}. For the sake of clarity, this formulation is referred to as "relative times formulation " in the following. The evaluation of the overall ΔV will generally start from the analysis of the first connecting arc from Earth to Mars. Suppose the relative times formulation is being used. Thus, referring to Figure 3.32, the quantities related to the first arc are characterized in the $T_E - t_{EM}$ plane. As both T_E and t_{EM} are design variables, in the framework

of the DA-based GASP, a box is readily identified by the DA representation of both variables, and the computation goes on as depicted in the previous sections. Suppose now the first arc has been processed, and the second arc from Mars to Jupiter is of interest. The quantities associated to the second arc are characterized in the $T_M - t_{MJ}$ plane, where T_M is the arrival epoch at Mars. However, T_M is not a design variable in the relative times formulation, and it is computed as $T_M = T_E + t_{EM}$. Even if t_{MJ} is a design variable, the size of the corresponding interval on T_M is the sum of the box size on T_E and t_{EM} in the DA-based GASP. The previous considerations can be easily inferred to MGA transfers involving more than three planets: the box size of the departure epoch from each planet increases along the transfer. This effect is strongly related to the dependencies associated to the relative times formulation: the last departure epoch of an MGA transfer depends on all the design variables associated to the previous arcs, and the box size increases accordingly.

The dependency problem is better highlighted in Figure 3.34. Focusing on the dependence of the planetary ephemerides on the design variables, the position of Earth, $\mathbf{r}_E$, depends only on the departure epoch T_E. The position of Mars, $\mathbf{r}_M$, is evaluated using the epoch at Mars T_M. Consequently, $\mathbf{r}_M$, will depend on the two variables T_E and t_{EM}. Similarly, the dependence of the position of Jupiter on the three variables T_E, T_M, and t_{EJ}, is highlighted. Thus, in a MGA transfer involving n planets, the position of the i-th planet will depend on the departure epoch from Earth, and all the transfer times associated to the prior $i-1$ connecting arcs. Therefore, the dimensionality of the dependency increases along the transfer, reaching its maximum corresponding to the last connecting arc, where quantities, e.g. the planet position, will depend on all n variables. Similar arguments hold for the associated ΔV, on which inequality constraints are usually set.

The previous dependency problems can be overcome if the alternative strategy suggested by Izzo et al. is used [21]. In particular, the departure epoch from Earth is kept within the set of design variables, whereas the transfer times are now replaced by the epoch at each remaining planet of the MGA transfer. Referring again to the Earth–Mars–Jupiter transfer, the new set of variables will include the epochs at Earth, T_E, Mars, T_M, and Jupiter, T_J. In contrast to the relative times formulation, the new formulation is referred to as the "absolute times formulation" in the following.

A review of the previous analyses will be of help to gain a valuable insight on the advantages of the new formulation. Consider again Figure 3.32 and Figure 3.33. Using the absolute times formulation allows both arcs to be characterized within planes that are directly defined by design variables. Thus, no box size increase occurs along the transfer. Referring instead to Fig-

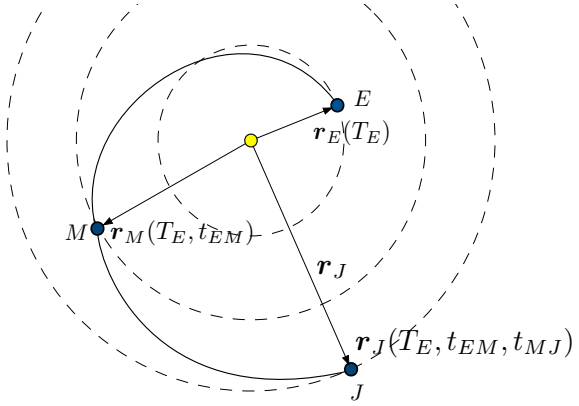


Figure 3.34: Dependency problem for the Earth–Mars–Jupiter transfer.

ure 3.34, the planetary ephemerides will depend on the epoch of the planet, which is now included in the set of design variables. If the ΔV associated to the whole transfer are of interest (which is the case in the pruning process of GASP), it can be easily shown that the maximum dimensionality of the dependency correspond to the ΔV of the powered gravity assist maneuvers, which will depend on three design variables.

The outcomes of the previous analysis led to the decision of adopting the absolute time formulation as the baseline approach in GASP-DA. For the sake of completeness, Figure 3.35 and Figure 3.36 report both the search space and the overall ΔV of the Earth–Mars transfer, which have been investigated in Figure 3.9 and Figure 3.10, on the new plane $T_E - T_M$.

3.4 Semi-analytical approximation

In the effort to further improve the algorithm performances in terms of computational time required by the pruning process, semi-analytical approaches to solve the implicit equations analyzed in section 3.1 have been investigated. The main idea is avoiding the iterative methods usually adopted for the solution of the implicit equations, i.e. Newton's method, using either suitable expansions of the equations or clever variable changes. The following sections are devoted to detail the techniques introduced for each equation.

3.4.1 Kepler's equation

For the sake of clarity, Kepler's equation 3.3 is reported again:

$$f(E) = E - e\sin E - M = 0 \tag{3.14}$$

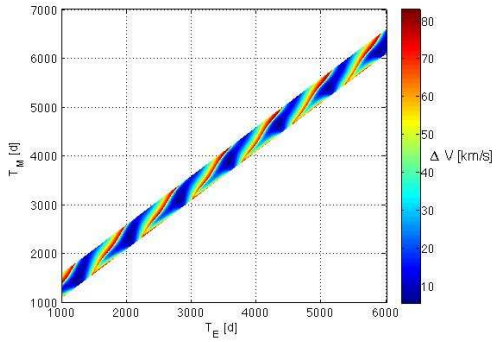


Figure 3.35: ΔV for the Earth–Mars transfer on the overall search space in the $T_E - T_M$ plane.

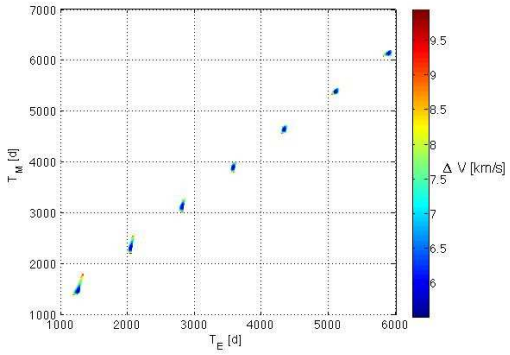


Figure 3.36: ΔV for the Earth–Mars transfer on the pruned search space in the $T_E - T_M$ plane.

This equation relates the planetary eccentricity, e, and mean anomaly, M, which are readily available from the analytical ephemeris model, to the eccentric anomaly E. An extensive analysis of this important equation can be found in literature, and several methods have been developed to solve it. Among all the available methods, the semi-analytical techniques are deemed particularly interesting for the current purpose. Battin [2] illustrates how a useful solution of equation (3.14) can be obtained by expanding it with respect to the eccentricity e. Such kind of solution turns out to be particularly appropriate in the framework of the ephemeris evaluation, since all the planetary orbits in the Solar System are characterized by relatively low eccentricity values. Consequently, a third order expansion of equation (3.14) is used to gain the explicit relation for E

$$E \simeq M + \frac{e \sin M}{1 - e \cos M} - \frac{1}{2}\left(\frac{e \sin M}{1 - e \sin M}\right)^3 . \tag{3.15}$$

The computed eccentric anomaly E is then used to evaluate planetary positions and velocities, through classical algebraic manipulations. Figure 3.37 and Figure 3.38 investigate the error of the semi-analytical approach with respect to the iterative process of a classical Newton's method. In particular, referring to a sample interval epoch of one Martian year, Figure 3.37 reports the difference between the position vectors resulting from the two approaches on the $x-y$ plane. The maximum error is of the order of 1000 km. A similar analysis is addressed in Figure 3.38 for the velocity vector, where a maximum error of the order of 0.1 m/s is detected. This error is acceptable, even though the orbital eccentricity of Mars is not negligible.

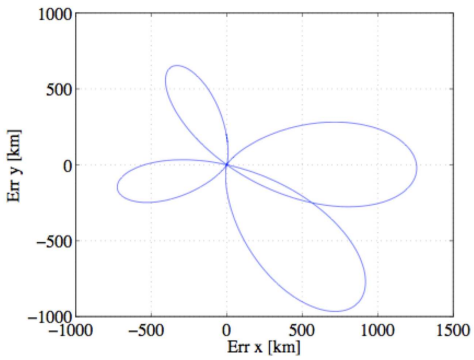


Figure 3.37: Position error of the semi-analytical solution of Kepler's equation w.r.t. Newton's method.

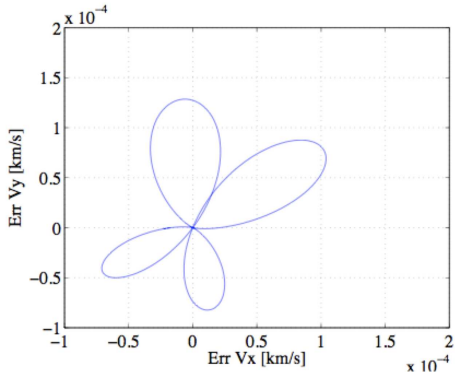


Figure 3.38: Velocity error of the semi-analytical solution of Kepler's equation w.r.t. Newton's method.

3.4.2 Lagrange's equation

The ΔV evaluation requires the solution of the Lambert's problem. As already pointed out in section 3.1, the Lagrange's equation for the time of flight,

$$f(x) = A(x) - t = 0, \tag{3.16}$$

must be solved, where (see [2] for details)

$$A(x) = a(x)^{3/2}((\alpha(x) - \sin(\alpha(x))) - (\beta(x))). \tag{3.17}$$

The functions α and β are related to x via the relations

$$\sin^2 \frac{1}{2}\alpha = \frac{s}{2a} \qquad \sin^2 \frac{1}{2}\beta = \frac{s-c}{2a}, \tag{3.18}$$

where s and c are geometrical parameters defined in [2], and a is the semimajor axis of the resulting conic arc. The semimajor axis is a direct function of x:

$$a = \frac{s}{2(1-x^2)}. \tag{3.19}$$

As suggested by Izzo, equation (3.16) can be suitably manipulated to improve the convergence performances of the iterative algorithms. In particular, the solution of the equation

$$\tilde{f} = \log A(x) - \log t = 0 \tag{3.20}$$

is sough, which replaces the original equation. Moreover, the following variable change is adopted:

$$\tilde{x} = \log(1 + x). \tag{3.21}$$

Using the new variable $\tilde{x}$, the curve associated to equation (3.20) is well approximated by a straight line. Consequently, a first order expansion of the equation can be used to approximate the solution. In particular, the solution is then approximated by

$$\tilde{x} = \tilde{x}_0 - \frac{\tilde{f}(\tilde{x}_0)}{\tilde{f}'(\tilde{x}_0)}, \tag{3.22}$$

where $\tilde{f}'(\tilde{x}_0)$ is computed analytically. The heuristics for the selection of the reference point $\tilde{x}_0$ is based on the identification of a circular orbit, whose angular velocity is equal to the mean angular velocity of the spacecraft on the actual connecting arc. More specifically, the transfer time t, as well as the angle between the initial and final position vectors $\Delta\theta$, are input to the Lambert's solver. The radius R_0 of a circular orbit is sough, which solves the equation

$$\sqrt{\frac{\mu}{R_0^3}}\, t = \Delta\theta. \tag{3.23}$$

The radius R_0 is then used to compute the reference point $\tilde{x}_0$ through the relations (3.19) and (3.21).

Once the solution $\tilde{x}$ of equation (3.20) is obtained, the solution of the original equation (3.16) can be readily computed using again the previous relations. Figure 3.39 reports the overall ΔV for the Earth–Mars transfer, which is compared with the same ΔV resulting from the semi-analytical solutions of both Kepler's and Lagrange's equations, reported in Figure 3.40. Once again, it is worth anticipating that all the semi-analytical approximations here introduced are only used within the pruning phase of the overall optimization process.

3.4.3 Bending angle equation

As described in section 3.1, in the framework of MGA transfers, the planetocentric trajectories at the gravity assists are made by two arcs of hyperbola patched together. Within the powered gravity assist model, the pericenter of the hyperbola is evaluated by solving the implicit equation

$$f(r_p) = \arcsin\frac{a^-}{a^- + r_p} + \arcsin\frac{a^+}{a^+ + r_p} - \alpha = 0 \tag{3.24}$$

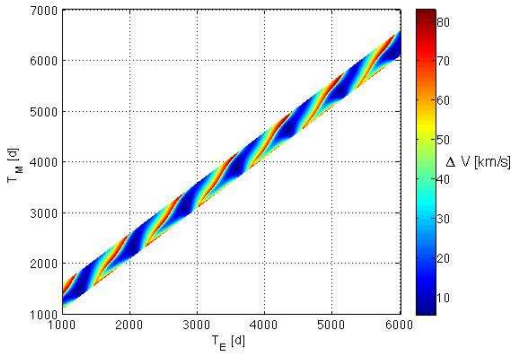


Figure 3.39: ΔV for the Earth–Mars transfer.

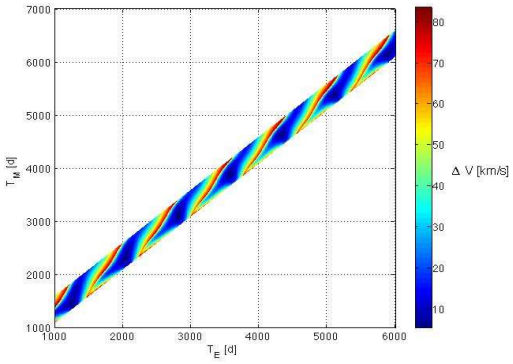


Figure 3.40: ΔV for the Earth–Mars transfer: semi-analytical approach.

where $a^- = 1/(\boldsymbol{v}_\infty^{in} \cdot \boldsymbol{v}_\infty^{in})$ and $a^+ = 1/(\boldsymbol{v}_\infty^{out} \cdot \boldsymbol{v}_\infty^{out})$. Similarly to the previous case, a suitable variable change is looked for to allow equation (3.24) to be well approximated by a linear expansion. The variable change

$$\tilde{x} = 1/r_p \tag{3.25}$$

has been identified to properly serve this purpose. Once again, using the new variable $\tilde{x}$, the first order expansion of the equation is used to approximate the solution by

$$\tilde{x} = \tilde{x}_0 - \frac{f(\tilde{x}_0)}{f'(\tilde{x}_0)}, \tag{3.26}$$

where $f'(\tilde{x}_0)$ is computed analytically. The reference point $\tilde{x}_0$ is chosen using information associated to both incoming and outgoing hyperbola. In particular, the incoming and outgoing hyperbolic arcs are processed separately to compute the corresponding pericenter radii, which make the spacecraft span the whole bending angle α, without any ΔV corrections at the pericenter. The two radii are then averaged to obtain the mean radius

$$r_{p0} = \frac{a^-(b-1) + a^+(b-1)}{2}, \tag{3.27}$$

where $b = 1/\sin(\alpha/2)$. The reference point $\tilde{x}_0$ is then computed through relation 3.25 using the radius r_{p0}. Once the solution $\tilde{x}$ of equation $f(\tilde{x}) = 0$ is available, the actual pericenter radius of the powered gravity assist maneuver can be computed using again equation 3.25.

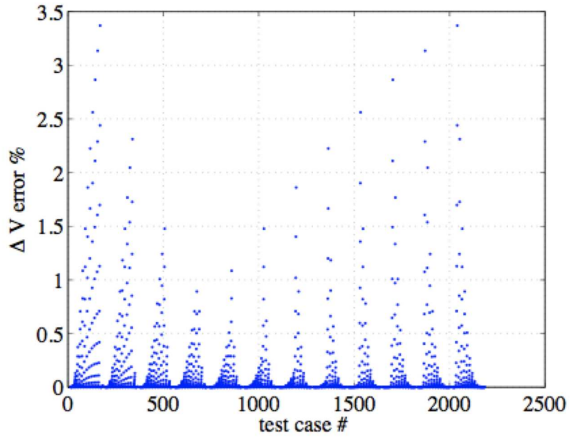


Figure 3.41: Bending angle equation: error of the semi-analytical approximation w.r.t. Newton's method corresponding to random gravity assist samples.

The accuracy of this semi-analytical approximation is investigated in Figure 3.41. Gravity assist samples are generated by randomly selecting values of α, v_∞^{out}, and v_∞^{in} within the ranges

$$\begin{aligned} v_\infty^{out}, v_\infty^{in} &\in [6, 40] \text{ km/s} \\ \alpha &\in [0, 60] \text{ deg.} \end{aligned} \tag{3.28}$$

For each sample, the corrective ΔV within the powered gravity assist model is computed by solving equation (3.24), using both Newton's method and the previous semi-analytical approach. The relative error is then plotted for each sample. The maximum relative error is about 3.5%, even if most samples are characterized by an error less than 0.1%. This is confirmed even in more complex MGA transfers. For example, the overall ΔV of the best-known Cassini–like transfer reported in section 3.6 is

$$\Delta V = 4.930710687853622 \text{ km/s}. \tag{3.29}$$

Using the semi-analytical approach, the same evaluation yields

$$\Delta V = 4.931949452024911 \text{ km/s}, \tag{3.30}$$

matching a relative error of 0.025%.

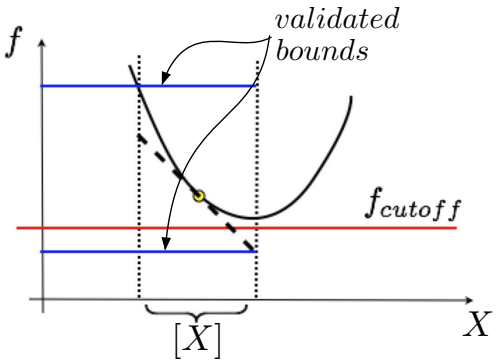


Figure 3.42: Validated bounds based on LDB.

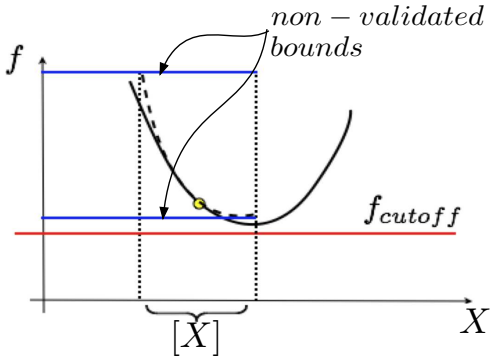


Figure 3.43: Non–validated bounds based on the quadratic bounder.

3.5 Quadratic polynomial bounder

As already pointed out, the Linear Dominated Bounder (LDB) has been used to estimate the range of the constraint functions over each box in the pruning processes of the previous examples. The LDB algorithm was introduced by Berz and Makino [26] in the framework of Taylor Models, which can be considered as the result of a joining process between differential algebraic techniques and interval analysis [29], aimed at performing computer–assisted validated computations (see chapter 6). Without loss of generality, suppose a polynomial of one variable has to be bounded within an interval I. LDB is based on the observation that, if d is the width of the domain interval, then the widths of the I^k are expected to scale as

$$\text{width}(I^k) \approx d^k. \tag{3.31}$$

Hence, the dominating part of the total bounds of a polynomial are expected to come from the linear part. A rigorous enclosure of the range is then obtained by suitably derived methods to bound higher order parts.

However, a careful analysis of the pruning process in MGA transfers has shown that polynomial bounding based on LDB is not as effective as desired. The reason of such behavior is schematically described in Figure 3.42. Within the pruning process, suppose a box $[X]$ can be pruned away if

$$\min_{[X]} f > f_{cutoff}, \tag{3.32}$$

where f is the constraint function under analysis, and f_{cutoff} is its maximum allowed value. The validated LDB uses the linear part of the Taylor

expansion of f around a reference point to get rigorous bounds of the Taylor expansion. Unfortunately, due to accuracy problems on the bounding process of nonlinear terms, the resulting interval enclosure overestimates the exact range. If LDB were used in the example of Figure 3.42, the box $[X]$ would not be pruned away, even if evidently satisfying inequality (3.32). This overestimation problem leads to ineffective pruning processes, and more boxes remain after pruning than expected from classical GASP results.

In order to overcome the previous difficulty, a non–validated quadratic bounder has been implemented, which uses the quadratic part of the Taylor expansion to get non–validated interval enclosures of the range of f over $[X]$. The process is illustrated in Figure 3.43. Second order information are added to the previous analysis. If the resulting quadratic polynomial turns out to be positive defined, its minimum is easily estimated by locating the zero–gradient point, and checking its inclusion within $[X]$. Only linear information are kept to estimate the minimum in case of lack of positive definition properties. In the example of Figure 3.43, the introduced non–validated bounder would correctly prune away the box. An evident drawback is that, in contrast to the validated LDB, the non–validated quadratic bounder could underestimate the exact interval enclosure. However, an extensive test phase confirmed the validity of the introduced approximations: using the resulting DA–based GASP algorithm, the best solutions available in literature for every test case reported in section 3.6 turn out to lie in the boxes remaining after pruning. This result is confirmed in Figure 3.44 and Figure 3.45, where the performances of GASP-DA are illustrated for the Earth–Mars transfer: the remaining boxes accurately enclose the pruned search space.

3.6 Test cases

In this section, significant test cases are addressed to properly assess the performances of the resulting DA–based GASP algorithm. Before reporting the results of the test phase, section 3.6.1 describes the philosophy adopted for the optimization process following the GASP-DA pruning process, which is evidently necessary to serve the purpose of optimizing the overall ΔV. Test cases are then reported, ordered by increasing complexity. In particular, relatively simple planet-to-planet transfers are addressed in sections from 3.6.2 to 3.6.4. MGA transfers involving three (sections 3.6.5 and 3.6.6), and four (section 3.6.7) planets follow. Finally, a Cassini–like transfer is optimized in section 3.6.8. The structure of each transfer is fixed and specified a priori in all test cases, i.e. the planets sequence of each transfer is fixed, thus not included in the set of design variables.

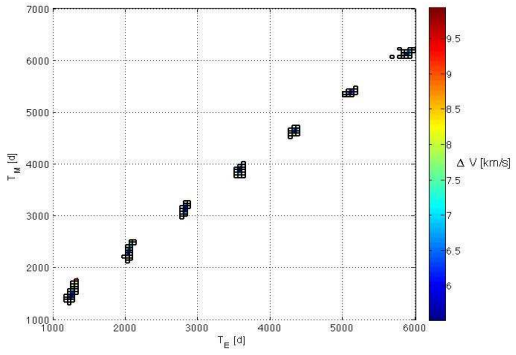


Figure 3.44: Performances of the non–validated quadratic bounder on the Earth–Mars transfer problem.

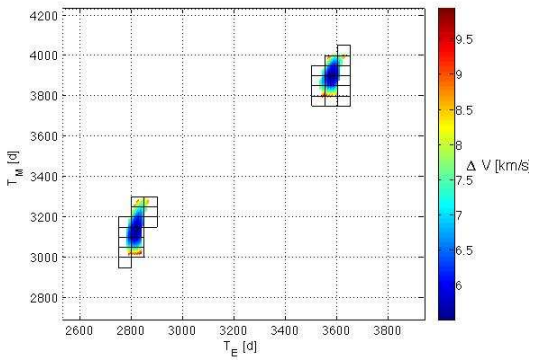


Figure 3.45: Performances of the non–validated quadratic bounder: detail of Figure 3.44.

For each test, the problem is first defined. The search space is identified by lower and upper bounds on each variable. Referring to the box-wise approach of GASP–DA, the size of the sampling box corresponding to each variable is reported. The cutoff values set for the departure and arrival ΔV, as well as for the corrective ΔV at each gravity assist are indicated. Pertaining instead the minimum allowed pericenter radii for the powered gravity assist maneuvers, a common philosophy has been adopted for all planets: given the mean radius of the generic planet P, R_P, the corresponding minimum allowed pericenter radius is set to $1.05\,R_P$.

