Efficient Usage of Self Validated Integrators for Space Applications

Final Report

Authors: E.M. Alessi, A. Farrés, À. Jorba, C. Simó, A. Vieiro
Affiliation: University of Barcelona

ESA Researcher(s): T. Vinkó

Date:

Contacts:

Àngel Jorba
Tel: +34 934035734
Fax: +34 934021601
e-mail: angel@maia.ub.es
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: 07/5202
Study Duration: 6 months
Contract Number: 20783/07/NL/CB

Efficient Usage of Self Validated Integrators for Space Applications

Final Report

February 2008

Senior researchers: Àngel Jorba, Carles Simó

Junior researchers: Elisa M. Alessi, Ariadna Farrés, Arturo Vieiro

Departament de Matemàtica Aplicada i Anàlisi
Universitat de Barcelona
Gran Via 585, 08007 Barcelona, Spain.

ESA Research Fellow / Technical Officer: Tamás Vinkó

Ariadna ID: 07/5202
Study duration: 6 months
Contract Number: 20783/07/NL/CB

Abstract

The present report addresses the possibilities offered by validated integrators to deal with some space problems. First, we survey previous results on validated methods for ODEs and we discuss their sharpness and efficiency, keeping in mind their suitability for space-related computations.

We have considered two concrete situations. The first one is the propagation of the orbit of a NEO asteroid, starting from data affected by observational error for a moderate time span. We have focused on the concrete case of 99942 Apophis, which is having close approaches to the Earth on the years 2029 and 2037. The goal here is to check if the use of validated methods can be useful to elucidate possible collisions of these objects with the Earth. The second problem considered is the transfer of a probe with low thrust propulsion from parking orbit around the Earth to a much higher orbit, including a capture by the Moon. The goal is to see how the uncertainties in the initial condition and the thrust affect the propagation of the orbit.

Contents

Appendixes

Chapter 1

Introduction

Many real life problems are well described by models consisting of *Ordinary Differential Equations* (ODE),

$$\frac{dx}{dt} = f(t, x, \lambda),$$

where x belongs to some phase or states space $\mathcal{E} \subset \mathbb{R}^n$ and λ, which accounts for the parameters of the model, belongs to some *parameter space* $\mathcal{P} \subset \mathbb{R}^p$. Here we will focus on equations such that the vector field f is given in closed form by means of analytic functions.

For most of the ODE the solutions to the Initial Value Problem (IVP) $x(t_0) = x_0$ are not known in closed form or cannot be obtained by solving a simple equation, as happens, e.g., in the case of the two-body problem which only requires the solution of Kepler's equation.

For concreteness we denote the flow by φ, that is, given t_0, x_0, λ, the solution at a final time t_f is denoted as

$$\varphi(t_f; t_0, x_0, \lambda).$$

This leads to the need of numerical integration of the IVP for ODE. Unfortunately all methods are affected by the presence of integration errors at each step and by the propagation of errors by the proper dynamics.

In many critical applications, like the motion of a NEO or space missions using a close fly-by to a massive body, the effect of the errors can be critical. Hence, it is natural to ask for methods providing solutions which are as correct as possible and, better, to guarantee that the solution at time t_f is inside a given subset of the phase space. Furthermore this subset should be "small", that is, the method should not overestimate its size.

On the other hand both initial data t_0, x_0 and the values of the parameters λ, are not exactly known. Consider, for instance, the case of a NEO a few days after it has been discovered (or even at present time). Due to bad conditioning of initial observations the elements of its orbit have a large uncertainty. Hence, one should be able to give rigorous and sharp estimates of the set

$$\varphi(t_f, t_0, X_0, \Lambda) = \left\{\varphi(t_f, t_0, x_0, \lambda), \text{ for all } x_0 \in X_0, \lambda \in \Lambda\right\},$$

where X_0 and Λ are the sets which contain all possible initial values of x_0, λ, respectively.

For conceptual simplicity, and without loss of generality, it is better to consider state and parameter variables together, that is, we can introduce

$$z = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad \frac{dz}{dt} = \begin{pmatrix} f(t, z) \\ 0 \end{pmatrix},$$

where $z \in \mathcal{EP} = \mathcal{E} \times \mathcal{P}$, the state–parameter space. Also initial data can be considered in a set Z_0 like $X_0 \times \Lambda$ or a subset of it.