Thus, the results of the pruning process are reported in terms of the total number of boxes, the number of feasible boxes remaining after pruning, and the CPU time required by the pruning process. It is worth observing that the total number of boxes is meant to give an idea on the dimension of the search space, and it is different from the number of processed boxes. The computational time is relative to a PC, 1.9 GHz CPU, 512 Mb RAM. Second order expansions are used in all test cases. As far as the subsequent optimization process is concerned, the best identified ΔV is reported, together with a plot of the corresponding optimal transfer. The values of the minimum pericenter radii considered for the GA throughout the report are given in Table 3.2.

Table 3.2: Minimum pericenter radii.

	V	E	M	J	S
$r_{p_{min}}$ [km]	6353.55	6689.55	3558.45	73406.55	61143.36

3.6.1 The optimization process

The outcome of the pruning process carried out by GASP–DA is a list of boxes, all fulfilling the requirement that a portion of them at least satisfies the feasibility conditions related to the constraining ΔV values and the minimum allowed pericenter values. An optimization process is then necessary to locate the minimum of the objective function, which is the purpose of the original optimization problem. This section is dedicated to the description of the philosophy adopted for this optimization phase.

Two approaches have been studied. The first approach is based on the use of the validated global optimizer COSY–GO, that is a branch-and-bound optimization algorithm, which takes advantage of Taylor models to bound the objective function value over the processed boxes [15]. Further details on COSY–GO can be found in chapter 6. Unfortunately, some general drawbacks can be identified which led to the decision of avoiding its use. First of all, the current version of COSY–GO can not manage constraints explicitly, which forces to either implement tailored techniques to handle constraints or to add them to the objective function as penalty terms. Moreover, due to the lack of the embedded possibility of exploiting the cascade structure of the MGA transfer problem, the computational time exponentially increases with the dimensionality.

The previous considerations have been of main influence for the decision of utilizing other approaches. In particular, multiple runs of a local optimizer have been exploited. The philosophy is based on the following steps:

1. The non–validated bounding process described in section 3.5 returns estimates of the minimum of the objective function, including its location, within each box.

2. The boxes are sorted based on the minimum objective function values estimated in step 1.

3. A certain number of boxes are suitable selected.

4. Starting from the estimated location of the minimum, a local optimization run is performed within each selected box.

The introduced optimization philosophy worths some comments. As already anticipated in the previous sections, different models and methods for the constraint and objective functions evaluation are implemented in the two main phases of the previous algorithm, i.e. search space pruning and total ΔV optimization. In particular, the planar Solar System model, and the semi–analytical approaches to the solution of the implicit equations are used within

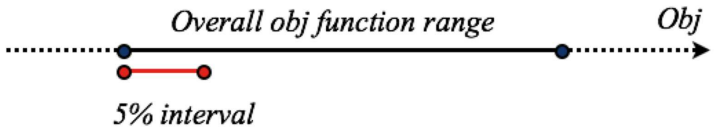


Figure 3.46: Heuristics for box selection.

the search space pruning phase only. This decision relies on the conservative hypothesis pertaining the planetary model depicted in section 3.2.2, and on the observation that the semi–analytical approaches deliver sufficiently accurate solutions of the implicit equations. The previous approximations are all abandoned in the subsequent optimization process of step 4, where the actual three-dimensional Solar System model, and the iterative Newton's method are used to evaluate the planetary ephemerides and to compute the solutions of the implicit equations.

Moreover, the boxes selection phase (step 3) is based on the following heuristics. Referring to Figure 3.46, suppose the overall objective function range is available, which is computed on all the minima identified within each box in step 1. The interval corresponding to 5% of the overall range is evaluated. The boxes whose estimated local minimum belongs to this interval are selected for the subsequent analyses. The 5% value is purely based on the experience gained during the extensive test phases. The parameter is anyway kept settable by the user.

The local optimization processes involved in step 4 are carried out within each box. This means that the identified local minima are interior to the feasible boxes, as well as the finally estimated global minimum. Consequently, if a minimum is identified to coincide with the best known solution available in literature, this turns out to lie in the boxes remaining after the pruning process based on GASP–DA. The local optimization processes are performed using the Matlab function "fmincon", which implements a sequential quadratic programming algorithm. No particular motivations underlie the selection of fmincon, which was mainly driven by the availability of interface tools between COSY-INFINITY and Matlab.

3.6.2 EM

Table 3.3: Search space and best identified solution for the EM transfer.

planet	variable	lower bound [days]	upper bound [days]	size [days]	cutoff (10) [km/s]	solution [days]
E	T_E	1000	6000	50	5	3573.188
M	t_{EM}	100	600	50	5	324.047

The first test case is the Earth–Mars transfer. The search space is defined in Table 3.3. It is worth observing that bounds on the departure epoch from Earth, T_E, and on the transfer time from Earth to Mars, t_{EM} are given. Consequently, the search space definition is made within the relative times formulation. However, as stated in section 3.3, the pruning processes are carried out in the absolute times formulation. This observation holds for all the following test cases. The box size along each epoch is indicated in the fifth column. The cutoff values for the maximum allowed departure and arrival ΔV are reported in the last column. A further constraint is imposed to the maximum allowed overall ΔV, which is reported within round brackets on the head row.

The main results pertaining the performances of GASP–DA are summa-

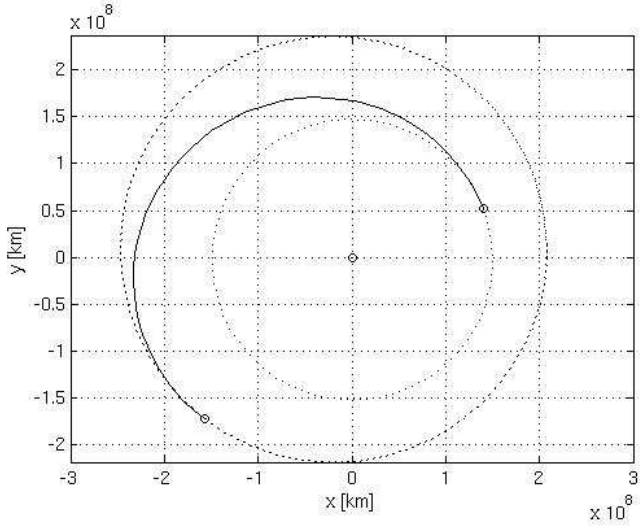


Figure 3.47: Trajectory of the best identified EM transfer.

rized in the following:

- Total number of boxes: 1000
- Feasible boxes: 64 (6.4%)
- CPU time: 0.220 s
- Best identified ΔV: 5.6673 km/s

The reported value of the best identified ΔV refers to the results of the optimization process described in section 3.6.1, which follows the pruning process of GASP–DA. The last column of Table 3.3 lists the values of the design variables corresponding to the best identified solution, whereas Figure 3.47 reports a two–dimensional plot of the related trajectory.

3.6.3 EV

Table 3.4: Search space and best identified solution for the EV transfer.

planet	variable	lower bound [days]	upper bound [days]	size [days]	cutoff (10) [km/s]	solution [days]
E	T_E	1000	6000	50	5	2706.421
V	t_{EV}	100	500	50	5	151.754

A further planet-to-planet transfer test case is here analyzed, selecting Venus as the arrival planet. Similarly to the previous case, the search space and the main parameters are defined in Table 3.4. The cutoff values for the maximum allowed departure and arrival ΔV are reported in the last column, and a further constraint of 10 km/s is added to the overall ΔV.

The main results pertaining the performances of GASP–DA are summarized in the following:

- Total number of boxes: 800
- Feasible boxes: 78 (9.7%)
- CPU time: 0.492 s
- Best identified ΔV: 6.0638 km/s

Figure 3.48 reports a two–dimensional plot of the best identified trajectory, whose solution vector is stated in the last column of Table 3.4.

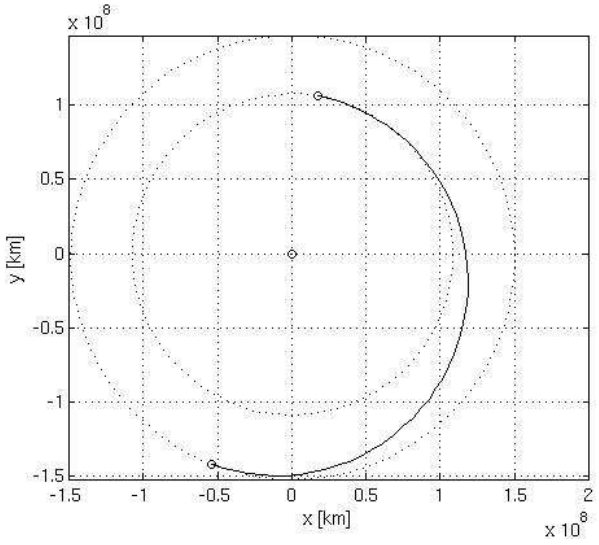


Figure 3.48: Trajectory of the best identified EV transfer.

3.6.4 EJ

Table 3.5: Search space and best identified solution for the EJ transfer.

planet	variable	lower bound [days]	upper bound [days]	size [days]	cutoff (16) [km/s]	solution [days]
E	T_E	1000	10000	50	8	6198.093
J	t_{EJ}	100	5000	100	8	1097.004

A last planet-to-planet transfer test case is here addressed. Jupiter is now the arrival planet, and the search space, together with the main algorithm settings, are defined in Table 3.5. A much larger launch window is considered in this case to better highlight the performances of the algorithm in terms of computational time, through comparison with the previous test cases. It is worth observing that a larger box size is used for T_J: the greater orbital period and the slower dynamics associated to the outer regions of the Solar System allows the user to select a wider sampling interval on the corresponding design variables, without significant loss of accuracy. A maximum allowed overall ΔV of 16 km/s is used as overall cutoff value.

Again, the main results pertaining the performances of GASP–DA are summarized in the following:

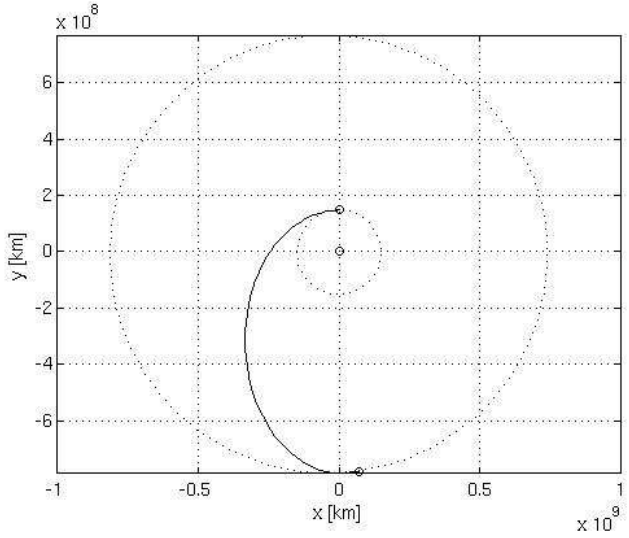


Figure 3.49: Trajectory of the best identified EJ transfer.

- Total number of boxes: 8820
- Feasible boxes: 46 (1.0%)
- CPU time: 2.116 s
- Best identified ΔV: 14.4197 km/s

The best solution found is identified by the values of the design variables reported in Table 3.5. Figure 3.49 illustrates a two–dimensional plot of the corresponding trajectory.

3.6.5 EVM

Table 3.6: Search space and best identified solution for the EVM transfer.

planet	variable	lower bound [days]	upper bound [days]	size [days]	cutoff (12) [km/s]	solution [days]
E	T_E	1000	6000	50	5	5611.480
V	t_{EV}	100	500	50	2	157.603
M	t_{VM}	100	1000	50	5	255.596

The study of actual MGA transfer starts from this section. In particular, one planet is added to the planets sequence, where a powered gravity assist maneuver is performed. In particular, an Earth–Venus–Mars transfer is investigated. The search space is defined in Table 3.6. Whereas the cutoff values at Earth and Mars are still related to the departure and arrival ΔV, the cutoff value of 2 km/s at Venus now refers to the maximum allowed corrective ΔV at the pericenter of the corresponding hyperbolic trajectory, as provided by the powered gravity assist model.

The main performance parameters are reported in the following, whereas the last column of Table 3.6 reports the optimal solution identified at the end of the pruning and optimization processes. A two-dimensional plot of the corresponding trajectory can be found in Figure 3.50.

- Total number of boxes: 14400
- Feasible boxes: 165 (1.1%)
- CPU time: 1.8321 s
- Best identified ΔV: 8.5226 km/s

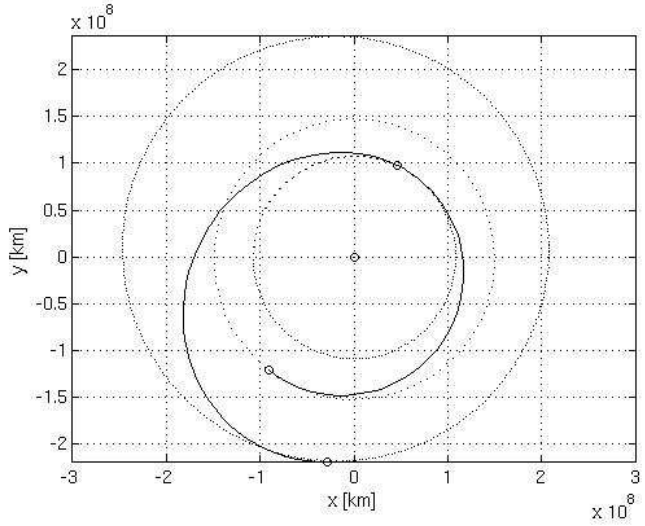


Figure 3.50: Trajectory of the best identified EVM transfer.

Although the total number of boxes is greater than in the previous test case, the required CPU time for the search space pruning turns out to be lower. This might be deemed affecting the correctness of the results. However, as

already pointed out, the reader must notice that the total number of boxes does not coincide with the number of boxes actually processed by GASP–DA, thanks to forward and backward constraining of GASP scheme [21].

3.6.6 EMJ

Table 3.7: Search space and best identified solution for the EMJ transfer.

planet	variable	lower bound [days]	upper bound [days]	size [days]	cutoff (20) [km/s]	solution [days]
E	T_E	1000	6000	50	10	3543.787
M	t_{EM}	100	1200	50	5	1121.130
J	t_{MJ}	100	2000	100	10	1109.307

A further MGA transfer involving three planets is investigated in this section. In particular, after departure from Earth, a powered gravity assist is performed at Mars before the arrival at Jupiter. The search space of the resulting Earth–Mars–Jupiter transfer is defined in Table 3.7. Cutoff values of 10, 5, and 10 km/s respectively are used at each planet, whereas a maximum value of 20 km/s constraints the overall ΔV.

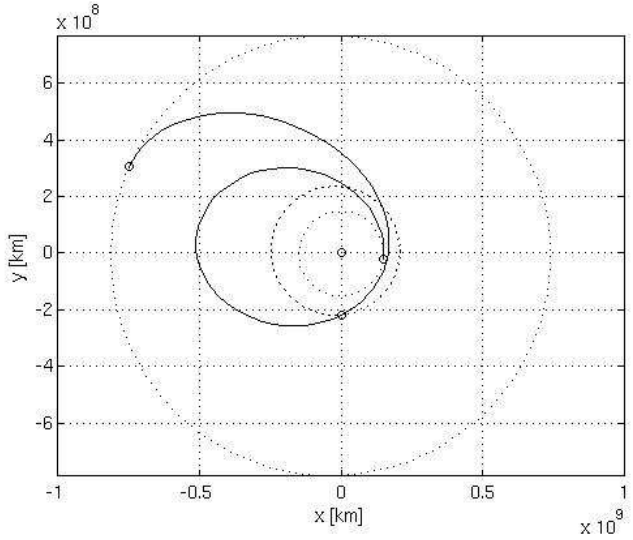


Figure 3.51: Trajectory of the best identified EMJ transfer.

The performances of GASP–DA are analyzed in the following:

- Total number of boxes: 41800
- Feasible boxes: 329 (0.7%)
- CPU time: 2.640 s
- Best identified ΔV: 13.4165 km/s

Figure 3.51 plots the trajectory corresponding to the lowest ΔV found, whose solution vector is reported in Table 3.7.

3.6.7 EVME

Table 3.8: Search space and best identified solution for the EVME transfer.

planet	variable	lower bound [days]	upper bound [days]	size [days]	cutoff (15) [km/s]	solution [days]
E	T_E	3000	4000	50	6	3985.096
V	t_{EV}	25	525	50	2	160.500
M	t_{VM}	20	520	50	2	168.941
E	t_{ME}	25	525	50	6	301.615

Two gravity assists are here introduced for an Earth–Venus–Mars–Earth transfer. Four planets are then involved, so leading to a four-dimensional optimization problem. The search space of this EVME transfer is defined in Table 3.8. A maximum value of 15 km/s is allowed for the overall ΔV.

The main results pertaining the performances of GASP–DA are summarized in the following, whereas Figure 3.52 reports a two-dimensional plot of the trajectory corresponding to the best identified solution. The related solution vector is supplied in Table 3.8.

- Total number of boxes: 20000
- Feasible boxes: 26 (0.1%)
- CPU time: 0.2 s
- Best identified ΔV: 12.4431 km/s

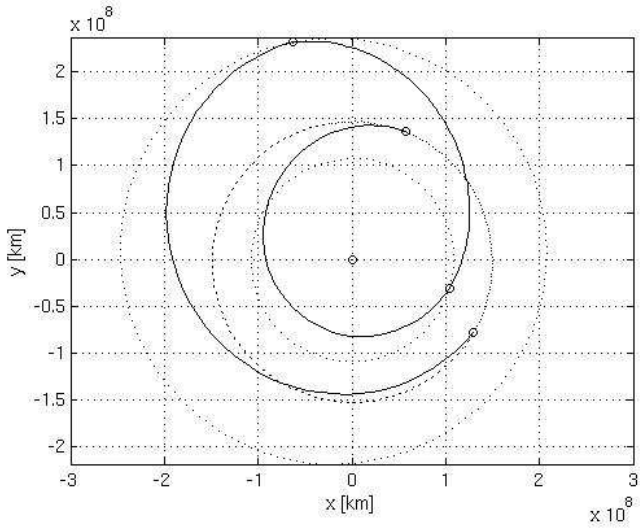


Figure 3.52: Trajectory of the best identified EVME transfer.

3.6.8 EVVEJS

This section is devoted to a complex MGA transfer problem, which has been already analyzed in the past [21, 16]. The arrival planet Saturn is reached after four gravity assists. Thus, the overall transfer involves six planets, so leading to a six-dimensional optimization problem. The sequence is fixed to Earth–Venus–Venus–Earth–Jupiter–Saturn. The previous sequence can be clearly related to the sequence of the Cassini mission, except no deep space maneuvers are allowed, whose introduction will be considered in the next chapter. Before presenting the results for this test case, some notes must be reported about a further necessary expedient which had to be added in case of MGA transfers where resonances might play an important role in the optimization, as in the case of the EVVEJS transfer.

As already pointed out in section 3.2, the discontinuity problem has been solved by adopting a planar planetary model. This expedient eliminates the discontinuities related to the transition from the short–way to the long–way solution of the Lambert's problem. Nevertheless, it is not able to solve the same problem for the related transition from the long–way to the "short way" solution. Consider the Venus–Venus arc of an EVVEJS transfer. Figure 3.53 shows that, when the angle between the first and second Venus encounter is close but less than 2π, the long way solution is selected. However, considering the same transfer time, the features of the resulting transfer suddenly change just after passing the angle 2π, which discriminates between the short–way

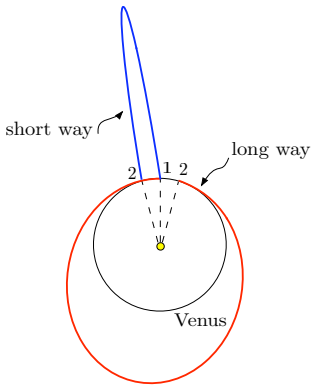


Figure 3.53: Venus–Venus arc: long–way to short–way solution.

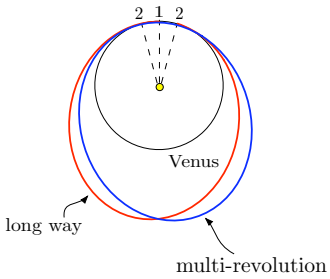


Figure 3.54: Venus–Venus arc: multi–revolution solution.

and the long–way solutions. The occurrence of this discontinuity is particularly undesirable in such kind of arcs, where resonance conditions are known to improve the overall transfer performances. The resonance occurs when the positions of the first and second planet encounter coincide. Consequently, if T_P indicates the orbital period of planet P, the transfer time associated to the connecting arc must satisfy

$$t_{VV} = r\ T_V, \tag{3.33}$$

where r is a positive integer number, $r \geq 2$. The presence of the previous discontinuity strongly affects the performances of the pruning algorithm: as the center of the box is used to discriminate between the two Lambert's problem solutions, boxes containing resonance situations will turn out to be obviously pruned away if the center of the box, which is the reference point for the Taylor expansions, corresponds to the short–way solution of Figure 3.53.

An expedient is used to overcome the previous difficulties, which is based on the observation that, in a planet-to-planet transfer involving only one planet, the discontinuity disappears if multi-revolution solutions are allowed (see Figure 3.54). In particular, given a box to be processed, is the enclosed transfer times include the resonance condition (3.33), and the enclosed angles between the first and second encounter include the resonance angle 2π, the multi–revolution solution is allowed. In this manner, the EVVEJS test case can be effectively managed by GASP–DA, as illustrated in the following test. The search space for the corresponding optimization problem is set as defined in Table 3.9. A maximum value of 12 km/s is used for the overall ΔV.

The main results pertaining the performances of GASP–DA are summarized in the following:

Table 3.9: Search space and best identified solution for the EVVEJS transfer.

planet	variable	lower bound [days]	upper bound [days]	size [days]	cutoff (12) [km/s]	solution [days]
E	T_E	-1000	0	50	4	-790.605
V	t_{EV}	10	410	25	2	159.962
V	t_{VV}	100	500	25	2	449.386
E	t_{VE}	10	410	25	2	54.171
J	t_{EJ}	400	2000	200	2	1029.979
S	t_{JS}	1000	6000	200	6	4560.625

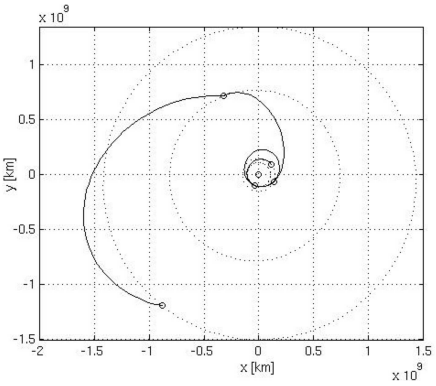


Figure 3.55: Trajectory of the best identified EVVEJS transfer.

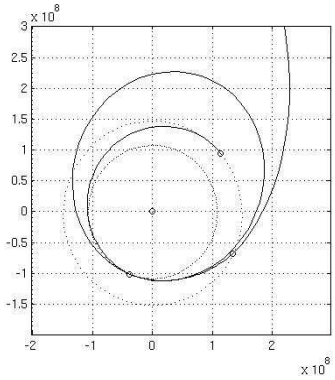


Figure 3.56: EVVEJS transfer: detail of Figure 3.55.

- Total number of boxes: 32768000
- Feasible boxes: 1085 (0.003%)
- CPU time: 1.93 s
- Best identified ΔV: 4.9357 km/s

Note that the best identified ΔV refers to the final insertion of the spacecraft into an orbit around Saturn of eccentricity 0.98 and periapsis 108950 km, as considered in [21]. Figure 3.55 reports a two–dimensional plot of the trajectory corresponding to the best identified solution, whose solution vector is listed in Table 3.9. The identified solution turns out to be characterized by an objective function value which is slightly higher than the known best solution in literature [32]. Further analyses are necessary to identify the

reason of such discrepancy, which is deemed to be related to differences on the minimum allowed pericenter radii at each gravity assist.

3.7 Final remarks

The introduction of DA techniques into GASP algorithm has been addressed in this chapter. The main idea was dealing with Taylor expansions of the constraint and objective functions over boxes that sample the search space, instead of handling point–wise evaluations as performed in the original GASP algorithm. This turned out to pose some significant challenges. First of all, the expansion of the solution of parametric implicit equations had to be achieved. Moreover, proper modeling decisions had to be taken, due to the presence of discontinuities on the constraint and objective functions. The resulting GASP–DA algorithm turned out to perform considerably well in typical test cases. This result is also the outcome of further important arguments, which led to the implementation of semi–analytical approaches to solve the implicit equations involved in the constraint and objective function evaluation, and to the development of a non–validated quadratic bounder. However, the planets sequence is kept fixed and specified a priori, which represents a key point for the future development of an effective MGA transfer optimizer.

Chapter 4

Introduction of DSM in GASP-DA

The GASP-DA algorithm, extensively discussed in the previous chapter, is now modified to manage DSM. These maneuvers are usually carried out to improve the performances of the transfer trajectories in terms of total cost. From the trajectory optimization standpoint, each DSM involves additional degrees of freedom that widen the search space and increase the chances of finding better global optima than those associated with pure MGA trajectories. The mathematical formulation of this new problem is not unique, and the final result is strongly affected by the problem transcription, especially when working in a DA frame. In this chapter we consider two different formulations of the MGA-DSM problem. We choose the strategy that better fits the GASP-DA algorithm already developed. In particular, we implement the GASP-DSM-DA algorithm that prunes the search space associated to the DSM variables beside that associated to the MGA ones (epochs).

In section 4.1 we make some preliminary considerations and point out the problems associated to the DSM introduction. Two different strategies for the problem formulation are deeply analyzed in sections 4.2 and 4.3. In particular, in both frameworks, the dependence of the functions on the problem variables is studied for both a planet-to-planet and a MGA case. The strategy that better fits the DA implementation is introduced in GASP-DA to produce the GASP-DSM-DA algorithm (section 4.4). This algorithm is tested for the solution of some sample cases discussed in section 4.5. Final considerations are made in section 4.6.

4.1 Preliminary Considerations

In a MGA problem, the transfer trajectory is made up by a number of Lambert's arcs linked together in proximity of the planets. In this framework, the objective function evaluation consists in the solution of a set of Lambert's problems with the positions of two consecutive planets as inputs. This information can be extracted once the epochs of the planets encounters are specified, and therefore the decision vector of a MGA problem consists in a collection of epochs.

In a MGA-DSM problem, the objective function evaluation consists in the solution of several Lambert's problems (between two consecutive planets or, depending on the problem structure, between a planet and a maneuver point, and vice versa).The location of the maneuver points has to be specified by *appending* new variables to the decision vector. It can be easily shown that, for each DSM introduced, the minimum number of additional variables is four in the 3D case and three in the 2D case. For the same reasons discussed in the previous chapter, in this study, the search space pruning of MGA-DSM transfers is carried out in the 2D space, whereas the subsequent optimization deals with the full 3D space. This means that, letting n_P and n_D be the number of planets and the number of DSM of a MGA-DSM transfer, respectively, the decision vector for the search space pruning, in our case, is made up by $n_P + 3n_D$ variables.

One of the purposes of this study is the implementation and the assessment of the search space pruning (of both MGA and MGA-DSM problems) exploiting the DA formalism. In particular, the pruning process relies on inequality constraints, as the magnitude of the Δv maneuvers and the closest distance to a planet, usually expressed as nonlinear functions of the decision vector. Analogously to the MGA case, the pruning process of MGA-DSM problems consists in: 1) representing these nonlinear functions, over appropriate subsets (boxes) of the search space, as Taylor polynomials of the decision variables; 2) bounding these functions using suitable bounders; 3) pruning away unfeasible boxes from the search space. Differently from a point-wise approach, the whole procedure is based on the reliability of the bounding process and, in order to efficiently prune the search space, working with smooth functions of as few variables as possible is crucial. This means that, in specifying the DSM point, different strategies generate functions having different dependencies from the decision variables, and so the outcome of the pruning process, in a DA frame, is different. This is the key that has to be taken into account for the strategy selection.

Another consideration concerns the increased computational burden when moving from the MGA to the MGA-DSM problem. This is not only related

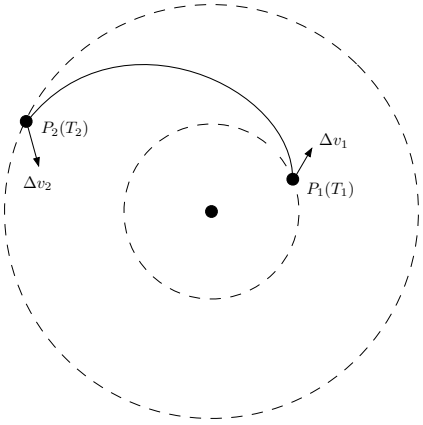


Figure 4.1: A planet-to-planet transfer.

to the increased dimension of the search space (from n_P to $n_P + 3n_D$), but rather it is an intrinsic consequence of representing a function with its Taylor expansion. The number of monomials needed to represent a function of v variables up to the order n is $NM = (n+v)!/(n!v!)$. This means that, even if the order is kept constant, the number of monomials necessary to deal with a MGA-DSM problem increases with factorial law if compared to a simple MGA problem. These monomials should be stored along the entire pruning process, and consequently the $3n_D$ additional variables strongly affect the algorithm performances. In particular, the time for the algorithm execution increases since a larger amount of memory has to be allocated and a greater number of operations is required.

The bottom line is that, in general, the complexity of a MGA-DSM problem increases if compared to a standard MGA problem. This complexity can be circumscribed by carefully selecting the strategy for the representation of the maneuver point. In particular, it is important to preserve the idea of the GASP algorithm: subdivide the problem into a cascade of subproblems and exploit the cut-off values to prune away unfeasible zones.

Let us briefly recall the statement of a planet-to-planet problem (Figure 4.1). In this simple problem, the transfer from P_1 to P_2 is considered. The two epochs, departure and arrival, are denoted with T_1 and T_2, respectively. The decision vector is then $\mathbf{x} = [T_1, T_2]$. The objective function is evaluated by solving one Lambert's problem. The pruning process consists in finding the feasible subset of the search space, $\mathcal{X}$, such that

$$\Delta v_1(\mathbf{x}^*) \leq \Delta v_1^{max}, \quad \Delta v_2(\mathbf{x}^*) \leq \Delta v_2^{max}, \quad \forall \mathbf{x}^* \in \mathcal{X}. \tag{4.1}$$

The size of the departure and arrival maneuvers is denoted by Δv_1 and Δv_2, respectively; the two corresponding cut-off values are Δv_1^{max} and Δv_2^{max}.

4.2 Forward Propagation Strategy

In this section we show the first devised strategy for the formulation of the MGA-DSM problem. In the *forward propagation* method, the location of the maneuver point is a function of all the variables involved in transfer up to the point considered. As it will be shown, this procedure intrinsically exploits some problem constraints but gives rise to functions depending on many number of variables. We show this technique, and the associated set of $3n_D$ additional variables, by means of two examples: a planet-to-planet transfer (with one DSM) and a MGA transfer (three planets and two DSM).

4.2.1 Planet-to-Planet Case

Let us consider the problem illustrated in Figure 4.2. A deep space maneuver, located at the point D, is performed in the transfer between the two planets P_1 and P_2. In order to specify the maneuver point, we introduce two angles

- α – *flight-path angle*: is the angle between the velocity of P_1, $\mathbf{v}_1$, and the vector $\Delta\mathbf{v}_1$;
- θ – *incremental anomaly*: is the anomaly of the point D relative to P_1.

The departure velocity, $\mathbf{v}'$, can be computed once both α and Δv are assigned, i.e. $\mathbf{v}' = \mathbf{v}'(\mathbf{v}_1, \alpha, \Delta v_1)$. The location of D can be determined by forward propagating, using Lagrangian coefficients [2], the new initial condition $\mathbf{y}'_1 = \{\mathbf{r}_1, \mathbf{v}'_1\}^T$. The propagation is carried out until the incremental anomaly of D is equal to θ. The position of D, indicated with $\mathbf{r}_D$, depends on

$$\mathbf{r}_D = \mathbf{r}_D(T_1, \alpha, \theta, \Delta v_1).$$

In this frame it is natural to assume a decision vector made up by $\mathbf{x} = [T_1, T_2, \alpha, \theta, \Delta v_1]$; the objective function evaluation requires one propagation (P_1–D arc) and the solution of one Lambert's problem (D–P_2 arc). The problem consists in finding $\mathcal{X}$ such that, $\forall \mathbf{x}^* \in \mathcal{X}$, we have

$$\Delta v_1(\mathbf{x}^*) \leq \Delta v_1^{max}, \quad \Delta v_D(\mathbf{x}^*) \leq \Delta v_D^{max}, \quad \Delta v_2(\mathbf{x}^*) \leq \Delta v_2^{max}, \tag{4.2}$$

where Δv_D is the size of the DSM and Δv_D^{max} is its maximum allowed value. It is not difficult to derive the explicit dependencies of the functions in Eq. (4.2) on the problem variables

$$\begin{aligned} \Delta v_1 &= \Delta v_1, \\ \Delta v_D &= \Delta v_D(T_1, T_2, \alpha, \theta, \Delta v_1), \\ \Delta v_2 &= \Delta v_2(T_1, T_2, \alpha, \theta, \Delta v_1). \end{aligned} \tag{4.3}$$

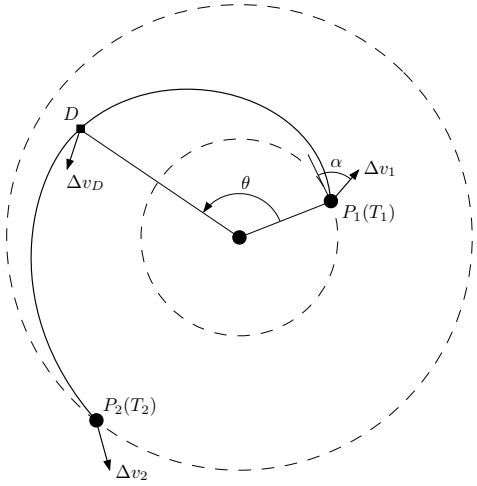


Figure 4.2: Forward propagation method: the planet-to-planet case.

Even in this simple case ($n_P = 2$, $n_D = 1$), the last two functions, Δv_D and Δv_2, depends on all the problem variable ($n_P + 3n_D$). Finally, it is worth noting that this strategy intrinsically exploits the first inequality constraint in (4.2) since Δv_1 is a problem variable. Assuming $\Delta v_1 \in [0, \Delta v_1^{max}]$, the respect of $\Delta v_1 \leq \Delta v_1^{max}$ is automatic.

4.2.2 MGA Case

As illustrated in Figure 4.3, we consider a MGA case with three planets (P_1, P_2, P_3) and two maneuvers (D_1 and D_2). In this case the decision vector is

$$\mathbf{x} = [T_1, T_2, T_3, \alpha, \theta_1, \Delta v_1, \theta_2, r_p, \Delta v_2],$$

where the last two variables, consistent with the problem structure, are taken in agreement with the principles of the forward propagation

r_p – *pericenter radius*: associated to the hyperbola of the P_2 gravity assist;

Δv_2 – *gravity assist burn*: is the size of the maneuver carried out, parallel to the velocity, at the hyperbola pericenter.

The pruning problem consists in finding $\mathcal{X}$ such that, $\forall \mathbf{x}^* \in \mathcal{X}$, we have

$$\begin{aligned} &\Delta v_1(\mathbf{x}^*) \leq \Delta v_1^{max}, \quad \Delta v_{D1}(\mathbf{x}^*) \leq \Delta v_D^{max}, \quad \Delta v_2(\mathbf{x}^*) \leq \Delta v_2^{max}, \\ &\Delta v_3(\mathbf{x}^*) \leq \Delta v_3^{max}, \quad \Delta v_{D2}(\mathbf{x}^*) \leq \Delta v_{D2}^{max}, \quad r_p(\mathbf{x}) \geq r_p^{min}, \end{aligned} \tag{4.4}$$

where r_p^{min} is a prescribed distance grater than P_2's radius. The last constraint is considered to have a safe P_2 gravity assist.

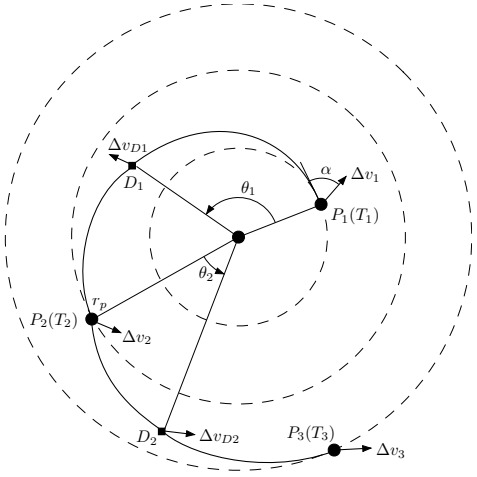


Figure 4.3: Forward propagation method: the MGA case.

Analogously to the previous case, we are interested in assessing the dependence of the D_1 and D_2 position vectors, $\mathbf{r}_{D1}$ and $\mathbf{r}_{D2}$, on the problem variables

$$\begin{aligned}\mathbf{r}_{D1} &= \mathbf{r}_{D1}(T_1, \alpha, \theta_1, \Delta v_1),\\ \mathbf{r}_{D2} &= \mathbf{r}_{D2}(T_1, T_2, \alpha, \theta_1, \Delta v_1, \theta_2, r_p, \Delta v_2).\end{aligned}$$

It can be shown that the functions to bound are

$$\begin{aligned}\Delta v_1 &= \Delta v_1,\\ \Delta v_{D1} &= \Delta v_{D1}(T_1, T_2, \alpha, \theta_1, \Delta v_1),\\ \Delta v_2 &= \Delta v_2,\\ r_p &= r_p,\\ \Delta v_{D2} &= \Delta v_{D2}(T_1, T_2, T_3, \alpha, \theta_1, \Delta v_1, \theta_2, r_p, \Delta v_2),\\ \Delta v_3 &= \Delta v_3(T_1, T_2, T_3, \alpha, \theta_1, \Delta v_1, \theta_2, r_p, \Delta v_2).\end{aligned} \tag{4.5}$$

The objective function evaluation requires, in this case, two propagations (P_1–D_1 and P_2–D_2 arcs) and the solution of two Lambert's problems (D_1–P_2 and D_2–P_3 arcs). Three of the six constraints in (4.5) are again automatically respected thanks to an appropriate choice of the problem variables.

Nevertheless wte note that, as in the simple planet-to-planet case, the last two functions depend on all the $n_P + 3n_D$ variables. In general, with the forward propagation strategy, the more planets and DSM are involved in the transfer, the more complicated are the functions to bound. This feature does not depend on *where* the maneuver is located. It holds independently

from the problem structure. This strategy is not appropriate in a DA environment because bounding highly nonlinear functions of many variables is very difficult and may easily lead to incorrect results. With this strategy the problem complexity blows-up.

It is necessary to reformulate the MGA-DSM problem in suitable coordinates (for the representation of the $\mathbf{r}_{Di}$) such that the dependency chain in Eq. (4.5) is broken. This reformulation is analogous to the passage from the time-of-flight variables to the epochs in the transcription of the GASP-DA algorithm (see Chapter 3). The aim is to reduce the problem complexity.

4.3 Absolute Variables Strategy

In this section we show the *absolute variables* strategy. The idea of this method is to represent the maneuver points, D_i, as direct functions of the problem variables avoiding, in this way, any forward propagation. With this approach the whole transfer trajectory is broken into a series of Lambert's arcs. It can be shown that the number of Lambert's problems to solve is $n_P + n_D - 1$, therefore the computational charge increases a little bit if compared to the forward propagation strategy. We will show that this small drawback is worth the problem simplification. The absolute variables strategy is again illustrated by means of a planet-to-planet and a MGA transfers.

4.3.1 Planet-to-Planet Case

The planet-to-planet case, in absolute variables, is shown in Figure 4.4. Two new variables are introduced

r_D – *maneuver radius*: is the distance between D and the Sun;

t_D – *partial tof*: is the time-of-flight associated to the P_1–D arc.

It is straightforward that the latter is subject to the constraint $t_D \leq T_2 - T_1$. The location of D can be obtained by rotating $\mathbf{r}_1 = \mathbf{r}_1(T_1)$ of the angle θ and by taking, on this line, a point with distance r_D from the origin. With this strategy the following dependency holds

$$\mathbf{r}_D = \mathbf{r}_D(T_1, r_D, \theta).$$

The decision vector is made up by $\mathbf{x} = [T_1, T_2, r_D, \theta, t_D]$ and the objective function evaluation requires the solution of two Lambert's problems. The nonlinear constraints (4.2) are valid for this case as well. As already pointed

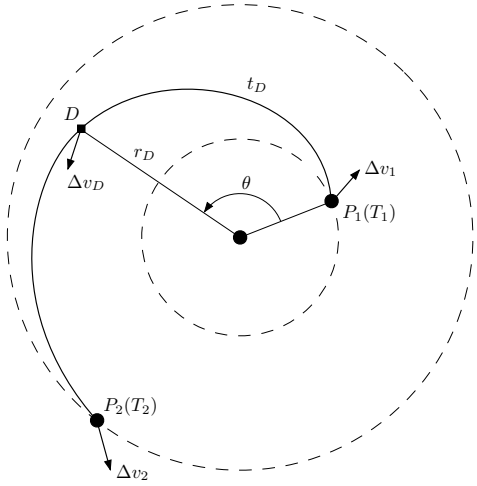


Figure 4.4: Absolute variables method: the planet-to-planet case.

out, we are interested in the dependencies of the functions to bound on the problem variables. In this case we have

$$\begin{aligned} \Delta v_1 &= \Delta v_1(T_1, r_D, \theta, t_D), \\ \Delta v_D &= \Delta v_D(T_1, T_2, r_D, \theta, t_D), \\ \Delta v_2 &= \Delta v_2(T_1, T_2, r_D, \theta, t_D). \end{aligned} \tag{4.6}$$

It seems, apparently, that this strategy does not simplify the problem as relations (4.6) are even more complicated than (4.3). (The first of (4.6) depends on four variables; this function was trivial in (4.3).) The last two functions still depend on $n_P + 3n_D$ variables. The usefulness of the absolute variables approach can be appreciated in the following case.

4.3.2 MGA Case

The MGA problem in absolute variables is illustrated in Figure 4.5. In analogy to the MGA problem in the forward propagation strategy, this problem has nine optimization variables. The decision vector indeed is $\mathbf{x} = [T_1, T_2, T_3, r_{D1}, \theta_1, t_{D1}, r_{D2}, \theta_2, t_{D2}]$. The last three variables, r_{D2}, θ_2, t_{D2}, have the same meaning of those introduced in the previous section. The inequality for t_{D1} still holds, whereas t_{D2} is subject to $t_{D2} \leq T_3 - T_2$. The pruning problem is analogous to the that stated by the conditions (4.4). The objective function evaluation requires the solution of four Lambert's problems.

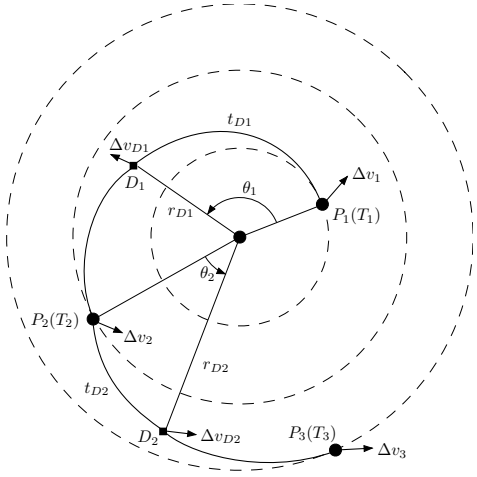


Figure 4.5: Absolute variables method: the MGA case.

The point is that the variables useful to specify D_2 are counted from P_2 and so the two position vectors $\mathbf{r}_{D1}$ and $\mathbf{r}_{D2}$ have the following dependencies

$$\begin{aligned}\mathbf{r}_{D1} &= \mathbf{r}_{D1}(T_1, r_{D1}, \theta_1),\\ \mathbf{r}_{D2} &= \mathbf{r}_{D2}(T_2, r_{D2}, \theta_2).\end{aligned}$$

It is worth comparing these relations to those obtained in the MGA problem with the forward propagation. In this case we have functions of three variables; in the previous case we came up with functions of four and eight variables. The functions to bound are

$$\begin{aligned}\Delta v_1 &= \Delta v_1(T_1, r_{D1}, \theta_1, t_{D1}),\\ \Delta v_{D1} &= \Delta v_{D1}(T_1, T_2, r_{D1}, \theta_1, t_{D1}),\\ \Delta v_2 &= \Delta v_2(T_1, T_2, r_{D1}, \theta_1, t_{D1}, r_{D2}, \theta_2, t_{D2}),\\ r_p &= r_p(T_1, T_2, r_{D1}, \theta_1, t_{D1}, r_{D2}, \theta_2, t_{D2}),\\ \Delta v_{D2} &= \Delta v_{D2}(T_2, r_{D2}, \theta_2, t_{D2}),\\ \Delta v_3 &= \Delta v_3(T_2, T_3, r_{D2}, \theta_2, t_{D2}).\end{aligned} \tag{4.7}$$

We now compare relations (4.5) to (4.7). First of all, we note that the last two functions in (4.7) depend on four and five variables, respectively, while the last two functions in (4.5) depend on nine variables. This occurs because, in this case, the variables describing D_2 (r_{D2}, θ_2, t_{D2}) are taken from P_2. The only critical functions in this case are Δv_2 and r_p (quantities at the P_2 gravity assist). With the forward propagation method, Δv_2 and r_p are problem variables, and therefore there are no dependencies at all (Eqs.

(4.5)). In this case Δv_2 and r_p are functions of eight variables. Nevertheless, this kind of complexity (i.e. functions of 8 variables) occurs only when two maneuvers are located across a gravity assist. In general, with the absolute variables method, a function depends *at most* on 8 variables. In the former case, the last two functions depend on $n_P + 3n_D$ variables, independently from the problem structure. (For instance, a simple problem with $n_P = 4$ and $n_D = 2$ produce functions of ten variables.) Finally we observe that with the absolute variables strategy, even in complicated MGA-DSM problems, a pure planet-to-planet arc (no DSM) simplifies the problem because it cuts the variable dependencies in the functions to bound. A pure planet-to-planet arc depends on two variables (two epochs). The functions defined in the subsequent arcs do not depend on the variables defining the arcs before the planet-to-planet one. This is not true in the forward propagation approach.

It is clear now that the absolute variables strategy involves functions of four and five variables excepting the case when two maneuvers are located across a gravity assists (where functions of eight variables arise). This strategy fits the requirements of working in a DA frame since the produced functions are easier to bound. The only drawback is the increased number of Lambert's problems that have to be solved. (Nevertheles, as shown in the previous chapter, the analytic approximation reduces the solution of the Lambert's problem to a mere function evaluation; in this way, there is no sensitive growing of the computational charge with the absolute variables strategy.) It is important pointing out that in this study we have assumed that no more than one DSM can be performed along each planet-to-planet transfer.

4.4 Implementation of GASP-DSM-DA

The features of the two analyzed methods are summarized below.

Forward Propagation Method

Some functions depending on $n_P + 3n_D$ variables
Solution of $n_P - 1$ Lambert's problems
Advantages: Some cut-off (departure Δv, pericenter Δv, and pericenter radius) are intrinsically respected; reduced number of Lambert's problems.
Drawbacks: Functional dependence on all the problem variables; more complicated relations between the variables and the functions to bound; problem complexity depending on the problem structure (on "where" the maneuver is located); problem complexity blowing-up for difficult cases.