Different methods have been proposed to obtain rigorous estimates of $Z_f = \varphi(t_f, t_0, Z_0)$ based on the use of interval analysis and generically known as "Self Validated Integrators" (see Chapter 4).

The three main points which should be taken carefully into account, when considering all these methods and applications are:

a) Rigorous estimates should be provided.

b) The estimates should be sharp, that is, not producing a too large image domain.

c) They should be efficient from a CPU time point of view.

1.1 Sources of errors

The numerical integration of ODE is a key task in many applications. In this work we have been concerned with conservative systems, e.g., given by some Hamiltonian function, or close to conservative systems. In contrast to some dissipative systems having simple global attractors (like steady states or periodic solutions) the effect of the errors introduced by the numerical method can invalidate the numerically obtained solution.

One should distinguish clearly between regular and chaotic orbits. The difference in behaviour can be measured by means of the Lyapunov exponents (see, e.g. [Sim01]). They measure the exponential rate of separation of nearby orbits. We can consider as regular the orbits for which the maximal Lyapunov exponent is zero and then nearby orbits separate in a potential way (typically in a linear way) with respect to time. Otherwise, orbits for which the maximal Lyapunov exponent is positive can be considered as chaotic or with strong sensitivity to initial conditions.

One should note, however, that Lyapunov exponents are defined as a limit of the exponential rate of separation when time tends to infinity. In practice time will be finite and the orbit only explores a reduced part of the phase space. It is better to refer to some kind of local Lyapunov exponent where local refers to the fact that the orbits we can be interested in move only on a reduced subset of the available phase space.

Numerical integration of ODE is affected by different kinds of errors:

1. Errors in the initial data, x_0,

2. Errors in the model, for instance in the parameters λ describing the model,

3. Truncation errors due to the fact that the numerical method, even if computations were done with infinite precision, is just an approximation, depending on the order of the method,

4. Round off errors due to the fact that computations are done with a finite number of digits,

5. Propagation of the errors produced in the different steps due to the proper characteristics of the dynamics. This is specially critical in the case of chaotic orbits.

It is clear that the effect of items 3. and 4. can be decreased by high order methods and, if required, by doing computations with more digits (e.g., with quadruple precision or higher). But one should face the problems due to items 1. and 2. Concerning item 5. it is possible, in some cases, to decrease its effect by a suitable reformulation of the problem (e.g., using some regularising variables like KS variables for close approaches in perturbed two-body problems, see [SS71]).

In any case, a reasonable method should behave in a nice way if we consider the evolution of the theoretically preserved quantities, such as energy, momentum, etc.

Furthermore, it is also clear that to cope with item 5 a validated method requires information, at least, on the first variational equations. This allows to bound the derivatives of the final state with respect to the initial one (and also with respect to the parameters, of course). These equations are used for the computation of Lyapunov exponents. They allow to detect if the orbit behaves in a regular or chaotic way, which is relevant, even at a local scale, to decide about the more convenient set of variables to be used.

The strong relation between propagation of errors and dynamical behaviour of system at hand must necessarily be taken into account.

1.2 The Taylor method

Consider the initial value problem

$$\left\{ \begin{array}{rcl} x'(t) &=& f(t, x(t)), \\ x(a) &=& x_0, \end{array} \right. \tag{1.1}$$

where f is assumed to be analytic on its domain of definition, and that $x(t)$ is assumed to be defined for $t \in [a, b]$. We are interested in approximating the function $x(t)$ on $[a, b]$. The idea of the Taylor method is very simple: given the initial condition $x(t_0) = x_0$ ($t_0 = a$), the value $x(t_0 + h_0)$ is approximated from the Taylor series of the solution $x(t)$ at $t = t_0$,

$$\begin{array}{rcl} x_0 &=& x(t=0), \\ x_{m+1} &=& x_m + x'(t_m)h_m + \dfrac{x''(t_m)}{2!}h_m^2 + \cdots + \dfrac{x^{(p)}(t_m)}{p!}h_m^p, \quad m = 0, \ldots, M-1, \end{array} \tag{1.2}$$

where $t_{m+1} = t_m + h_m$, $h_m > 0$ and $t_M = b$.