Absolute Variables Method

Functions depending at most on 8 variables
Solution of $n_P + n_D - 1$ Lambert's problems
Advantages: Functions of at most 8 variables to bound (only in cases where two maneuvers are located across a gravity assist); maneuver point location specified with three variables; bounded problem complexity (the complexity does not blow-up along the transfer); a pure planet-to-planet arc simplifies the problem and cuts the functional dependencies.
Drawbacks: Increased number of Lambert's problems; increased search space (no constraints embedded in the problem formulation).

It is clear that, since the aim of working in a DA frame was having functions of as few variables as possible, the absolute variables strategy has been introduced in GASP-DA. The inclusion of DSM into this algorithm gives rise to the GASP-DSM-DA procedure. The pruning part of the algorithm is still based on a grid sampling on the search space and on representing the functions over boxes with Taylor polynomials. As in the GASP-DA algorithm, the box-size along each variable is chosen depending on the problem to solve. This is valid for both the n_P epochs and for the $3n_D$ auxiliary variables useful to specify the maneuver points. In next sections, we will show the results found for some sample cases. The amplitudes of the n_P epochs will be given; those of the $3n_D$ auxiliary variables are here discussed.

Table 4.1 reports both the amplitudes and the bounds for the DSM auxiliary variables. We have found that, for the auxiliary variables, amplitudes like those specified in the table represent a good trade-off between the accuracy of the representation and the total number of boxes. The table shows also the lower and upper bounds for these variables. These bounds are relative to a maneuver located in the transfer arc between P_1 and P_2 (Figure 4.4). The bounds for θ are trivial. The terms r_1 and r_2, $r_2 > r_1$, stand for the mean radii of the P_1 and P_2 orbits, respectively. Thus, the maneuver is constrained to lie into an annular region enclosing the planets' orbits. (If $r_1 > r_2$, then

Table 4.1: Bounds and box-size for the $3n_D$ auxiliary variables.

variable	lower bound	upper bound	amplitude	units
r_D	$0.9r_1$	$1.1r_2$	0.1	AU
θ	0	360	10	deg
t_D	0	$T_2 - T_1$	50	days

$r_D \in [r_2, r_1]$.) When the maneuver is located between two encounters of the same planet, that is $P_1 = P_2$, (like in EVdVEJ) we set $r_D \in [0.9r_1, 1.1a_{1:2}]$, where $a_{1:2}$ is the semimajor axis of an orbit in 1:2 resonance with the orbit of P_1 (Venus). We let the partial tof, t_D, to vary within $t_D \in [0, T_2 - T_1]$, where T_1 and T_2 are the epochs at the P_1 and P_2 encounter, respectively.

Analogously to GASP-DA, in the GASP-DSM-DA algorithm a local optimization is performed after the pruning process. If a box is feasible the local optimization is carried out on a domain enclosing, with a 10% tolerance, such box. This choice speeds-up the local optimization as the optimizer runs over small domains. Thus the whole pruning and optimization sequence is implemented in a deterministic way. It is worth noting that the repeatability of the results is preserved.

4.5 Test Cases

In the next subsections we show the results found for some sample cases. First, the classic Earth-Mars transfer, with an intermediate DSM, is discussed (Section 4.5.1). We show that, compared to the results obtained with GASP-DA, the DSM introduction improves the performance of the transfers. After this simple case, four possibilities for a mission to Jupiter are taken into account (Sections 4.5.2–4.5.5). These cases differ in the structure of the problem but, in any case, they all deal with a single DSM. In Sections 4.5.6 and 4.5.7 transfers with two DSM are considered. The first is a transfer to Mars via Venus; the second is an Earth-Venus-Mars-Earth transfer. The last two cases are devoted to the Cassini-like transfers (with a single maneuver - Section 4.5.8 - and with two DSM - Section 4.5.9).

The structure of these transfers is fixed and specified a priori. This means that the position of the maneuvers within the planets sequence is not optimized. In next chapter we show a possible way to both prune the search space and, at the same time, compute optimum MGA-DSM transfers with automatic introduction of DSM. (The maneuver is denoted by "d".) For each case we define the search space (we give the lower and upper bounds of the time variables), we give both the total number of boxes and the number of feasible ones, the computational time, the value of the best objective function, and, if possible, the best objective function found by GASP-DA. In addition, we give the time vector of the optimal solution for each case (last row of the summary table). It should be noted that the total number of boxes gives an idea on the dimension of the search space; it is not the number of processed boxes. The computational time is relative to a PC with 2.01 Ghz CPU and 512 Mb RAM.

4.5.1 EdM

Table 4.2: Time bounds and amplitudes for the EdM transfer.

	T_E MJD 2000	t_{EM} days
L_b	1000	200
U_b	2000	650
Δ	50	50
EM	1243.2	606.2

The search space, relative to the time variables only, of the EdM transfer is shown in Table 4.2. The box amplitude along each time variable is indicated in the last row. The whole search space takes into account the domain for the 3 additional variables (r_D, θ, t_D) stated in Table 4.1. The GASP-DSM-DA algorithm solves this problem in 253.2 s. We summarize below some features of the problem (the velocities are in km/s). The best solution found is shown in Figure 4.6.

- Problem constraints: $\Delta v_E \leq 3$, $\Delta v_d \leq 3$, $\Delta v_M \leq 3$, $\Delta v_{tot} \leq 7$
- Total number of boxes = 388800, feasible boxes = 1603 (0.41%)
- Best objective function = 5.632 (2.77 + 2.77 + 0.07)
- Best objective function with GASP-DA = 5.667

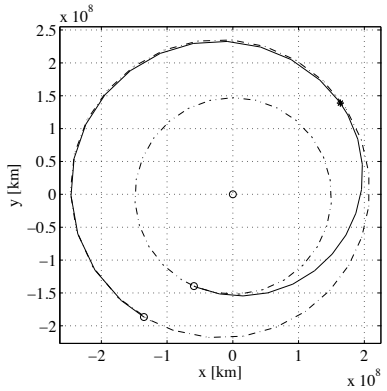


Figure 4.6: Optimal EdM transfer (the asterisk indicates the DSM).

4.5.2 EMdJ

	T_E MJD 2000	t_{EM} days	t_{MJ} days
L_b	1000	300	1000
U_b	3000	700	2000
Δ	50	50	100
EMJ	2804.7	321.9	1161.9

Table 4.3: Time bounds and amplitudes for the EMdJ transfer.

Table 4.3 states the bounds and the interval amplitudes of each time variable of the EMdJ problem. Table 4.1 should be considered for the DSM variables. In general, for the gravity assists, the same pericenter constraints stated in the previous chapter have been used. A purely ballistic Mars gravity assist can be achieved from the powered model by simply setting $\Delta v_M = 0$. As in the previous problem, the introduction of a DSM improves the objective function found by GASP-DA. The CPU time is 451 s.

- Constraints: $\Delta v_E \leq 5$, $\Delta v_M \leq 0$, $\Delta v_d \leq 5$, $\Delta v_J \leq 5$, $\Delta v_{tot} \leq 15$
- Total number of boxes = 8.52e7, feasible boxes = 323 (3.79e-4%)
- Best objective function = 12.481 (3.93 + 0 + 4.16 + 4.39)
- Best objective function with GASP-DA = 13.416

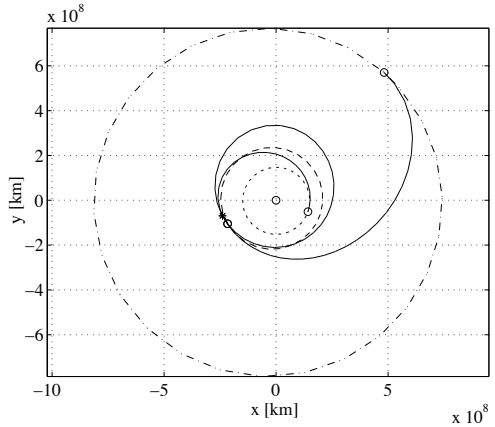


Figure 4.7: Optimal EMdJ transfer.

4.5.3 EMdMJ

	T_E MJD 2000	t_{EM} days	t_{MM} days	t_{MJ} days
L_b	3650	30	330	600
U_b	7300	430	830	2000
Δ	50	100	100	200
EMMJ	4353.8	371.1	915.8	1129.5

Table 4.4: Time bounds and amplitudes for the EMdMJ transfer.

The time domain for the EMdMJ problem is stated in Table 4.4. In this case the maneuver radius is search within $r_D \in [0.9r_M, 1.1a_{1:2}]$, where r_M is the mean radius of the Mars' orbit whereas $a_{1:2}$ is the semimajor axis of a 1:2 resonant orbit with Mars' orbit. This problem is solved in 144.2 s. The result found with GASP-DA for the same problem is once again improved.

- $\Delta v_E \leq 4$, $\Delta v_M \leq 0$, $\Delta v_d \leq 3$, $\Delta v_M \leq 0$, $\Delta v_J \leq 7$, $\Delta v_{tot} \leq 12$
- Total number of boxes = 9.19e7, feasible boxes = 717 (7.8e-3%)
- Best objective function = 10.843 (3.18 + 0 + 1.01 + 2.42 + 4.21)
- Best objective function with GASP-DA = 12.864

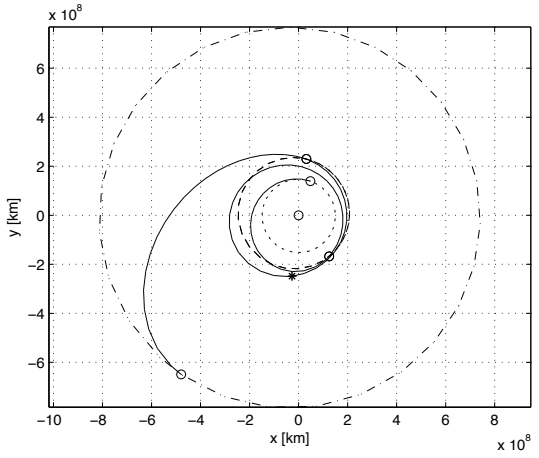


Figure 4.8: Optimal EMdMJ transfer.

4.5.4 EVdVEJ

	T_E MJD 2000	t_{EV} days	t_{VV} days	t_{VE} days	t_{EJ} days
L_b	3650	80	80	80	600
U_b	7300	430	830	830	2000
Δ	50	25	25	50	200
EVVEJ	3859.5	119.2	429.6	564.8	1244.3

Table 4.5: Time bounds and amplitudes for the EVdVEJ transfer.

The five-dimensional time domain for the EVdVEJ problem is described in Table 4.5. When joint to the domain for the DSM, the search space results huge and subdivided into a large number of boxes (1e10). Thanks to the GASP-DSM-DA algorithm, this problem has been solved efficiently in 2770 s. The problem constraints are shown below ($\Delta v_V \leq 0$ is valid for both GA). The found global optimal trajectory is illustrated in Figure 4.9.

- $\Delta v_E \leq 4.5$, $\Delta v_V \leq 0$, $\Delta v_d \leq 0.5$, $\Delta v_E \leq 0$, $\Delta v_J \leq 7$, $\Delta v_{tot} \leq 12$
- Total number of boxes = 1.85e10, feasible boxes = 3.80e4 (2.06e-4%)
- Best objective function = 9.304 (3.04 + 0 + 0.26 + 0 + 0 + 5.99)
- Reference solution [31] = 10.503

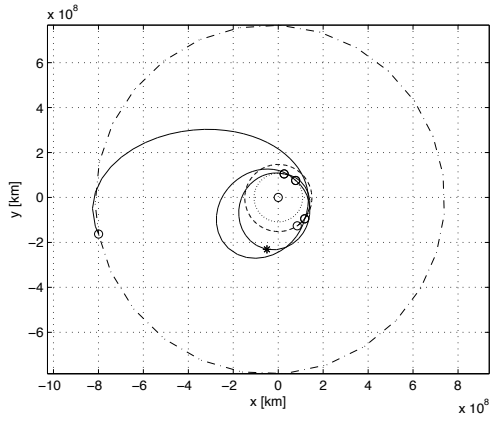


Figure 4.9: Optimal EVdVEJ transfer.

4.5.5 EVEdEJ

	T_E MJD 2000	t_{EV} days	t_{VE} days	t_{EE} days	t_{EJ} days
L_b	3650	80	80	80	600
U_b	7300	430	830	830	2000
Δ	50	25	50	50	200
EVEEJ	3863.4	128.8	288.4	713.3	1068.2

Table 4.6: Time bounds and amplitudes for the EVEdEJ transfer.

The EVEdEJ problem is stated in Table 4.6 in terms of time variables. As the previous problem, the EVEdEJ case has a huge search space and a grid sampling would have been impossible without a pruning algorithm. This problem has been solved in 2392 s. The main features are listed below.

- $\Delta v_E \leq 4$, $\Delta v_V \leq 0$, $\Delta v_E \leq 0$, $\Delta v_d \leq 3$, $\Delta v_E \leq 0$, $\Delta v_J \leq 7$, $\Delta v_{tot} \leq 12$
- Total number of boxes = 9.25e9, feasible boxes = 4.84e4 (5.23e-4%)
- Best objective function = 8.670 (2.84 + 0 + 0 + 0.39 + 0 + 5.42)
- Best objective function with GASP-DA = 10.09, ref. sol. [31] = 8.680

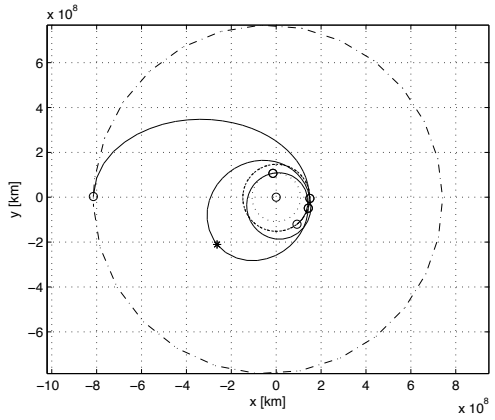


Figure 4.10: Optimal EVEdEJ transfer.

4.5.6 EdVdM

	T_E MJD 2000	t_{EV} days	t_{VM} days
L_b	0	50	200
U_b	1000	400	1000
Δ	50	25	50
EVM	949.4	131.1	765.8

Table 4.7: Time bounds and amplitudes for the EdVdM transfer.

Two DSM are introduced in the EdVdM transfer. This is the case where the two maneuvers lie across a GA. With the absolute variables formulation implemented in GASP-DSM-DA, the functions at the Venus GA, r_P and Δv_V, depend on eight variables. Nevertheless, these functions have been efficiently bounded and the search space has been considerably pruned. The CPU time is 2324 s. We list here below the main features of the EdVdM problem.

- $\Delta v_E \leq 4$, $\Delta v_d \leq 1$, $\Delta v_V \leq 0$, $\Delta v_d \leq 5$, $\Delta v_M \leq 3$, $\Delta v_{tot} \leq 10$
- Total number of boxes = 8.70e9, feasible boxes = 1.13e4 (1.24e-4%)
- Best objective function = 8.167 (2.89 + 0 + 0 + 4.37 + 0.90)
- Best objective function with GASP-DA = 8.522

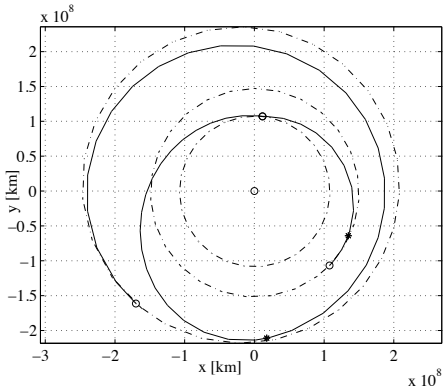


Figure 4.11: Optimal EdVdM transfer.

4.5.7 EVdMdE

	T_E MJD 2000	t_{EV} days	t_{VM} days	t_{ME} days
L_b	3000	25	20	25
U_b	4000	425	420	425
Δ	50	25	50	50
EVME	3985.0	160.9	172.6	427.8

Table 4.8: Time bounds and amplitudes for the EVdMdE transfer.

The EVdMdE problem is stated in Table 4.8. This is a ten-dimensional problem with the search space split in approximately 5e10 small boxes over which the constrained functions are represented. This problem has been solved in 2456 s; the search space is efficiently pruned ($\mathcal{X}$ represents only the 4e-5% of the whole search space). The main features of this problem are listed below, and the optimal solution found is represented in Figure 4.12.

- $\Delta v_E \leq 5$, $\Delta v_V \leq 0$, $\Delta v_d \leq 2$, $\Delta v_M \leq 0$, $\Delta v_d \leq 4$, $\Delta v_E \leq 5$, $\Delta v_{tot} \leq 13$
- Total number of boxes = 4.97e10, feasible boxes = 2.05e4 (4.0e-5%)
- Best objective function = 10.931 (5.38 + 0 + 0 + 0.97 + 3.64 + 0.93)
- Best objective function with GASP-DA = 12.443

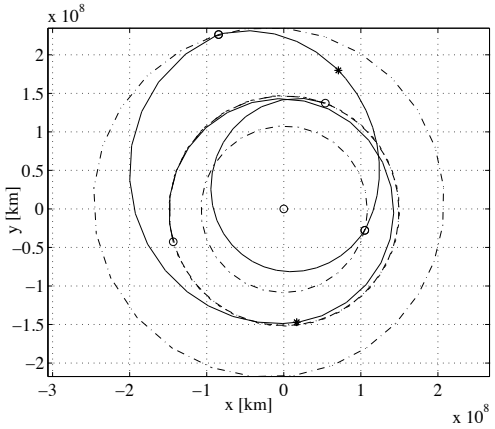


Figure 4.12: Optimal EVdMdE transfer.

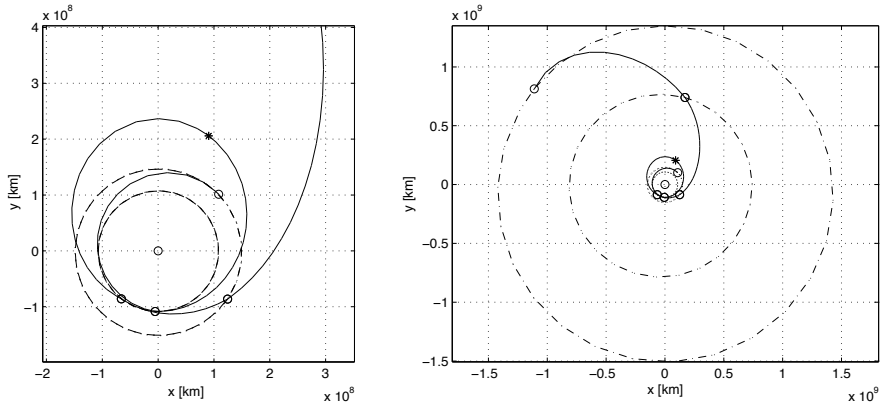
4.5.8 EVdVEJS

	T_E MJD 2000	t_{EV} days	t_{VV} days	t_{VE} days	t_{EJ} days	t_{JS} days
L_b	-1000	80	200	30	400	800
U_b	0	430	500	180	1600	2200
Δ	50	25	25	50	200	200
EVVEJS	-787.0	165.8	427.7	57.7	596.1	2200

Table 4.9: Time bounds and amplitudes for the EVdVEJS transfer.

The problem stated in Table 4.9 represents a Cassini-like transfer with a DSM between the two Venus gravity assists. Even if this problem is difficult, the GASP-DSM-DA algorithm is able to solve it in 210 s. This occurs because the large number of two-dimensional pure planet-to-planet transfers simplify the whole nine-dimensional problem. We constrained $\Delta v_{tot} \leq 13$.

- $\Delta v_E \leq 4$, $\Delta v_V \leq 1$, $\Delta v_d \leq 1$, $\Delta v_V \leq 0$, $\Delta v_E \leq 0$, $\Delta v_J \leq 0$, $\Delta v_S \leq 5$
- Total number of boxes = 3.92e8, feasible boxes = 2281 (1e-3%)
- Best obj. funct. = 8.299 (2.93 + 0.69 + 0.42 + 0 + 0 + 0 + 4.24)
- Best objective function with GASP-DA = 8.619



(a) EVdVE portion. (b) Whole transfer.

Figure 4.13: Optimal EVdVEJS transfer.

4.5.9 EVdVEJdS

We have solved another Cassini-like transfer with two DSM: the first one, as in the previous problem, is located across the two Venus gravity assists; the second maneuver lies in the Jupiter-Sautrn arc. The time domain is analogous to that shown in Table 4.9 but for the last time-of-flight that now is allowed to be $t_{JS} \in [1600, 3000]$. Although this is a twelve-dimensional problem, the GASP-DSM-DA algorithm is able to efficiently prune the solution space as only 1e-6% of the initial domain is feasible (note that the total number of boxes is 1e12). The ability to sample and prune this huge domains is due to the planet-to-planet transfers that simplify the problem and cut the variable dependencies (in the absolute variables formulation). The CPU time is 2000 s. The main results of this problem are listed below. The total cost of the transfer is constrained to $\Delta v_{tot} \leq 11$, whereas both the maneuvers are constrained to $\Delta v_d \leq 1$.

- $\Delta v_E \leq 4$, $\Delta v_V \leq 1$, $\Delta v_d \leq 1$, $\Delta v_V \leq 0$, $\Delta v_E \leq 0$, $\Delta v_J \leq 0$, $\Delta v_S \leq 5$
- Total number of boxes = 1.41e12, feasible boxes = 2.23e4 (1.6e-6%)
- Best obj. funct. = 8.276 (2.78 + 0.89 + 0.38 + 0 + 0 + 0 + 0 + 4.22)
- Best objective function with GASP-DA = 8.619

4.6 Final Remarks

In this chapter we have introduced the DSM into the GASP-DA algorithm. This process gives rise to the GASP-DSM-DA procedure. We have analyzed the problem of introducing the DSM in a DA frame. In particular, we have studied the MGA-DSM transfer formulation with two different strategies. Since the driving requirement was having functions (to represent by Taylor series and to bound) of as few variables as possible, we chose an absolute variables strategy for the representation of the maneuver point. It has been discussed that, with this strategy, functions of at most eight variables arise and the problem complexity is bounded (the problem does not blow-up along the transfer). The worst case occurs only when two consecutive DSM are located across a GA. The result is that the GA variables (pericenter radius and Δv burn) depended both on eight variables.

The GASP-DSM-DA algorithm has been tested to solve several sample problems discussed in the second part of the chapter. We have shown that the developed algorithm is able to solve difficult problems by efficiently pruning

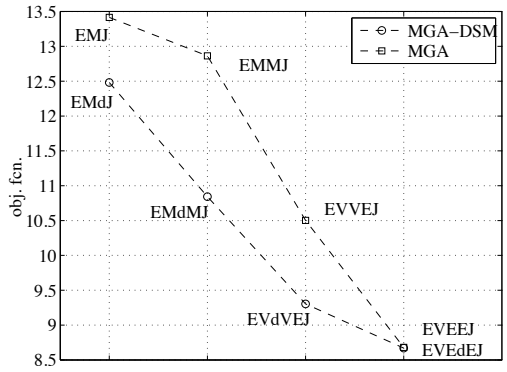


Figure 4.14: Options for a transfer to Jupiter. Both the optimal MGA (squares) and MGA-DSM (circles) solutions are shown.

the search space. The local optimization carried out within each feasible box makes the whole procedure deterministic and gives repeatable results.

The introduction of the DSM deserves a final comment. In Figure 4.14 we have reported, in terms of objective function value, the optimal MGA and MGA-DSM transfers, obtained with GASP-DA and GASP-DSM-DA, respectively, for a mission to Jupiter. It is straightforward that different transfer structures (i.e. different planets sequence) involve different results. It is worth noting that for the cases represented in Figure 4.14, and generally for the MGA-DSM transfers, the introduction of DSM improves the optimal solutions. This means that, in general, the MGA-DSM transfers outperform the pure MGA ones. We also note from the solved cases that a MGA-DSM transfer has a total time-of-flight that is longer than the MGA ones. Finally, the DSM considered in this chapter should not be intended as trajectory correction maneuvers (i.e. small Δv), but rather they are of the order of 1 km/s. The practical implementation of this kind of maneuvers for real missions has not been taken into account.

Chapter 5

Alternative Approach for MGA-DSM Transfers

An alternative approach to optimize MGA-DSM transfers is presented. This approach uses the solution set obtained with GASP-DA and by the subsequent local optimization as an input for the optimization of interplanetary transfers including DSM. As the present approach is based only on GASP-DA, the main drawbacks encountered in DSM space pruning are avoided. As a consequence, the number of maneuvers inserted and the functional dependencies have a little impact on the problem solution. Figure 5.1 resumes the algorithmic flow of the proposed approach.

The chapter is organized as follows. The criterion for selecting the solutions of GASP to be modified by the inclusion of DSM is given first. The DSM modeling and the first guesses generation is then described in detail. Subsequently the overall optimization problem is formulated and the obtained results are discussed. Some final considerations conclude the chapter.

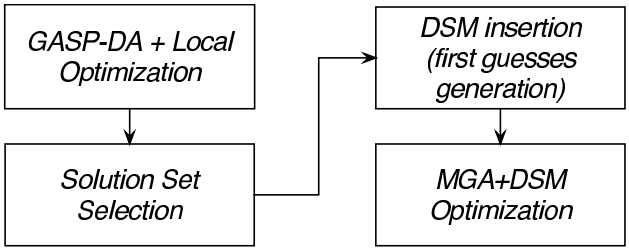


Figure 5.1: Alternative approach algorithmic flow.

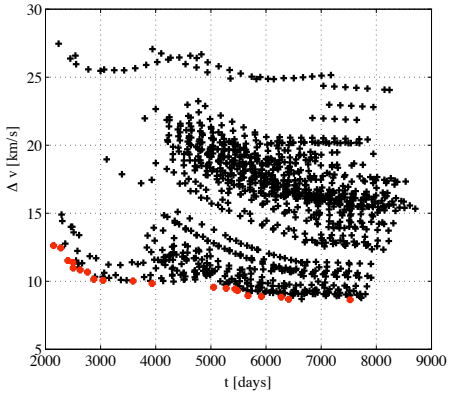


Figure 5.2: Cassini-like transfer solution set: "+"generic solution, "·"non-dominated solutions.

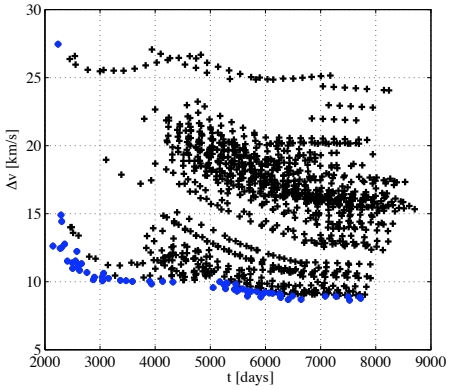


Figure 5.3: Cassini-like transfer solution set: "+"generic solution, "·"solution with $D_l \leq 2$.

5.1 Solution Set Selection

As described in Chapter 3, the outcome of GASP-DA and the subsequent local optimization is a set of solutions. The present approach relies on the consideration that good MGA solutions can be modified to obtain good first guesses for the MGA-DSM transfers optimization. Consequently the first problem that arises is the suitable selection of the solutions.

In Figure 5.2 the solution set obtained for a Cassini-like transfer is plotted, using the + marker, in the time-of-flight vs. Δv plane. It is clear that a large number of solutions is characterized by Δv values that are higher than the optimal one, and then useless for our aim. On the other hand, ranking the solutions using only the Δv information is superficial. The tradeoff between Δv and time-of-flight is an important aspect in mission design indeed.

Based on these considerations the solutions are ranked by means of dominance level, using the time-of-flight and the Δv as performance indexes. Figure 5.4 shows the dominance concept for two generic solution x_a and x_b and performance indexes f_1 and f_2. Any solution plotted in Figure 5.2 can be identified by the pair $(\Delta v, t)$ and its dominance level value D_l. A $D_l = 0$ characterizes a non-dominated solution, $D_l = 1$ a single dominated solution, and so forth.

The red dots in Figure 5.2 shows the non-dominated solution set for the Cassini-like transfer, i.e. all the solution with $D_l = 0$. As the number of non-dominated solutions could be quite small, and also solutions close to the Pareto front can yield interesting results, it is decided to choose as candidate first guesses all the solutions with $D_l \leq 2$. The blue dots in Figure 5.3 show

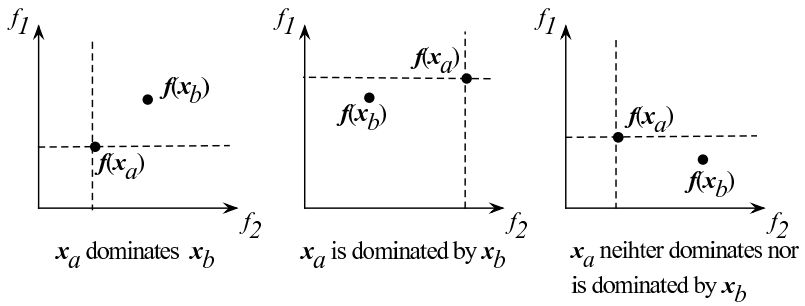


Figure 5.4: Dominance relations between solutions.

this set for the Cassini-like transfer.

5.2 DSM modeling

The DSM model adopted for the present approach is based on the model described in Vasile and De Pascale [31] with the only difference that a Δv is allowed at the pericenter of the GA. Such model, detailed below, is more suitable for the present approach, as the dependency problem described in Chapter 4 is now irrelevant. In fact, the latter issue arises only when the problem of DSM space pruning is addressed, whereas here only the GASP is necessary.

First of all it has to be stressed that a DSM in the first arc after the launch is not allowed as the feasibility of the departure Δv is already assured by the cutoff value considered in the GASP phase. The only limitation given by this choice is a lower flexibility in dealing with resonance orbits with the departure planet but, on the other hand, a reduction of the optimization variables is gained. Concluding, a DSM can only be placed on arcs following a GA, and that's the case we discuss here. The reader can use Figure 5.5 as visual aid. Note that arcs without DSM are modeled as described in Chapter 3.

The absolute incoming velocity $\boldsymbol{v}^-$ at the sphere of influence of the planet preceding the DSM is given by the solution of a Lambert's problem. The incoming relative velocity is simply

$$\boldsymbol{v}_\infty^- = \boldsymbol{v}^- - \boldsymbol{v}_p, \tag{5.1}$$

where $\boldsymbol{v}_p$ is the planet's velocity. The introduction of an auxiliary angle η is then necessary to define the plane of the hyperbolic passage, used to rotate the relative velocity vector. This angle is necessary as in the linked-conic approximation the point where the interplanetary trajectories pierces the

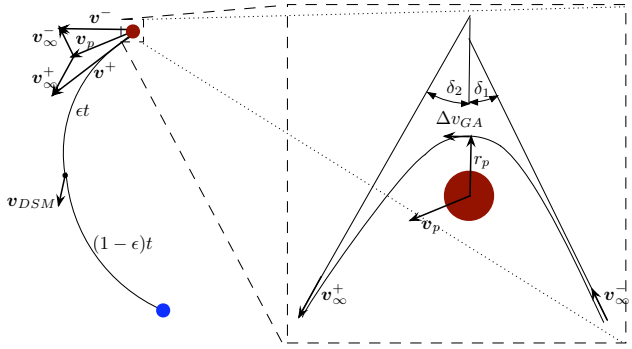


Figure 5.5: DSM model: heliocentric and planetary view.

sphere of influence is not defined, thus the plane where the planetocentric hyperbolic motion occurs remains undetermined. For a detailed definition of this angle see Vasile and De Pascale [31].

The first part of the overall rotation angle δ_1 can be computed, once the hyperbola pericenter radius r_p is chosen, by

$$\begin{aligned} e_1 &= 1 + \frac{r_p {v_\infty^-}^2}{\mu} \\ \delta_1 &= \text{asin}(1/e_1), \end{aligned} \tag{5.2}$$

where e_1 is the eccentricity of the first hyperbolic arc and μ the planet's gravitational parameter. A corrective Δv_{GA} aligned with the velocity vector is allowed at the hyperbola pericenter, thus inserting the spacecraft into a second coplanar hyperbolic arc. Based on the Δv_{GA}, the second part of the rotation angle δ_2 is computed (using the equivalent equation of 5.2) as well as the relative velocity at the exit of the sphere of influence $\boldsymbol{v}_\infty^+$. The latter velocity vector is easily obtained rotating the vector $\boldsymbol{v}_\infty^-$ of an angle $\delta = \delta_1 + \delta_2$ around a vector normal to the orbital plane. The absolute velocity at the exit of the sphere of influence is simply obtained through

$$\boldsymbol{v}^+ = \boldsymbol{v}_\infty^+ + \boldsymbol{v}_p. \tag{5.3}$$

The DSM location is obtained by analytically propagating the spacecraft state for the fraction ϵ of total transfer time t. Finally the Δv_{DSM} is computed by means of a Lambert's arc connecting the DSM position vector and the position vector of the successive planet. Using the described formulation the inclusion of a DSM requires four additional variables, namely $[\eta, r_p, \Delta v, \epsilon]$.

5.3 First Guess Generation

The next step is to modify the selected MGA solutions by including DSM in order to generate good first guesses for the MGA-DSM transfer optimization. For each selected MGA, a MGA-DSM first guess is obtained and optimized.

Any selected trajectory involving n_P planets is completely defined by n_P epochs, and the associated total Δv is given by the sum of the final and arrival Δv and the $n_P - 2$ Δv_{GA}. Based on these Δv each trajectory is modified to include the DSM. More specifically whenever a Δv_{GA} is greater than a chosen threshold Δv^*_{GA} a DSM is inserted in the subsequent arc of the trajectory. Thus a high Δv_{GA} is considered as an index of the DSM utility, and the DSM are used to reduce, and possibly cancel, all the Δv_{GA}.

Once the Δv^*_{GA} (one for each GA) are chosen, the structure of each first guess is determined; this means that the sequence of arcs with and without DSM is given. It's worth noting that, as the selected MGA solutions are characterized by different values of Δv_{GA}, several different first guess structures can be obtained for the same transfer at hand.

For each DSM added, the values for the four additional variables $[\eta, r_p, \Delta v, \epsilon]$ must be guessed in order to generate the first guess trajectory. The values of η and r_p are kept equal to those of the homologous MGA trajectories. The values of the remaining variables are:

$$\begin{array}{lcl} \Delta v_{GA} & = & 0 \\ \epsilon & = & 0.5. \end{array} \tag{5.4}$$

This choice means that in the first guess solutions the DSM are used to cancel the pericenter burns and the DSM occur at half of each transfer.

5.4 Problem Formulation

Once the solution set selection and the first guesses generation is performed, the formulation of DSM transfer optimization problem is straightforward. For a trajectory with n_P planets and n_D DSM the $n_P + 4n_D$ optimization variables are n_P epochs defining the planets configurations and n_D subsets $[\eta, r_p, \Delta v, \epsilon]$.

The function to be minimized is the overall Δv which includes also the Δv_{DSM}. The nonlinear constraints on pericenter radii are imposed only for those GA which are not followed by a DSM, whereas for the others they reduce to simple box constraints, as r_p are optimization variables. The bounds

on the DSM variables are:

$$\begin{array}{rcccl} 0 & \leq & \eta & \leq & 2\pi \\ 0 & \leq & 1/r_p & \leq & 1/r_p^* \\ -\Delta v_{GA}^* & \leq & \Delta v_{GA} & \leq & \Delta v_{GA}^* \\ 0.01 & \leq & \epsilon & \leq & 0.99. \end{array} \tag{5.5}$$

Note that r_p^* is the minimum allowed pericenter altitude which differs from planet to planet and Δv_{GA}^* is the threshold used to decide whether a DSM is required.

5.5 Test Cases

In this section some of test cases analyzed in the previous chapter are addressed. The goal is to enlighten the capability of the method to replicate the results obtained with the DSM space pruning using only the GA space pruning. For each transfer a table with the search space definition and the optimal variable vector (for the departure epoch, transfer times, and ϵ) is given, whereas the Δv values are summarized for all the cases in Table 5.3. In the transfer definition each capital letter corresponds to a planet (e.g. E stands for Earth, M for Mars and so forth) and the "d" to a deep space maneuver. The third column, labeled as Δv_{GA-DSM}, shows the sequence of Δv associated either to gravity assists or deep space maneuvers. For each gravity assist, the minimum pericenter radius is 1.05 times the radius of the encountered planet.

5.5.1 EVM

The first test case is an Earth-Venus-Mars transfer. As the transfer includes only one GA, a single DSM is allowed. Two test cases, labeled as EVM_1 and

Table 5.1: Time bounds and optimal solution for EVM_1.

	T_E MJD 2000	t_{EV} days	t_{VM} days
L_b	0	100	100
U_b	2000	1200	2000
EVM_1	1536.7	111.5	1250.7 $\epsilon = 0.55$

Table 5.2: Time bounds and optimal solution for EVM_2.

	T_E MJD 2000	t_{EV} days	t_{VM} days
L_b	1000	100	100
U_b	6000	500	1000
EVM_2	1150.1	171.3	779.2 $\epsilon = 0.34$

EVM_2, characterized by two different sets of bounds for the launch windows and the transfer times are analyzed. This choice allows us to compare the results with those obtained by Izzo [23] and with the solutions obtained without DSM, respectively. As it can be seen from Figure 5.6 and Figure 5.7 it is evident that the DSM is used to inject the spacecraft into an orbit close to the Mars' one, thus minimizing the final Δv. In both cases the effect of the DSM is to delete the Δv_{GA} as shown by the third column of Table 5.3.

The optimal Δv for the EVM_1 is about 7.75 km/s which is lower than the 8.15 km/s obtained by Izzo [23]. (It has to be remarked that the difference could be due to the different minimum pericenter allowed for the Mars gravity assist). The second transfer shows that the inclusion of a DSM allows a 450 m/s saving with the respect to a GA trasfer.

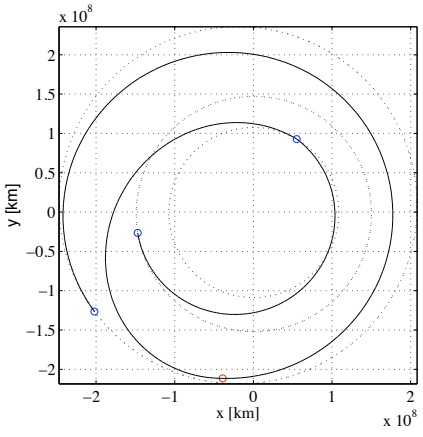


Figure 5.6: EVM_1 optimal trajectory.

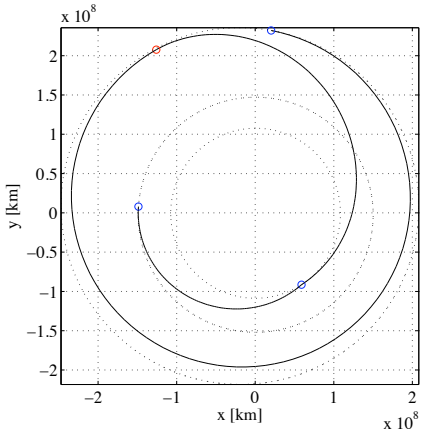


Figure 5.7: EVM_2 optimal trajectory.

Table 5.3: Optimal transfers Δv [km/s].

	Δv_{Dep}	Δv_{GA-DSM}	Δv_{Arr}	Δv_{TOT}
$EVdM_1$	3.428	0 + 3.521	0.802	7.752
$EVdM_2$	2.953	0 + 3.904	1.225	8.081
EMdJ	3.930	0 + 4.046	4.418	12.394
EVdMdE	5.465	0 + 0.705 + 0 + 4.211	0.493	10.874
EMdMdJ	3.260	0 + ≈0 + 0 + 3.195	4.513	10.968
EVEdEJ	2.834	0 + 0 + 0.426 + 0	5.426	8.687
EVdVEJS	2.933	0.693 + 0.428 + 0+ 0 + 0	4.244	8.588
EVdVEJdS	3.574	0.330 + 0.330 + 0 + 0 + 0 + 2.570	0.798	7.600

5.5.2 EMJ

The second test case concerns an Earth-Mars-Jupiter transfer. The problem definition and the optimal solution found in terms of launch epoch and transfer times is given in Table 5.4.

The trajectory of Figure 5.8 enlightens the usefulness of the DSM, used to replicate a Mars resonance. After the Mars GA, the spacecraft is placed into a near 1:2 resonant orbit with Mars, with the difference that the second gravity assist of the Martian planet (which is not allowed by the structure of the problem) is replaced with the DSM. The optimal solution has a total Δv of 12.39 km/s which is slightly better than the 12.48 km/s found with the previous approach. Furthermore the introduction of the DSM allows to save about 1 km/s with respect to the MGA trajectory.

Table 5.4: Time bounds and optimal solution for EMJ.

	T_E MJD 2000	t_{EM} days	t_{MJ} days
L_b	1000	100	100
U_b	6000	1200	2000
EMJ	2070.6	403.4	2000 $\epsilon = 0.45$

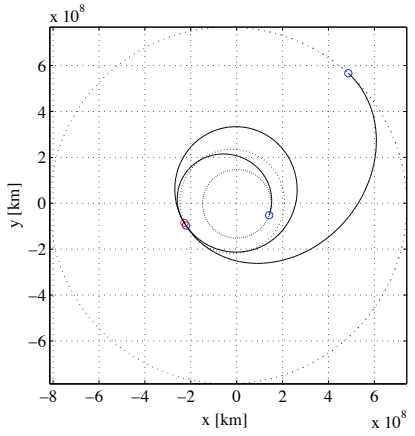


Figure 5.8: EMJ optimal trajectory.

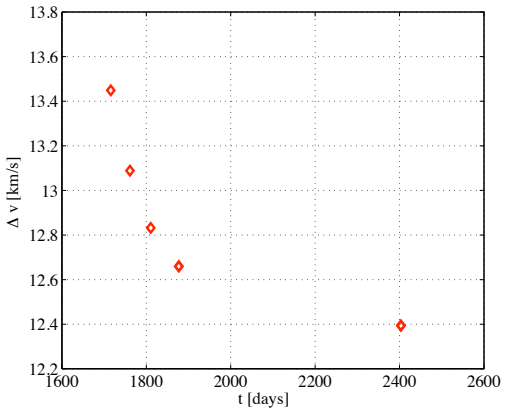


Figure 5.9: EMJ transfer, set of non-dominated solutions.

An interesting result that can be obtained with the present approach is underlined by Figure 5.9. As the solution set selection described in Section 5.1 is based on the dominance level, the result of the DSM likely reproduces the Pareto front of the problem when the total Δv and the time-of-flight are considered as performance indexes. This result can be useful in the preliminary design of the mission, when the tradeoff between transfer time and Δv is of particular interest.

5.5.3 EVME

The third case involves two GA, thus two DSM can be included at most. The optimal solution found exploits both the DSM to reduce the overall Δv. The first one is a small corrective maneuver, whereas the second injects the

Table 5.5: Time bounds and optimal solution for EVME.

	T_E MJD 2000	t_{EV} days	t_{VM} days	t_{ME} days
L_b	3000	25	20	25
U_b	4000	525	520	525
EVME	3990.6	158.3	207.1 $\epsilon = 0.54$	442.4 $\epsilon = 0.48$

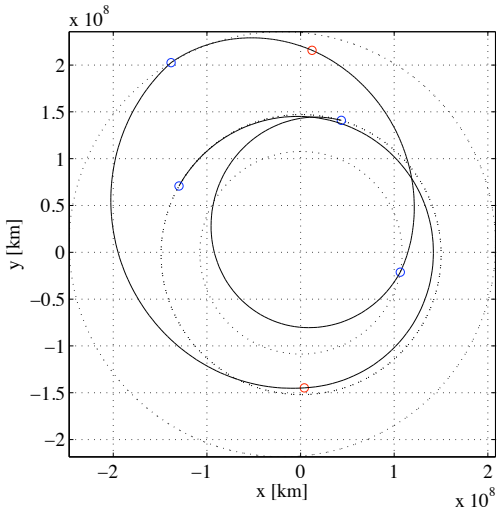


Figure 5.10: EVME optimal trajectory.

spacecraft into an orbit close to the Earth's one, thus minimizing the final Δv (see Figure 5.10). The solution is qualitatively the same found in Chapter 5 although the overall Δv is slightly smaller.

5.5.4 EMMJ

A transfer to Jupiter exploiting a double Mars GA is now considered. Also in this case the optimal structure of the transfer includes both the allowed DSM, although the first one is almost negligible. On the other hand (see Figure 5.11) the second maneuver employs a huge Δv, which is used to realize almost a Hohmann transfer to Jupiter. Also in this case the solution found is qualitative equal to the one shown in the previous chapter and to that of Vasile and De Pascale [31], characterized by a Δv of 11.05 km/s.

Table 5.6: Time bounds and optimal solution for EMMJ.

	T_E MJD 2000	t_{EM} days	t_{MM} days	t_{MJ} days
L_b	3650	80	330	1000
U_b	5500	430	830	2000
EMMJ	5188.1	376.0	687.0 $\epsilon = 0.21$	1228.3 $\epsilon = 0.13$

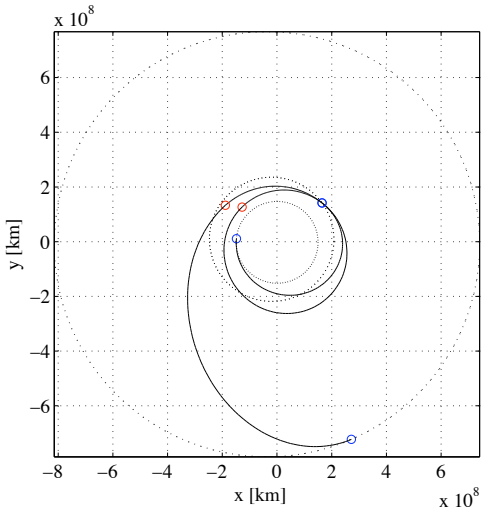


Figure 5.11: EMMJ optimal trjectory.

5.5.5 EVEEJ

The last Jupiter transfer analyzed exploits a Venus and a double Earth gravity assist. It's worth noting that the optimal solution found uses only one of the three possible DSM, in particular to improve the Earth-Earth sequence. Like in all the tests analyzed so far, the optimal solution exhibits ballistic gravity assists, i.e. no pericenter Δv is considered.

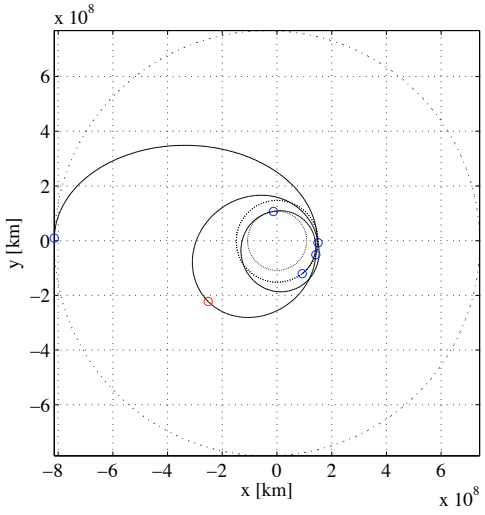


Figure 5.12: EVEEJ optimal trajectory.

Table 5.7: Time bounds and optimal solution for EVEEJ.

	T_E MJD 2000	t_{EV} days	t_{VE} days	t_{EE} days	t_{EJ} days
L_b	3650	80	80	80	600
U_b	7300	430	830	830	2000
EVEEJ	3863.1	128.8	288	712.9 $\epsilon = 0.59$	1064

5.5.6 EVVEJS

The last test case is a Cassini-like transfer, in particular two different time-of-flight are considered for the Jupiter-Saturn arc (see Table 5.8 and 5.9). It's worth noting that when the shorter transfer time is considered (EVVEJS$_1$) the optimal structure includes only one DSM in between the two Venus GA. On the other hand, when the longer transfer time is considered (EVVEJS$_2$), a DSM in the JS arc effectively reduces the transfer Δv. This particular solution enlightens the capability of the algorithm to optimize the structure of the transfer, i.e. the number and the sequence of DSM.

The solution EVVEJS$_1$ shows similar departure epoch and transfer times of that described in Chapter 5, unless it requires an additional Δv of 300 m/s.

The second solution makes clear that the price to pay for a sensible reduction of the overall Δv is an enormous increase in the mission duration, thus making the optimal solution unpractical. Note, from Table 5.3, that in both the solutions the possibility of using a small impulsive burn at the first Venus GA is exploited. This aspect does not allow us to compare these

Table 5.8: Time bounds and optimal solution for EVVEJS$_1$.

	T_E MJD 2000	t_{EV} days	t_{VV} days	t_{VE} days	t_{EJ} days	t_{JS} days
L_b	-1000	80	200	30	400	800
U_b	0	430	500	180	1600	2200
EVVEJS$_1$	-788.9	176.9	424.3	53.3 $\epsilon = 0.53$	589.8	2200

Table 5.9: Time bounds and optimal solution for EVVEJS$_2$.

	T_E MJD 2000	t_{EV} days	t_{VV} days	t_{VE} days	t_{EJ} days	t_{JS} days
L_b	-1000	80	200	30	400	800
U_b	0	430	500	180	1600	6000
EVVEJS$_2$	-792.7	182.1	426.3	52.7 $\epsilon = 0.55$	768	6000 $\epsilon = 0.47$

results with those available in literature.

5.6 Final Remarks

In the present chapter an alternative approach has been introduced for the MGA-DSM transfer optimization. Being based only on MGA space pruning, all the complications arising in the DSM space pruning (i.e. dimensionality, functional dependencies, computational time etc) are avoided. Furthermore the computational times are kept as low as those of GASP-DA, thus allowing to easily manage an increased number of maneuvers. Moreover the solutions obtained with this method are comparable to those obtained in the previous chapter, making it a promising approach for MGA-DSM transfer optimization. It's worth noting that as only GASP is used here, the user is not forced

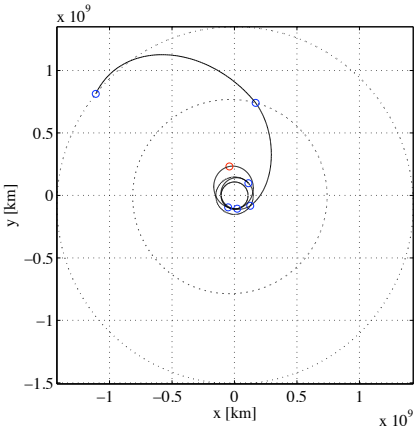


Figure 5.13: EVVEJS$_1$ optimal trajectory.

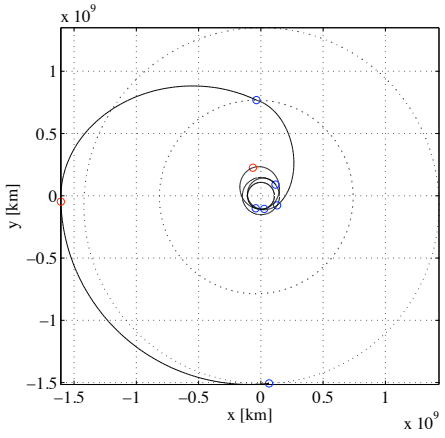


Figure 5.14: EVVEJS$_2$ optimal trajectory.

to select the cutoff values for the DSM; in this way one of the pitfall of the GASP-DSM approach is avoided.

Chapter 6

Validated Optimization of MGA Transfers

This chapter presents the attempt to solve the MGA transfer optimization in a validated manner. The tool used to tackle this problem is the validated global optimizer COSY-GO [15] which is based on the method of Remainder-enhanced Differential Algebra (RDA), also known as Taylor Model (TM) method. As this tool delivers the mathematically proven global optimum of the problem at hand, the potentials of its application to trajectory optimization problems are evident.

Some notes on RDA method are presented first based on the PhD thesis of Kyoko Makino [26]. We then give some hints on how these methods can be used to develop a validated global optimizer, as done in COSY-GO. The global optimization of Earth-Mars and a Earth-Venus-Mars transfers is then addressed and the achieved results critically discussed.

6.1 Differential Algebra and Interval Arithmetic

While DA methods can provide the derivatives of functional dependencies and solutions of ODEs to high orders, in a rigorous sense they fail to provide information about the range of the function. A simple example that dramatically illustrates this phenomenon is the function shown in Figure 6.1

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ \exp(-1/x^2) & \text{else.} \end{cases} \tag{6.1}$$

The value of the function and all the derivatives at $x = 0$ are 0. Thus the Taylor polynomial at the reference point $x = 0$ is just the constant

0. In particular, this also implies that the Taylor expansion of f converges everywhere, but it fails to agree with $f(x)$ everywhere but at $x = 0$.

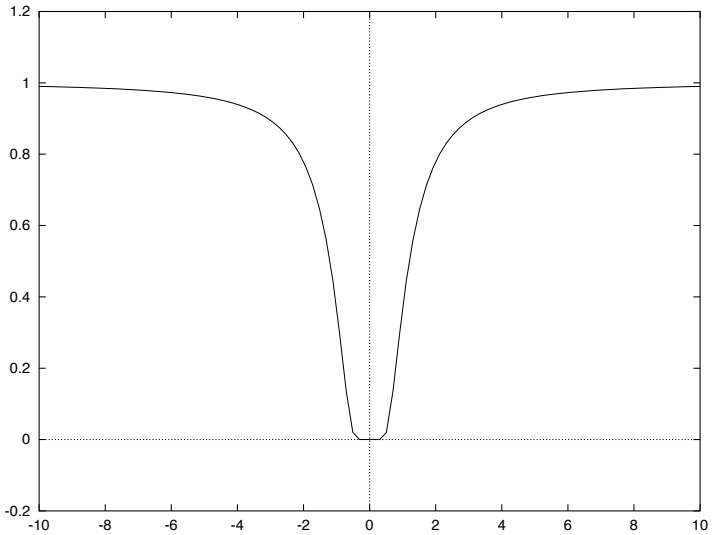


Figure 6.1: Function $f(x) = \exp(-1/x^2)$ if $x \neq 0$; 0 else, and its Taylor polynomial, which vanishes identically.

For the purpose of bounding functional dependencies, the methods of interval arithmetic provides a conceptual contrast. Both extended domains of numbers as well as individual real numbers are represented via rigorous inclusions of floating point intervals. Arithmetic operations are introduced on intervals such that for any numbers in the intervals, a real arithmetic operation on the two numbers always leads to a result that is contained in the interval obtained from the corresponding arithmetic operation on the intervals. Table 6.1 lists some elementary properties of interval arithmetic.

By evaluating a function in interval arithmetic, it is thus possible to carry rigorous bounds information through the operations, and in the end obtain rigorous bounds of the function. However, while reasonably fast in practice, interval methods have some severe disadvantages, which limits their applicability for complicated functions. First, the width of resulting intervals scales with the width of the original intervals; and second, artificial blow-up usually occurs in extended calculations. Another practical limitation arises if scanning with small intervals is needed in the case of multiple dimensions because of the fast increase of the computational expense.

To illustrate the blow-up phenomenon with a trivial example, we consider the interval $I = [a, b]$, which has the width $b - a$. We compute the addition

Table 6.1: Elementary properties of interval arithmetic; $I_1 = [a_1, b_1], I_2 = [a_2, b_2]$.

$I_1 + I_2 = [a_1 + a_2, b_1 + b_2]$
$-I_1 = [-b_1, -a_1]$
$I_1 \cdot I_2 = [\min(a_1a_2, a_1b_2, b_1a_2, b_1b_2), \max(a_1a_2, a_1b_2, b_1a_2, b_1b_2)]$
If $0 \notin I_1$, $1/I_1 = [1/b_1, 1/a_1]$

of I to itself and its subtraction from itself:

$$\begin{aligned} I + I &= [a, b] + [a, b] = [a + a, b + b] = [2a, 2b] \\ I - I &= [a, b] - [a, b] = [a, b] + [-b, -a] = [a - b, b - a]. \end{aligned}$$

In both cases the resulting width is $2(b - a)$, which is twice the original width, although we know that regardless of what unknown quantity x is characterized by I, certainly $x - x$ should equal zero.

Polynomials under their conventional operations form a commutative algebra with unity. However, interval arithmetic does not have even a group structure for either addition or multiplication, since intervals with nonzero width have no inverses. Furthermore, instead of distributivity, we have the sub-distributivity,

$$I_1 \cdot (I_2 + I_3) \subseteq I_1 \cdot I_2 + I_1 \cdot I_3. \tag{6.2}$$

The concepts of interval methods are discussed at length and in depth in several sources, as in [29].

6.2 Remainder-enhanced Differential Algebraic Operations

RAD is a method that combines the advantage of rigor of the interval approach, while largely avoiding the blow-up problem through the use of DA techniques. The key idea is to describe the bulk of the functional dependence

through a Taylor polynomial, and bound the deviation of the original function from the Taylor polynomial by an interval. In this endeavor, the Taylor theorem plays an important role.

Theorem (Taylor): *Suppose that a function* $f : [\vec{a}, \vec{b}] \subset R^v \to R$ *is* $(n+1)$ *times continuously partially differentiable on* $[\vec{a}, \vec{b}]$. *Assume* $\vec{x}_0 \in [\vec{a}, \vec{b}]$. *Then for each* $\vec{x} \in [\vec{a}, \vec{b}]$, *there is* $\theta \in R$ *with* $0 < \theta < 1$ *such that*

$$\begin{aligned} f(\vec{x}) &= \sum_{\nu=0}^{n} \frac{1}{\nu!} \left((\vec{x} - \vec{x}_0) \cdot \vec{\nabla} \right)^{\nu} f(\vec{x}_0) + \\ &+ \frac{1}{(n+1)!} \left((\vec{x} - \vec{x}_0) \cdot \vec{\nabla} \right)^{n+1} f\left(\vec{x}_0 + (\vec{x} - \vec{x}_0)\theta \right), \end{aligned}$$

where the partial differential operator $\left(\vec{h} \cdot \vec{\nabla} \right)^k$ *operates as*

$$\left(\vec{h} \cdot \vec{\nabla} \right)^k = \sum_{\substack{0 \le i_1, \cdots, i_v \le k \\ i_1 + \cdots + i_v = k}} \frac{k!}{i_1! \cdots i_v!} h_1^{i_1} \cdots h_v^{i_v} \frac{\partial^k}{\partial x_1^{i_1} \cdots \partial x_v^{i_v}}.$$

Depending on the situation at hand, the remainder term also can be cast into a variety of well-known other forms.

Taylor's theorem allows a quantitative estimate of the error that is to be expected when approximating a function by its Taylor polynomial. Furthermore, it even offers a way to obtain bounds for the error in practice, based on bounding the $(n+1)$st derivative, a method that has sometimes been employed in interval calculations.

Roughly speaking, Taylor's theorem suggests that in many cases the error decreases with the order as the width of the interval raised to the order being considered, and its practical use is often connected to this observation. However, certain examples illustrate that this behavior does not have to occur; one such example is (6.1) in the previous section.

For notational convenience, we introduce a parameter α to describe the details of a given Taylor expansion, namely, the order of the Taylor polynomial n, and the reference point of expansion $\vec{x}_0$. For the purpose to derive bounds for the remainder, it is also necessary to include the domain interval $[\vec{a}, \vec{b}]$ on which the function is to be considered; altogether, we have

$$\alpha = (n, \vec{x}_0, [\vec{a}, \vec{b}]). \tag{6.3}$$

We now write an $(n+1)$ times continuously partially differentiable function $f : [\vec{a}, \vec{b}] \subset R^v \to R$ as a sum of its Taylor polynomial $P_{\alpha,f}$ of nth order and a remainder $\varepsilon_{\alpha,f}$ as

$$f(\vec{x}) = P_{\alpha,f}(\vec{x} - \vec{x}_0) + \varepsilon_{\alpha,f}(\vec{x} - \vec{x}_0),$$

where $\varepsilon_{\alpha,f}(\vec{x}-\vec{x}_0)$ is continuous (even continuously differentiable) on the domain interval and thus bounded. Let the interval $I_{\alpha,f}$ be such that

$$\forall \vec{x} \in [\vec{a},\vec{b}], \quad \varepsilon_{\alpha,f}(\vec{x}-\vec{x}_0) \in I_{\alpha,f}.$$

Then

$$\forall \vec{x} \in [\vec{a},\vec{b}], \quad f(\vec{x}) \in P_{\alpha,f}(\vec{x}-\vec{x}_0) + I_{\alpha,f}. \tag{6.4}$$

Because of the special form of the Taylor remainder term $\varepsilon_{\alpha,f}$, in practice the remainder usually decreases as $|\vec{x}-\vec{x}_0|^{n+1}$. Hence, if $|\vec{x}-\vec{x}_0|$ is chosen to be small, the interval $I_{\alpha,f}$, which from now on we refer to as the interval remainder bound, can become so small that even the effect of considerable blow-up is not detrimental. The set $P_{\alpha,f}(\vec{x}-\vec{x}_0)+I_{\alpha,f}$ containing f consists of the Taylor polynomial $P_{\alpha,f}(\vec{x}-\vec{x}_0)$ and the interval remainder bounds $I_{\alpha,f}$. We say a pair $(P_{\alpha,f}, I_{\alpha,f})$ of a Taylor polynomial $P_{\alpha,f}(\vec{x}-\vec{x}_0)$ and an interval remainder bounds $I_{\alpha,f}$ is a Taylor model of f if and only if (6.4) is satisfied. In this case, we denote the Taylor model by

$$T_{\alpha,f} = (P_{\alpha,f}, I_{\alpha,f}).$$

We call n the order of the Taylor model, $\vec{x}_0$ the reference point of the Taylor model, $[\vec{a},\vec{b}]$ the domain interval of the Taylor model, and α the parameter of the Taylor model.

In the following, we develop tools that allow us to efficiently calculate Taylor models for all functions representable on a computer. The key is to begin with the Taylor model for the identity function, which is trivial, and then successively build up Taylor models for the total function from its pieces. This requires methods to determine Taylor models for sums and products from those of the summands or factors, as well as from intrinsics applied to functions with known Taylor model.

6.2.1 Addition and Multiplication

In this subsection, we discuss how a Taylor model of a sum or product of two functions can be obtained from the Taylor models of the two individual functions. This represents the first step toward the computation of Taylor models for any function that can be represented on a computer.

Let the functions $f, g : [\vec{a},\vec{b}] \subset R^v \to R$ have Taylor models

$$T_{\alpha,f} = (P_{\alpha,f}, I_{\alpha,f}) \quad \text{and} \quad T_{\alpha,g} = (P_{\alpha,g}, I_{\alpha,g}),$$

which entails that

$$\begin{array}{lrcll} \forall \vec{x} \in [\vec{a}, \vec{b}], & f(\vec{x}) & \in & P_{\alpha,f}(\vec{x} - \vec{x}_0) + I_{\alpha,f} & \text{and} \\ & g(\vec{x}) & \in & P_{\alpha,g}(\vec{x} - \vec{x}_0) + I_{\alpha,g}. & \end{array}$$

Then it is straightforward to obtain a Taylor model for $f + g$; in fact, for any $\vec{x} \in [\vec{a}, \vec{b}]$,

$$\begin{aligned} f(\vec{x}) + g(\vec{x}) &\in (P_{\alpha,f}(\vec{x} - \vec{x}_0) + I_{\alpha,f}) + (P_{\alpha,g}(\vec{x} - \vec{x}_0) + I_{\alpha,g}) \\ &= (P_{\alpha,f}(\vec{x} - \vec{x}_0) + P_{\alpha,g}(\vec{x} - \vec{x}_0)) + (I_{\alpha,f} + I_{\alpha,g}), \end{aligned}$$

so that a Taylor model $T_{\alpha,f+g}$ for $f + g$ can be obtained via

$$P_{\alpha,f+g} = P_{\alpha,f} + P_{\alpha,g} \quad \text{and} \quad I_{\alpha,f+g} = I_{\alpha,f} + I_{\alpha,g}. \tag{6.5}$$

Thus we define

$$T_{\alpha,f} + T_{\alpha,g} = (P_{\alpha,f} + P_{\alpha,g}, I_{\alpha,f} + I_{\alpha,g}),$$

and we obtain that $T_{\alpha,f} + T_{\alpha,g} = (P_{\alpha,f+g}, I_{\alpha,f+g})$ is a Taylor model for $f + g$. Note that the above addition of Taylor models is both commutative and associative.

The goal in defining a multiplication of Taylor models is to determine a Taylor model for $f \cdot g$ from the knowledge of the Taylor models $T_{\alpha,f}$ and $T_{\alpha,g}$ for f and g. Observe that for any $\vec{x} \in [\vec{a}, \vec{b}]$,

$$\begin{aligned} f(\vec{x}) \cdot g(\vec{x}) &\in (P_{\alpha,f}(\vec{x} - \vec{x}_0) + I_{\alpha,f}) \cdot (P_{\alpha,g}(\vec{x} - \vec{x}_0) + I_{\alpha,g}) \\ &\subseteq P_{\alpha,f}(\vec{x} - \vec{x}_0) \cdot P_{\alpha,g}(\vec{x} - \vec{x}_0) \\ &\quad + P_{\alpha,f}(\vec{x} - \vec{x}_0) \cdot I_{\alpha,g} + P_{\alpha,g}(\vec{x} - \vec{x}_0) \cdot I_{\alpha,f} + I_{\alpha,f} \cdot I_{\alpha,g}. \end{aligned}$$

Note that $P_{\alpha,f} \cdot P_{\alpha,g}$ is a polynomial of $(2n)$th order. We split it into the part of up to nth order, which agrees with the Taylor polynomial $P_{\alpha,f\cdot g}$ of order n of $f \cdot g$, and the extra polynomial P_e, so that we have

$$P_{\alpha,f}(\vec{x} - \vec{x}_0) \cdot P_{\alpha,g}(\vec{x} - \vec{x}_0) = P_{\alpha,f\cdot g}(\vec{x} - \vec{x}_0) + P_e(\vec{x} - \vec{x}_0). \tag{6.6}$$

A Taylor model for $f \cdot g$ can now be obtained by finding a bound interval for all the terms except $P_{\alpha,f\cdot g}$. For this purpose, let $B(P)$ be bounds of the polynomial $P : [\vec{a}, \vec{b}] \subset R^v \to R$, namely,

$$\forall \vec{x} \in [\vec{a}, \vec{b}], \quad P(\vec{x}) \in B(P).$$

Apparently the efficient practical determination of $B(P)$ is not completely trivial; depending on the order and number of variables, different strategies

may be employed, ranging from analytical estimates to interval evaluations. However, thanks to the specific circumstances, the occurring contributions are very small, and even moderate overestimation is not critical. Various methods for determining $B(P)$ will be discussed below.

Altogether, interval remainder bounds for $f \cdot g$ can be found via

$$I_{\alpha,f\cdot g} = B(P_e) + B(P_{\alpha,f}) \cdot I_{\alpha,g} + B(P_{\alpha,g}) \cdot I_{\alpha,f} + I_{\alpha,f} \cdot I_{\alpha,g}. \tag{6.7}$$

Thus we define $T_{\alpha,f} \cdot T_{\alpha,g} = (P_{\alpha,f\cdot g}, I_{\alpha,f\cdot g})$, and obtain that $T_{\alpha,f} \cdot T_{\alpha,g}$ is a Taylor model for $f \cdot g$. Note that commutativity of multiplication holds, $T_{\alpha,f} \cdot T_{\alpha,g} = T_{\alpha,g} \cdot T_{\alpha,f}$, while multiplication is not generally associative, and also distributivity does not generally hold.

While the idea of Taylor models of constant functions is almost trivial, we mention it for the sake of completeness. For a constant function $f(\vec{x}) \equiv t$, the Taylor model of f is

$$T_{\alpha,f} \equiv T_{\alpha,t} = (P_{\alpha,t}, I_{\alpha,t}) = (t, [0,0]).$$

Having introduced addition and multiplication as well as scalar multiplication, we can compute any polynomial of a Taylor model. Let $Q(f)$ be a polynomial of a function f, that is, $Q(f) = t_0 + t_1 f + t_2 f^2 + \cdots + t_k f^k$. In practice it is useful to evaluate $Q(f)$ via Horner's scheme,

$$Q(f) = t_0 + f \cdot \Big(t_1 + f \cdot \Big(t_2 + f \cdot (\cdots (t_{k-1} + f \cdot t_k) \cdots) \Big) \Big), \tag{6.8}$$

in order to minimize operations. Furthermore, Horner's scheme is often of advantage for interval related arithmetic because of the sub-distributivity (6.2) of interval arithmetic. Assume that we have already found the Taylor model of the function f to be $T_{\alpha,f} = (P_{\alpha,f}, I_{\alpha,f})$. Then, using additions and multiplications of Taylor models described above, we can compute a Taylor model for the function $Q(f)$ via

$$T_{\alpha,Q(f)} = \left(P_{\alpha,Q(f)}, I_{\alpha,Q(f)}\right).$$

6.2.2 Intrinsic Functions

In the preceding subsection, we showed how Taylor models for sums and products of functions can be obtained from those of the individual functions. The computation led to the definition of addition and multiplication of Taylor models. Here we study the computation of Taylor models for intrinsic functions, including the reciprocal applied to a given function f from the Taylor model of f.

The key idea is to employ Taylor's theorem of the function under consideration: However, in order to ensure that the resulting remainder term yields a small remainder interval and does not contribute anything to the Taylor polynomial, several additional steps are necessary.

Let us begin the study with the **exponential** function. Assume that we have already found the Taylor model of the function f to be $T_{\alpha,f} = (P_{\alpha,f}, I_{\alpha,f})$. Write the constant part of the function f around $\vec{x}_0$ as $c_{\alpha,f}$, which agrees with the constant part of the Taylor polynomial $P_{\alpha,f}$, and write the remaining part as $\bar{f}$; that is,

$$f(\vec{x}) = c_{\alpha,f} + \bar{f}(\vec{x}).$$

A Taylor model of $\bar{f}$ is then $T_{\alpha,\bar{f}} = (P_{\alpha,\bar{f}}, I_{\alpha,\bar{f}})$, where

$$P_{\alpha,\bar{f}}(\vec{x} - \vec{x}_0) = P_{\alpha,f}(\vec{x} - \vec{x}_0) - c_{\alpha,f} \qquad \text{and} \qquad I_{\alpha,\bar{f}} = I_{\alpha,f}.$$

Now we can write

$$\begin{aligned} \exp(f(\vec{x})) &= \exp\left(c_{\alpha,f} + \bar{f}(\vec{x})\right) = \exp(c_{\alpha,f}) \cdot \exp\left(\bar{f}(\vec{x})\right) \\ &= \exp(c_{\alpha,f}) \cdot \left\{1 + \bar{f}(\vec{x}) + \frac{1}{2!}(\bar{f}(\vec{x}))^2 + \cdots + \frac{1}{k!}(\bar{f}(\vec{x}))^k \right. \\ &\qquad \left. + \frac{1}{(k+1)!}(\bar{f}(\vec{x}))^{k+1} \exp\left(\theta \cdot \bar{f}(\vec{x})\right)\right\}, \end{aligned}$$

where $0 < \theta < 1$. Taking $k \geq n$, where n is the order of Taylor model, the part

$$\exp(c_{\alpha,f}) \cdot \left\{1 + \bar{f}(\vec{x}) + \frac{1}{2!}(\bar{f}(\vec{x}))^2 + \cdots + \frac{1}{n!}(\bar{f}(\vec{x}))^n\right\}$$

is a polynomial of $\bar{f}$, of which we can obtain the Taylor model as outlined in the preceding subsection. The remainder part of $\exp(f(\vec{x}))$,

$$\exp(c_{\alpha,f}) \cdot \left\{\frac{1}{(n+1)!}(\bar{f}(\vec{x}))^{n+1} + \cdots + \frac{1}{(k+1)!}(\bar{f}(\vec{x}))^{k+1} \exp\left(\theta \cdot \bar{f}(\vec{x})\right)\right\}, \tag{6.9}$$

will be bounded by an interval. Since $P_{\alpha,\bar{f}}(\vec{x} - \vec{x}_0)$ does not have a constant part, $(P_{\alpha,\bar{f}}(\vec{x} - \vec{x}_0))^m$ starts from mth order. Thus, in the Taylor model computation, the remainder part (6.9) has vanishing polynomial part. The remainder bound interval for the Lagrange remainder term

$$\exp(c_{\alpha,f}) \frac{1}{(k+1)!}(\bar{f}(\vec{x}))^{k+1} \exp\left(\theta \cdot \bar{f}(\vec{x})\right)$$

can be estimated because, for any $\vec{x} \in [\vec{a}, \vec{b}]$, $P_{\alpha,\bar{f}}(\vec{x} - \vec{x}_0) \in B(P_{\alpha,\bar{f}})$, and $0 < \theta < 1$, and so

$$(\bar{f}(\vec{x}))^{k+1} \exp\left(\theta \cdot \bar{f}(\vec{x})\right) \in \left(B(P_{\alpha,\bar{f}}) + I_{\alpha,\bar{f}}\right)^{k+1} \exp\left([0,1] \cdot (B(P_{\alpha,\bar{f}}) + I_{\alpha,\bar{f}})\right). \tag{6.10}$$

Since the exponential function is monotonically increasing, the estimation of the interval bounds of the part $\exp\left([0,1] \cdot (B(P_{\alpha,\bar{f}}) + I_{\alpha,\bar{f}})\right)$ is achieved by inserting the upper and lower bounds of the argument in the exponential.

Similar procedures can be used to obtain Taylor models for square root, multiplicative inverse of square root, sine, cosine, hyperbolic sine, hyperbolic cosine, arcsine, arccosine, and arctanget. Altogether, it is now possible to compute Taylor models along for any function that can be represented in a computer environment along with the mere evaluation of the function by simple operator overloading, in much the same way as the mere computation of derivatives, Taylor polynomials, or interval bounds, along with the mere evaluation of the function.

6.2.3 Derivations and Antiderivations

In the spirit of the idea of embedding the elementary operations of addition, multiplication, and differentiation and their inverses that are defined on the class of C^∞ functions onto the structure of Taylor Models, we now come to the mapping of the derivation operation ∂ as well as its inverse ∂^{-1}. Similar to the case of the DA, and following one of the main thrusts of the history of differential algebra, we will use these for the solution of the initial value problem

$$\frac{d}{dt}\vec{r}(t) = \vec{F}(\vec{r}(t), t),$$

where $\vec{F}$ is continuous and bounded. We are interested in both the case of a specific initial condition $\vec{r}_0$, as well as the case in which the initial condition $\vec{r}_0$ is a variable, in which case our interest is in the flow of the differential equation

$$\vec{r}(t) = \mathcal{M}(\vec{r}_0, t).$$

As in the case of the conventional DA method, in order to prevent loss of order in the differentiation process, the derivation ∂ can be evaluated only in the context of a Lie derivative $L_g = g \cdot \partial$, where $g(\vec{x}_0) = 0$. However, in the case of Taylor models, an additional complication is connected to the

fact that from the Taylor model alone, it is impossible to determine bounds for the derivative, since nothing is known about the rate of change of the function $(f - P_{\alpha,f})$ within the remainder bounds I_f. The situation can be remedied by a further extension of the Taylor model concept to contain not only bounds for the remainder, but also a low-parameter bounding sequence for all the higher derivatives that can occur. In contrast to the derivation ∂, Taylor models of its inverse ∂^{-1} are readily available.

Given an n-th order Taylor model $(P_{n,f}, I_{n,f})$ of a function $f : [\vec{a}, \vec{b}] \subset R^v \to R$ around the reference point $\vec{x}_0$, we can determine a Taylor model for the indefinite integral $\partial_i^{-1} f = \int f \, dx_i$ with respect to the variable x_i. The Taylor polynomial part is obviously just given by $\int_0^{x_i} P_{n-1,f}(\vec{x}) dx_i$. Since the part of the Taylor polynomial $P_{n,f}$ that is of precise order n is $P_{n,f} - P_{n-1,f}$, remainder bounds can be obtained as $(B(P_{n,f} - P_{n-1,f}) + I_{n,f}) \cdot B(x_i)$, where $B(x_i)$ is obtained from the range of definition of x_i as $b_i - a_i$. We thus define the operator ∂_i^{-1}on the space of Taylor models as

$$\begin{aligned} & \partial_i^{-1}(P_{n,f}, I_{n,f}) = (P_{n,\partial^{-1}f}, I_{n,\partial^{-1}f}) \\ = & \left(\int_0^{x_i} P_{n-1,f}(\vec{x}) dx_i \, , \; (B(P_{n,f} - P_{n-1,f}) + I_{n,f}) \cdot B(x_i) \right). \end{aligned} \tag{6.11}$$

With this definition, bounds for a definite integral over variable x_i from x_{il} to x_{iu} both in $[a_i, b_i]$, the domain of validity of the Taylor model of a function, can be obtained as

$$\int_{x_{il}}^{x_{ij}} f(\vec{x}) dx_i \in (P_{n,\partial^{-1}f}(\vec{x}|_{x_i = x_{iu} - x_{i0}}) - P_{n,\partial^{-1}f}(\vec{x}|_{x_i = x_{il} - x_{i0}}), I_{n,\partial^{-1}f}). \tag{6.12}$$

6.3 Examples

RDA have many applications, including global optimization, quadrature, and solution of differential equations. We begin our discussion with the determination of sharp bounds for a simple example function using RDA. The sharpness of the resulting bounds will be compared with the results that can be obtained in other ways. Secondly, we show schematically how the method compares with the interval method to obtain bound enclosures of functions in one and two dimensional cases.

6.3.1 A Simple Function

The function under consideration is

$$f(x) = \frac{1}{x} + x. \tag{6.13}$$

For an actual computation, we set the parameter α of (6.3) to $\alpha = (n, x_0, [a, b])$ $= (3, 2, [1.9, 2.1])$.

As in the case of DA, the evaluation begins with the representation of the identity function, expressed in terms of a Taylor polynomial expanded at the reference point. This identity function i has the form

$$i(x) = x = x_0 + (x - x_0) = 2 + (x - 2).$$

Since this representation is exact, the remainder bound interval is $[0, 0]$. Hence, a Taylor model of the identity function i is

$$T_{\alpha,i} = (x_0 + (x - x_0), [0, 0]) = (2 + (x - 2), [0, 0]).$$

The constant part of i around $x_0 = 2$ is $c_{\alpha,i} = x_0 = 2$, and the nonconstant part of i is $\bar{i}(x) = x - x_0 = x - 2$. The Taylor model of $\bar{i}$ is

$$T_{\alpha,\bar{i}} = ((x - x_0), [0, 0]) = ((x - 2), [0, 0])\,.$$

The computation of the inverse requires the knowledge of bounds of $P_{\alpha,\bar{i}}$, which here is readily obtained: $B(P_{\alpha,\bar{i}}) = B(x - x_0) = [a - x_0, b - x_0] = [-0.1, 0.1]$. We have furthermore $B(P_{\alpha,\bar{i}}) + I_{\alpha,\bar{i}} = [-0.1, 0.1] + [0, 0] = [-0.1, 0.1]$. Using (6.6) and (6.7), we have for the Taylor model of $(\bar{i})^2$

$$T_{\alpha,(\bar{i})^2} = \left((x - 2)^2, [0, 0]\right).$$

The Taylor model of $(\bar{i})^3$ is computed similarly: $T_{\alpha,(\bar{i})^3} = ((x - 2)^3, [0, 0])$. As can be seen, so far all remainder intervals are of zero size. The first nonzero remainder interval comes from the evaluation of the Taylor remainder term, which is

$$\begin{aligned} \frac{(\bar{i}(x))^4}{c_{\alpha,i}^5} \frac{1}{\left(1 + \theta \cdot \bar{i}(x)/c_{\alpha,i}\right)^5} &\in \frac{(B(P_{\alpha,\bar{i}}) + I_{\alpha,\bar{i}})^4}{x_0^5 \cdot \left(1 + [0, 1] \cdot (B(P_{\alpha,\bar{i}}) + I_{\alpha,\bar{i}})/x_0\right)^5} \\ &\subseteq \frac{[0, 0.0001]}{2^5 \cdot ([0.95, 1.05])^5} \subseteq [0, 4.038 \times 10^{-6}]. \end{aligned} \tag{6.14}$$

As expected, this remainder term is "small of order four". The Taylor model of $1/i$ is

$$T_{\alpha,\frac{1}{i}} = \left(\frac{1}{2} - \frac{1}{2^2}(x - 2) + \frac{1}{2^3}(x - 2)^2 - \frac{1}{2^4}(x - 2)^3, [0, 4.038 \times 10^{-6}]\right),$$

and the remainder interval is indeed still very sharp. Using (6.5), we obtain as the final Taylor model of $1/i + i$

$$T_{\alpha,\frac{1}{i}+i} = T_{\alpha,\frac{1}{i}} + T_{\alpha,i} = \left(P_{\alpha,\frac{1}{i}+i}, I_{\alpha,\frac{1}{i}+i}\right) \tag{6.15}$$

$$= \left(\left(2+\frac{1}{2}\right) + \left(1-\frac{1}{2^2}\right)(x-2) + \frac{1}{2^3}(x-2)^2 - \frac{1}{2^4}(x-2)^3, [0, 4.038\times 10^{-6}]\right) \tag{6.16}$$

$$= \left(2.5 + 0.75(x-2) + 0.125(x-2)^2 - 0.0625(x-2)^3, [0, 4.038\times 10^{-6}]\right).$$

Since the polynomial $P_{\alpha,\frac{1}{i}+i}$ is monotonically increasing in the domain $[a,b] = [1.9, 2.1]$, the bound interval of the polynomial is

$$B\left(P_{\alpha,\frac{1}{i}+i}\right) = \left[P_{\alpha,\frac{1}{i}+i}(-0.1), P_{\alpha,\frac{1}{i}+i}(0.1)\right] = [2.42631, 2.57618].$$

The width of the bound interval of the Taylor polynomial is 0.14987, and the width of the remainder bound interval is 4.038×10^{-6} in the third-order Taylor model evaluation; thus the remainder part is just a minor addition. The size of the remainder bounds depends strongly on the order and decreases quickly with order.

The Taylor model computation is assessed by noting the bound interval B of the original function (6.13), which is

$$B\left(\frac{1}{x}+x\right) = \left[\frac{1}{a}+a, \frac{1}{b}+b\right] = [2.42631, 2.57619].$$

It is illuminating to compare the sharpness of the bounding of the function with the sharpness that can be obtained from conventional interval methods. Evaluating the function with just one interval yields

$$\frac{1}{[a,b]} + [a,b] = \frac{1}{[1.9, 2.1]} + [1.9, 2.1] \subseteq [2.37619, 2.62631].$$

The width of the bound interval obtained by interval arithmetic is 0.25012, and so this simple example already shows a noticeable blow-up. By dividing the domain interval into many subintervals, the blow-up can be suppressed substantially. However, to achieve the sharpness of the third-order Taylor model, the domain has to be split into about $24,000$ subintervals.

Practically more important are optimization problems in several variables, and in this case, the situation becomes more dramatic. We wish first to illustrate the computational effort necessary for an accurate calculation of

Table 6.2: The total number of FP operations required to bound a simple function like $f(\vec{x}) = \sum_j (1/x_j + x_j)$.

	One Dimensional	Six Dimensional
Interval	~ 10	~ 10
10^4 divided intervals	$\sim 10^5$	$\sim 10^{25}$
3rd order Taylor model	~ 10	$\sim 10^4$

the result by estimating the required number of floating-point operations. We use a simple example function of six variables such as $f(\vec{x}) = \sum_j (1/x_j + x_j)$ to get a rough idea of the computational expense in the case of functions of many variables. In the one-dimensional case, one interval calculation $1/[a,b] + [a,b]$ requires two additions and two divisions. To compare with the third-order Taylor model computation, we divide the domain into 10^4 subintervals, on which additions and divisions total $\sim 10^5$ floating-point operations. Thus, in the multidimensional case with six independent variables, the number of floating-point operations explodes to $(10^4)^6 \times (\sim 10) = \sim 10^{25}$. Again, sophisticated interval optimization methods will be more favorable than these numbers suggest, but typically there is still a very noticeable growth of complexity.

To estimate the performance of the Taylor model approach, we note that the one-dimensional Taylor model in the third-order computation involves a total of about 35 additions, multiplications, and divisions. As we use more variables, however, the total number of terms in the polynomial grows only modestly. For example, order three in six variables requires only a total of 84 terms. Thus in total, the number of floating-point operations of the third-order Taylor model is $\sim 10^4$. A summary of the number of floating-point operations is given in Table 6.2.

6.3.2 Bound Enclosures of Functions

In this subsection, we use some simple functions in one dimension and two dimensions to show schematically how the RDA method bounds functions in comparison with the interval method. The first function is a one dimensional function

$$f(x) = x(x - 1.1)(x + 2)(x + 2.2)(x + 2.5)(x + 3) \cdot \sin(1.7x + 0.5).$$

Figure 6.2 shows the function and its bound enclosures in the domain $[-0.5, 1.0]$. The interval method is applied to bound the function using smaller domain

intervals divided into 25 subintervals and 50 subintervals. The method of Taylor models computes the polynomial part of the function and the remainder bound interval. The pictures show the bands of the enclosures of the function around its polynomial parts by the remainder bounds using 8th order Taylor models.

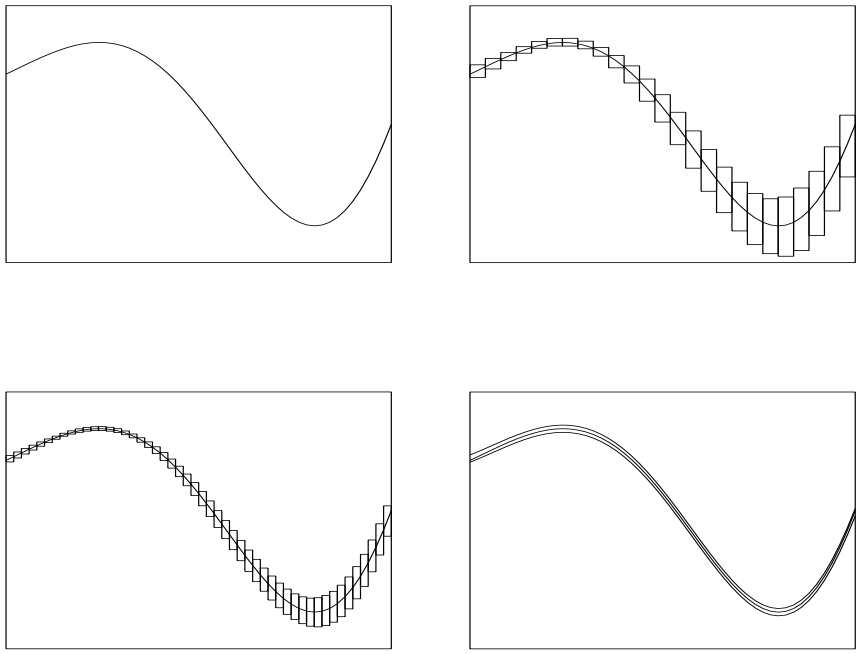


Figure 6.2: One dimensional function and its bound enclosures. From top to bottom right: the function, the bounds by the interval method with the 25 and 50 divided domain intervals, the bound by the 8th order Taylor models.

A similar schematical comparison between the interval method and the method of Taylor models is made for a two dimensional function. We worked on a function

$$f(x, y) = \sin(1.7x + 0.5) \cdot (y + 2) \cdot \sin(1.5y)$$

in the domain $[-1, 1] \times [-1, 1]$. Figure 6.3 shows the function enclosures by the interval method with the smaller domain intervals divided into 10×10, 20×20, 40×40 and 80×80. Figure 6.4 shows the function enclosures around its polynomial parts by the remainder bounds using 7th, 8th, 9th and 10th order Taylor models. Even in this modest case of only two dimensions,

the Taylor model approach requires much less effort to provide a similar level of sharpness; the 1600 subintervals used to include the function are in contrast to only 66 expansion coefficients, plus one remainder bound interval. As dimension is increased, the number of subintervals necessary to provide an accurate modeling increases dramatically, while the number of Taylor coefficients grows much more slowly.

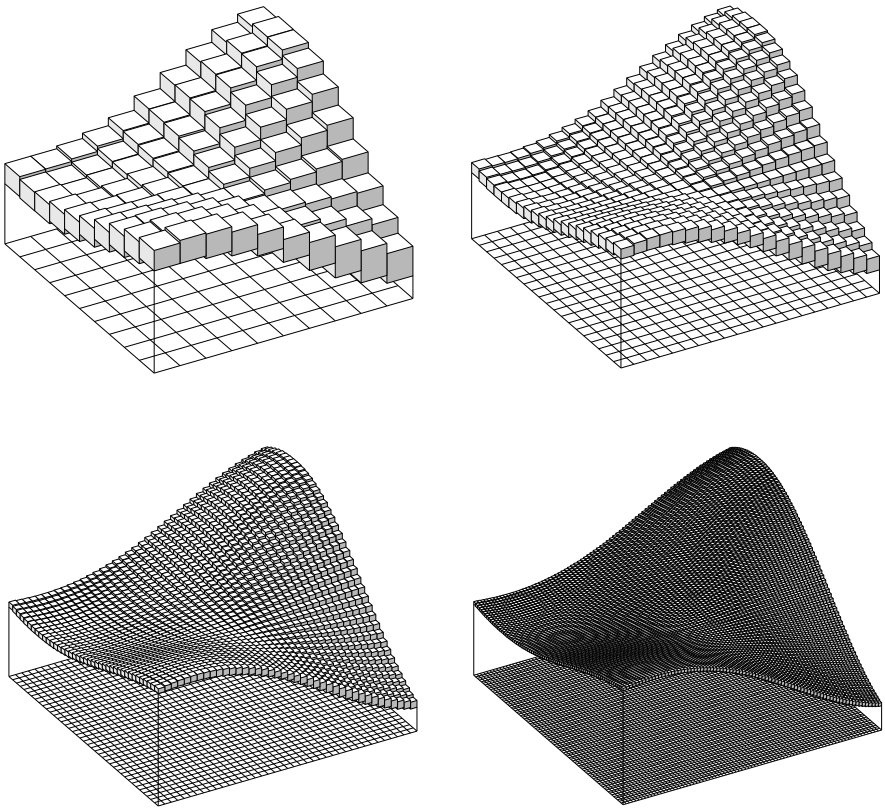


Figure 6.3: Bound enclosures of a two dimensional function by the interval method. From top to bottom right: The domain is divided to 10×10, 20×20, 40×40 and 80×80 subintervals.

6.4 Notes on COSY-GO

COSY-GO [15] is a classical branch-and-bound optimization algorithm which takes advantages of the bounding performances assured by TM methods. Suppose we want to find the global minimum of a sufficiently regular scalar

function f on a given domain $A \subseteq \Re^m$. The algorithm starts with an initial guess for the global optimum, the *cutoff* value, and it proceeds by analyzing at each step a subdomain for possible elimination. At each step the following tasks are performed.

a) A lower bound l of the function is obtained using various bounding schemes in a hierarchical manner. If the lower bound is above the cutoff value, the box is eliminated; if not, the box is bisected. As a first test the polynomial part of the objective function is evaluated in interval arithmetic. When it fails to eliminate the box, a linear dominated bounder (LDB) is used and possible domain reduction is executed. If it also fails to eliminate the box, and if the quadratic part of the polynomial representation of the objective function P is positive definite, a quadratic fast bounding is performed.

b) The cutoff value is updated using various schemes. The linear and quadratic parts of P are utilized to obtain a potential cutoff update. In particular, if the quadratic part of the polynomial is positive definite, the minimizer of the quadratic polynomial is tested. If the quadratic part is not positive definite, the minimizer of the quadratic part in the direction of the negative gradient is tested.

The algorithm continues to split and examine the domain until the minimum dimension allowed is reached. The result of the otpimization is the validated enclosure of the minimum of the problem.

6.5 Validated Solution of Implicit Equations

As seen in previous chapter the evaluation of the overall Δv of a MGA transfer requires the solution of implicit equations. In the first chapter we have shown how the Taylor expansion of the solution manifold can be obtained by means of DA. In order to address the validated optimization of MGA transfers something additional is required. The computation involved in the objective function evaluation must be performed in validated manner indeed, including the solution of implicit equation. In the following the validated solution of a scalar implicit equation is detailed, the extension to the m dimensional case is omitted.

Let $P(x) + I$ be an n-th order Taylor model of the $(n+1)$ times differentiable function f over the domain $D = [-1; 1]$ so that

$$f(x) \in P(x) + I \text{ for all } x \in D \tag{6.17}$$

Let R be an enclosure of the range of $P(x) + I$ over D. Assume P has nonvanishing derivative everywhere in D; without loss of generality P is monotonically increasing, and $P'(x) > d > 0$ on D.

Problem: Find a Taylor model $c(y) + J$ on R such that any solution $x \in D$ of the problem $f(x) = y$ lies in

$$c(y) + J. \tag{6.18}$$

Without loss of generality, we assume $P(0) = 0$. First, determine the polynomial $c(y) = c1y + c2y^2 + \ldots$ by DA inversion of P in the well known way, so that we have

$$P(c(y)) =_n y. \tag{6.19}$$

Indeed, $c(y)$ is the n-th order Taylor polynomial of the inverse of f at the point $0 = P(0)$. It has the property that for any given y it gives an approximate value $x = c(y)$ that satisfies $P(x) \approx y$, and thus also $f(x) \approx y$, depending on how well the inverse is representable by its Taylor expansion over the domain R. However, it is not the true inverse: if we evaluate $P(c(y))$ in n-th order Taylor model arithmetic, we obtain $P(c(y)) \in y + \tilde{J}$, where $\tilde{J}$ is due to the terms of orders exceeding n in $P(c(y))$; and thus scales with at least order $(n + 1)$.

Next we want to find a rigorous remainder J for $c(y)$ so that all solutions of $f(x) = y$ lie in $c(y) + J$. We do this by studying the consequences of small corrections x to $c(y)$. We observe that, according to the mean value theorem,

$$\begin{aligned} f(c(y) + \Delta x) - y &\in P(c(y) + \Delta x) - y + I \\ &= P(c(y)) + \Delta x \cdot P'(\xi) - y + I \\ &\subset y + \tilde{J} + \Delta x \cdot P'(\xi) - y + I \\ &= \Delta x \cdot P'(\xi) + I + \tilde{J} \end{aligned} \tag{6.20}$$

for some suitable ξ that lies between $c(y)$ and $c(y) + \Delta x$. However, since on $[-1; 1]$, P' is bounded below by d, we observe that in $[-1; 1]$, but outside of the interval

$$J = -\frac{I + \tilde{J}}{d} \tag{6.21}$$

the set $\Delta x \cdot P'(\xi) + I + \tilde{J}$ will never contain zero, and thus no solution of $f(c(y) + \Delta x) - y$ exist there. So we obtain:

Theorem. *Any solution $x \in D = [-1; 1]$ of $f(x) = y$ lies in the Taylor model $c(y) + J$.*

Let us make some remarks.

1) No information on the derivative f' is needed, which is helpful in practice.

2) Indeed, if f is not monotonic, it is conceivable that $c(y) + J$ contains multiple solutions of $f(x) = y$

3) For any $y \in [P(-1) + u(I); P(+1) + l(I)]$, at least one solution exists due to the intermediate value theorem.

4) For any $y \notin [P(-1) + l(I); P(+1) + u(I)]$, no solution exists

As an important consequence, we obtain the following:

Corollary. *If f is invertible over D, $c(y) + J$ is a Taylor model of order $(n + 1)$ of its right inverse.*

6.6 Test Cases

Two test cases are addressed, the first one is a simple planet-to-planet transfer whereas the second exploits a gravity assist. As it will be shown the addition of a single gravity assist significantly complicates the optimization problem.

6.6.1 EM

The first test case is an Earth-Mars transfer. This problem is representative of any planet-to-planet transfer as it involves only two ephemerides evaluations and one Lambert's problem solution. The transfer is formulated in departure epoch and time of flight in order to deal with a rectangular search domain, thus easing the branch-and-bound algorithm implemented in COSY-GO. As no GA is considered, we deal with an unconstrained optimization problem, and this is particularly beneficial as COSY-GO does not explicitly handle constraints. The search space and the enclosure of the optimal solution is shown in Table 6.3.

The enclosure of the optimal Δv is $[5.6673264, 5.6673272]$ km/s which includes the optimal solution of 5.6673270 km/s found with GASP and indicated in Figure 6.5 with the red star. The price to pay for obtaining the validated optimum solution is the computational time: the optimization process takes 4954.39 s on a Pentium IV 3.06 GHz laptop platform. The main reason of this result is the presence of discontinuity lines of the objective function within the search domain. Whenever a discontinuity is found in a box, the validated evaluation of the objective function is an algebraic failure. In these cases the optimizer is forced to split the domain until the minimum

Table 6.3: Time bounds and enclosure of the optimal solution for EM transfer

	T_E MJD 2000	t_{EM} days
L_b	1000	100
U_b	6000	600
EM	[3573.176, 3573.212]	[324.034, 324.088]

box size is reached. This situation is clear by looking at Figure 6.6 which shows that most of the iterations do not bring any improvement in the cutoff value.

At the end of the optimization the remaining domain includes both the box containing the optimal solution and the boxes which contains discontinuities. For this reason the result can not be considered the validated enclosure of the solution, but rather the global optimum from an engineering point of view. Indeed it is improbable to find a better enclosure of the solution among boxes with algebraic failures as a very small minimum box size is set. Furthermore it's worth noting that the objective function discontinuities are not physical but the result of the problem formulation (Lambert's problem solution). In order to achieve a mathematical proven optimal solution a different formulation of the problem, which avoids the solution of the Lambert's problem, should be adopted.

6.6.2 EVM

The second test case considers a transfer to Mars and includes a Venus GA. The simple addition of a GA makes the problem tougher as the constraint on the minimum pericenter altitude must be considered. As COSY-GO is an unconstrained optimizer there are two possible ways to deal with constraints: penalty functions and space pruning. The first one consists in augmenting the objective function with a term representative of the constraint violation. The second one assigns an arbitrary high value to the objective function whenever a box entirely violates the constraint. The result presented here uses the second technique. The main advantage is that no weighting factor must be chosen; on the other hand there's the possibility that the optimal solution found violates the constraint. This problem is unlikely to happen if the minimum box dimension allowed is very small.

Table 6.4 shows the problem definition as well as the enclosure of the optimal solution. It's worth noting that, unless the results are presented

Table 6.4: Time bounds and enclosure of the optimal solution for EVM transfer

	T_E MJD 2000	t_{EV} days	t_{VM} days
L_b	5000	140	200
U_b	6000	240	400
EVM	[5611.475, 5611.512]	[157.592, 157.623]	[255.564, 255.620]

in term of departure epoch and time-of-flights, the optimization problem is formulated in absolute times in order to reduce the functional dependencies. The enclosure of the optimal minimum is $\Delta v \in [8.5220251, 8.5231393]$ km/s which includes the optimal value of 8.5226 obtained with GASP.

The increased problem dimension has a dramatic impact on the computational time; the optimization process takes about three weeks on the same platform used for the previous test indeed. The boxes that includes a discontinuity must be split until all the three sizes are under the minimum value allowed, thus causing a blow-up in the computational time even when a small search domain is analyzed.

6.7 Final Remarks

The TM-based validated optimizer is a potential tool for finding the global optimum of any MGA transfer. On the other hand, several issues must be addressed to make its use viable for complex MGA transfers. A method for managing the discontinuities is necessary; to this aim an ad hoc problem formulation could be a viable option. Furthermore a more efficient and safe method to handle the constraints is mandatory. A possible solution is represented by the explicit computation of the constraint manifold. Finally the reduction of the computational time could be achieved by running COSY-GO in parallel on several machines. In any case the large computational time enlightened in the previous section is the price to pay for gaining a validated enclosure of the optimal solution.

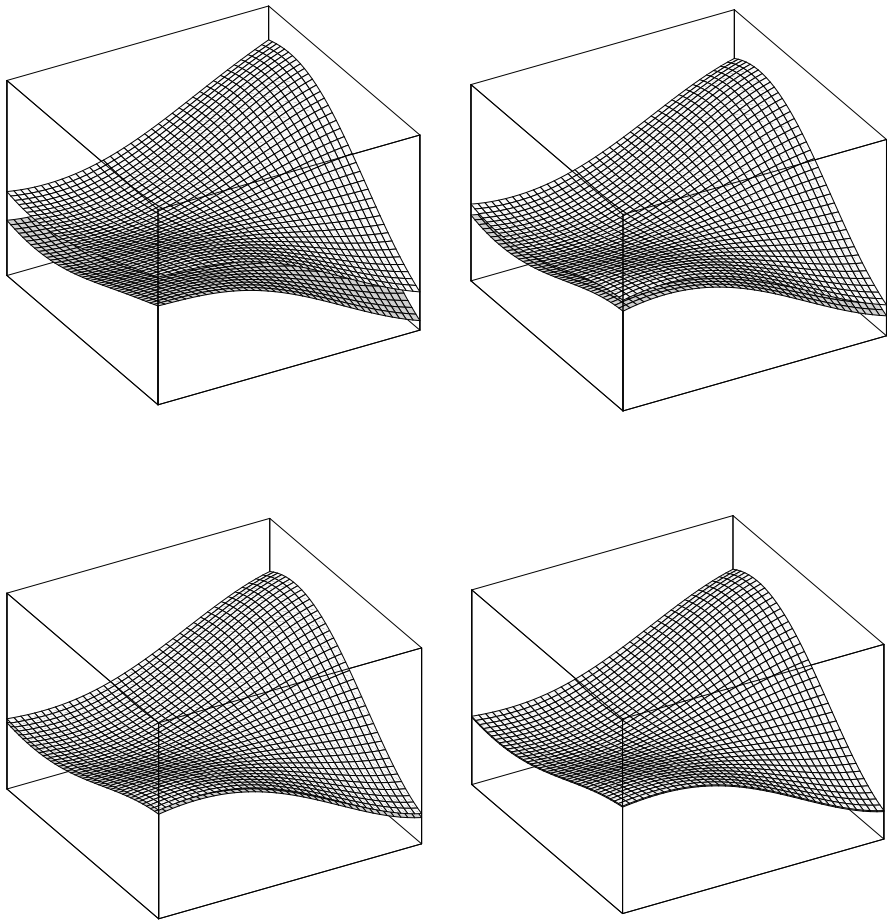


Figure 6.4: Bound enclosures of a two dimensional function by the RDA method. From top to bottom right: Computations in 7th, 8th, 9th and 10th order Taylor models.

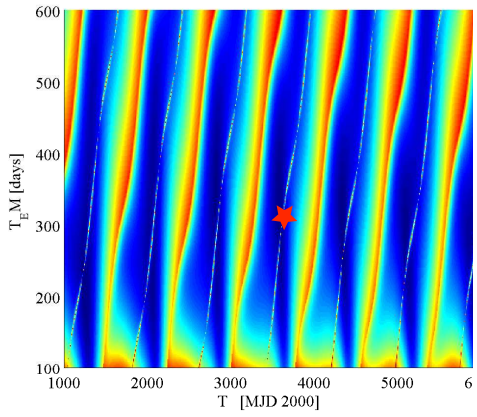


Figure 6.5: Objective function values and optimal solution found.

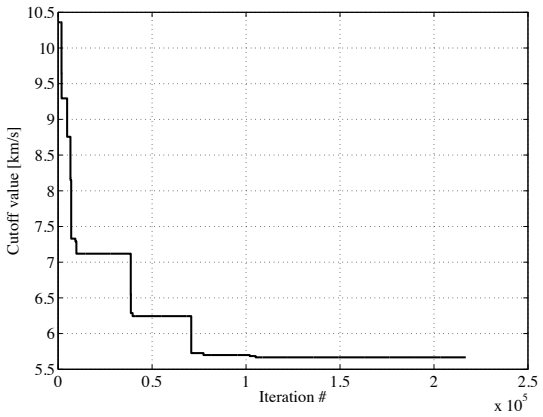


Figure 6.6: Δv cutoff history.

Chapter 7

Conclusions and Final Remarks

This chapter is devoted to trace the work described throughout the report, and to identify key points for possible future developments. The final remarks pertaining the pruning and optimization of MGA transfers involving DSM are first presented, whereas special comments are dedicated to the validated global optimization of MGA transfers.

A version of the classical pruning algorithm GASP, based on the use of differential algebraic techniques, has been implemented and described in chapter 3. Significant work was aimed at addressing the discontinuity and dependency problems. The major solution to the discontinuity problem was the use of a planar planetary model for the Solar System. Based on the observation that the discontinuity corresponding to the transition from the short–way to the long–way solution of the Lambert's problem disappears, the planar model is used to perform the pruning process. Furthermore, considering the low inclinations of all planetary orbits, the overall ΔV associated to the planar model has been conjectured to be usually lower than the corresponding ΔV in the actual three–dimensional model. Consequently, no information and solutions are lost by performing the pruning process in the planar model, and then optimizing the transfer in the actual three–dimensional model. The dependency problem was solved by replacing the classical relative times formulation for MGA transfer optimization problems, with the effective absolute times formulation, based on the use of the epochs at each planet as design variables. As demonstrated by the test phase, the resulting algorithm GASP–DA turns out to effectively prune and optimize MGA transfers without DSM.

Then, GASP–DA has been extended to include DSM. Chapter 3 showed that a dependency problem still occur, which is evidently similar to the same problem in classical MGA transfers. This led to the necessity of modeling

the whole transfer as a sequence of Lambert's arcs, which corresponds to adopt a formulation, which has been again referred to as absolute variables formulation. Several successful test cases have been carried out to assess the performances of the resulting algorithm.

The use of the resulting GASP–DSM–DA algorithm is not free of significant drawbacks. First of all, a problem related to the allocated memory can be identified. In particular, as reported in chapter 2, a DA number is a vector of $(n+v)!/(n!\,v!)$ coefficients, where n is the order of the corresponding expansion and v is the number of variables. Consequently, the vector size quickly increases with n and v (see [14] for further details). It is worth highlighting that many coefficients are usually equal to zero in practical computations. However, the tool COSY INFINITY, that implements differential algebra and has been used within this work, does not allow to dynamically allocate memory. Consequently, a great amount of memory must be allocated anyway at the beginning of the computation, which often increases prohibitively in practical computations.

A further drawback is associated again to the dependency problem. Specifically for MGA transfers involving DSM, three variables must be added to the set of design variables for each additional DSM. Given the structure of the MGA–DSM problem, and the periodicities associated to most of the entailed variables, the multimodality of the optimization problem increases with the number of DSM introduced, and a growing number of local minima tends to characterize the optimization landscape. In such a complicated framework, Taylor expansions tend to lose their accuracy, and smaller box sizes are required to reach reasonable pruning performances. Alternative approximation techniques might worth further investigation in this scenery, as the use of Fourier series to replace Taylor series when periodicities occur.

The pruning process of MGA transfers, including or not DSM, is anyway aimed at easing the subsequent necessary optimization phase. Thanks to the imposed constraints, large parts of the search space are intended to be pruned away. In this way, the effectiveness of the optimization algorithms at identifying the actual global optimum of the original problem is finally enhanced. Given the steep increase of computational burden, associated to the pruning of MGA transfers involving DSM, alternative strategies have been sough to solve the ultimate optimization problem, which have been described in chapter 5. Specifically, considering the high efficiency of GASP and GASP–DA when pruning MGA transfers without DSM, an alternative strategy has been developed. Given the sequence of planets involved in a MGA transfer:

1. GASP–DA is used to identify a set of good solutions for the MGA

transfer, without considering the introduction of DSM;

2. based on the concept of Pareto–optimality, solutions are selected from the previous set;

3. DSM are introduced based on suitable heuristics;

4. the corresponding solutions are used as first guesses for a subsequent optimization process.

Despite the algorithm does not rely directly on search space pruning of MGA transfers involving DSM, it proves some important advantages when compared to GASP–DSM–DA. First of all, the optimization process is carried out in less computational time, and larger search spaces can be processed with affordable computational burden. Moreover, the optimal solutions are comparable in many transfers of practical interest, as shown by Table 7.1.

Table 7.1: Comparison of the three developed methods in terms of optimal Δv.

	Δv [m/s]		
	GASP–DA	GASP–DA DSM	DSM Altern.
EVM	8.523	8.167	8.081
EMJ	13.42	12.481	12.394
EMMJ	12.864	10.843	10.968
EVEEJ	10.09	8.670	8.687
EVEME	12.443	10.931	10.874
EVVEJS	8.619	8.299	8.588

Clearly, all the implemented pruning and optimization algorithms are not free of further possible improvements. First of all, pertaining GASP–DA, a key-point for significant enhancements is including the automatic selection of the optimal planets sequence within the optimization process. This would lead to a mixed continuous–combinatorial optimization problem, whose solution can be extremely difficult, especially if a high number of planets for GA maneuvers is considered. Moreover, concerning both GASP–DSM–DA and the alternative strategy developed in chapter 5, the possibility of inserting more than one DSM per arc should be addressed. This advance serves

the purpose using DSM to approximate low–thrust transfers at the beginning of the design phase, as widely suggested in the space–trajectory design community. Within the framework of low–thrust arcs modeling during the preliminary phases of space–trajectory design, a further important step will be the introduction of exponential sinusoid in GASP–DA. Based on the work of Izzo [22], the Lagrange equation for the time of flight is substituted by an alternative implicit equation, which needs to be solved within differential algebra, using techniques similar to the one presented in chapter 2.

Separate comments are instead dedicated to the validated global optimization of MGA transfers investigated in chapter 6. It is worth gaining more valuable insights on the main drawbacks encountered, and proposing possible promising solutions. As highlighted in chapter 6, the main difficulty is identified to be related to the management of constraints in COSY–GO. Referring to Figure 7.1, suppose the function obj must be minimized. A box $[X]$ is being processed and part of the box turns out to be unfeasible under the imposed constraints. The current version of COSY–GO does not handle constraints automatically, especially if boxes lying on the constraint manifold are being analyzed. Consequently, the box in Figure 7.1 would be considered as entirely feasible if no expedients are used. Suppose now a Taylor model evaluation of the objective function over $[X]$ is performed, which leads to the range reported in Figure 7.2. The algorithm now tries to update the cutoff value for the optimization process using the estimation from above of the minimum of the objective function over $[X]$. The exact estimation from above of the minimum within the box is reported as $\overline{obj}^*_{min}$ in Figure 7.2. However, if the whole box $[X]$ was considered as entirely feasible, the estimation actually used by COSY–GO would be $\overline{obj}_{min}$, which clearly underestimate the actual feasible minimum $\bar{obj}^*_{min}$. The main consequence is that, if $\overline{obj}_{min}$ is used to update the cutoff value for the optimization process, actual solutions might be pruned away. Intensive investigations must be devoted to overcome this problem. Besides the possibility of adding constraints to the objective function as penalty terms, which might turn out to modify the landscape of the original objective function within the feasible region of the search space, an alternative solution can be outlined. Ad–hoc techniques exist to compute validated enclosures of the constraint manifolds, which are based on the use of Taylor models. In this way, given a box to be processed, the validated enclosure is used to rigorously identify the feasible regions of the box. The objective function is the evaluated over these regions only, and the resulting estimated minimum can be correctly used to update the cutoff value for the optimization process.

A further drawback of the use of COSY–GO for MGA transfers is the steep increase of the computational time with the number of design vari-

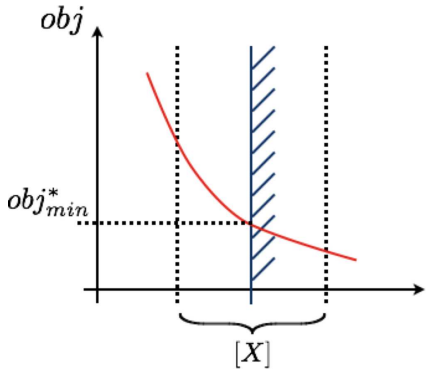


Figure 7.1: Constraint management in COSY–GO.

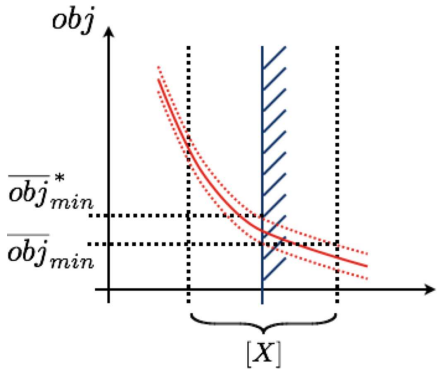


Figure 7.2: Underestimation of the actual minimum.

ables, which does not exploit the problem structure in the default setting, as performed in GASP. Suitable adjustments on the current version of COSY can be considered as viable solutions for future applications. However, it is worth observing that the versatility of COSY–GO has an important benefit: COSY–GO algorithm is fully parallelizable. This means that the same optimization process can be distributed on an high number of processors to significantly reduce the computational time.

Bibliography

[1] Ascher, U., and Petzold, L., *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, 1998.

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

[3] Berz, M., "The new method of TPSA algebra for the description of beam dynamics to high orders", Technical Report AT-6:ATN-86-16, Los Alamos National Laboratory, 1986.

[4] Berz, M., "The method of power series tracking for the mathematical description of beam dynamics", Nuclear Instruments and Methods A258, 431, 1987.

[5] Berz, M., "Differential algebraic description of beam dynamics to very high orders", Technical Report SSC-152, SSC Central Design Group, Berkeley, CA, 1988.

[6] Berz, M., Fawley, B., and Hahn, K., "High order calculation of the multipole content of three dimensional electrostatic geometries", Nuclear Instruments and Methods A307, 1, 1991.

[7] Berz, M., "High-Order Computation and Normal Form Analysis of Repetitive Systems," *Physics of Particle Accelerators*, Volume AIP 249, pp. 456, 1991.

[8] Berz, M., "COSY INFINITY Version 6", In M. Berz, S. Martin, and K. Ziegler (Eds.), Proc. Nonlinear Effects in Accelerators, page 125, IOP Publishing, 1992.

[9] Berz, M., "Automatic differentiation as nonarchimedean analysis", In Computer Arithmetic and Enclosure Methods, Amsterdam, page 439, Elsevier Science Publishers, 1992.

[10] Berz, M., "New features in COSY INFINITY", In Third Computational Accelerator Physics Conference, page 267, AIP Conference Proceedings 297, 1993.

[11] Berz, M., Hoffstätter, G., Wan, W., Shamseddine, K., and Makino, K., "COSY INFINITY and ist application to nonlinear dynamics", In Computational Differentiation: Techniques, Applications, and Tools, M. Berz, C. Bischof, G Corliss, and A. Griewank (Eds.), SIAM, 1996.

[12] Berz, M., "COSY INFINITY version 8 reference manual", Technical Report MSUCL-1088, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 1997.

[13] Berz, M., "Differential Algebraic Techniques", Entry in "Handbook of Accelerator Physics and Engineering", M. Tigner and A. Chao (Eds.), World Scientific, 1999.

[14] Berz, M., *Modern Map Methods in Particle Beam Physics*, Academic Press, 1999.

[15] Berz, M., Makino, K., and Kim, Y., "Long-term stability of the tevatron by verified global optimisation", Nuclear Instruments and Methods, A558, pp. 1-10, 2005.

[16] Di Lizia, P., and Radice, G., "Advanced Global Optimisation for Mission Analysis and Design", Final Report, Ariadna id: 03/4101, Contract No. 18139/04/NL/MV, 2004. (Available at: http://www.esa.int/gsp/ACT/doc/ACTRPT-03-4101-ARIADNA-GlobalOptimisationGlasgow.pdf)

[17] Di Lizia, P., Radice, G., and Vasile, M., "On the solution of interplanetary trajectory design problems by global optimisation methods", International Workshop on Global Optimisation, Almería, Spain, 18-22 September, 2005.

[18] Gobetz, F., "Optimal transfer between hyperbolic asymptotes", AIAA Journal, 1, 1963.

[19] Griepentrog, E., and Marz, R., *Differential-Algebraic Equations and Their Numerical Treatment*, Teubner, 1986.

[20] He, J., Watson, L., Ramakrishnan, N., Shaffer, C. A., Verstak, A., Jiang, J., Bae, K., and Tranter, W., "Dynamic data structures for a direct search algorithm", Technical Report TR-01-09, Computer Science, Virginia Tech, 2001.

[21] Izzo, D., Becerra, V., Myatt, D., Nasuto S., and Bishop, J., "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories", Journal of Global Optimisation, 38, pp. 283–296, 2006.

[22] Izzo D., "Lamberts problem for exponential sinusoids", Journal of Guidance, Dynamics, and Control, 29, pp. 1242–1245, 2006.

[23] Izzo, D., "Advances in Global Optimisation for Space Trajectory Design", 25th International Symposium on Space Technology and Science, 2006.

[24] Jones, D., "A taxonomy of global optimisation methods based on response surfaces", Journal of Global Optimisation, 21, pp. 345–383, 2001.

[25] Kennedy, J., and Eberhart, R., "Particle swarm optimisation", Proc. IEEE Int. Conf. on Neural Networks, pp. 1942–1948, 1995.

[26] Makino, K., "Rigorous Analysis of Nonlinear Motion in Particle Accelerators", Ph.D. Thesis, Michigan State University, East Lansing, Michigan, USA, 1998.

[27] Michelotti, L., "MXYZTPLK: A practical, user friendly c++ implementation of differential algebra", Technical Report, Fermilab, 1990.

[28] Myatt, D., Becera, V., Nasuto, S., and Bishop, J., "Advanced Global Optimisation for Mission Analysis and Design", Final Report, Ariadna id: 03/4101, Contract No. 18138/04/NL/MV, 2004. (Available: http://www.esa.int/gsp/ACT/doc/ACT-RPT-03-4101-ARIADNA-GlobalOptimisationReading.pdf)

[29] Moore, R.E., *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.

[30] Van Zeijts, J., and Neri, F., "The arbitrary order design code TLIE 1.0", In M. Berz, S. Martin, and K. Ziegler (Eds.), Proceedings Workshop on Nonlinear Effects in Accelerators, IOP Publishing, 1993.

[31] Vasile, M., and De Pascale, P. "Preliminary Design of Multiple Gravity-Assist Trajectories", Journal of Spacecraft and Rockets, Vol. 43, N. 4, pp. 794–805, 2006.

[32] Vinko, T., Izzo, D., and Bombardelli, C. "Benchmarking different global optimisation techniques for preliminary space trajectory design" 58th International Astronautical Congress, Hyderabad, India, 2007.

[33] Yan, Y., "ZLIB and related programs for beam dynamics studies", In Third Computational Accelerator Physics Conference, pp. 279, AIP Conference Proceedings, 297, 1993.