For a practical implementation one needs an effective method to compute the values of the derivatives $x^{(j)}(t_m)$. A first procedure to obtain them is to differentiate the first equation in (1.1) w.r.t. t, at the point $t = t_m$. Hence,

$$x'(t_m) = f(t_m, x(t_m)), \quad x''(t_m) = f_t(t_m, x(t_m)) + f_x(t_m, x(t_m))x'(t_m),$$

and so on. Therefore, the first step to apply this method is, for a given f, to compute these derivatives up to a suitable order. Then, for each step of the integration (see (1.2)), we have to evaluate these expressions to obtain the coefficients of the power series of $x(t)$ at $t = t_m$. Usually, these expressions will be very cumbersome, so it will take a significant amount of time to evaluate them numerically. This, jointly with the initial effort to compute the derivatives of f, is the main drawback of this approach for the Taylor method.

This difficulty can be overcome by means of the so-called *automatic differentiation* (see [BKSF59], [Wen64], [Moo66], [Ral81], [GC91], [BCCG92], [BBCG96], [Gri00]). This is a procedure that allows for a fast evaluation of the derivatives of a given function, up to arbitrarily high orders. As far as we know, these ideas were first used in Celestial Mechanics problems ([Ste56], [Ste57]; see also [Bro71]).

We note that the algorithm to compute these derivatives by automatic differentiation has to be coded separately for different systems. This coding can be either done by a human (see, for instance, [Bro71] for an example with the N-body problem) or by another program (see [BKSF59, Gib60, CC94, JZ05] for general-purpose computer programs). An alternative procedure to apply the Taylor method can be found in [SV87] and [IS90]. We can also find some public domain software to generate numerical integrators of ODEs using Taylor methods:

- ATOMFT, http://www.eng.mu.edu/corlissg/FtpStuff/Atom3_11/. ATOMFT is written in Fortran 77 and it reads Fortran-like statements of the system of ODEs and writes a Fortran 77 program that is run to solve numerically the system using Taylor series.
- taylor. It can be obtained from http://www.maia.ub.es/~angel/taylor/. It reads a file with a system of ODEs and it outputs a time-stepper for it (in C/C++), with automatic selection of order and step size. Several extended precision arithmetics are supported.

There is also a public domain package for automatic differentiation, ADOL-C, included as an option in many Linux distributions (home page: http://www.math.tu-dresden.de/~adol-c/). It facilitates the evaluation of first and higher derivatives of vector functions that are defined by computer programs written in C or C++.

There are several papers that focus on computer implementations of the Taylor method in different contexts; see, for instance, [BWZ70], [CC82], [CC94] and [Hoe01]. A good survey is [NJC99] (see also [Cor95]).

1.2.1 Automatic differentiation

As it has been mentioned before, automatic differentiation is a recursive procedure to compute the value of the derivatives of certain functions at a given point (relevant references are [Moo66, Ral81, Gri00]). The functions considered are those that can be obtained by sum, product, quotient, and composition of elementary functions (elementary functions include polynomials, trigonometric functions, real powers, exponentials and logarithms). We note that the vector fields used in Celestial Mechanics and Astrodynamics belong to this category. Other functions can be considered as elementary if they are defined as the solution of some differential equation whose coefficients are previously known to be elementary functions. A notorious case of a non-elementary function is $\Gamma(x)$. A celebrated theorem of Hölder states that Γ does not satisfy any algebraic differential equation whose coefficients are rational functions.

Assume that a is a real function of a real variable t.

Definition 1.2.1 *The normalised j-th derivative of a at the point t is*

$$a^{[j]}(t) = \frac{1}{j!} a^{(j)}(t).$$

Assume now that $a(t) = F(b(t), c(t))$ and that we know the values $b^{[j]}(t)$ and $c^{[j]}(t)$, $j = 0, \dots, n$, for a given t. The next proposition gives the n-th derivative of a at t for some functions F.

Proposition 1.2.1 *If the functions b and c are of class C^n, and $\alpha \in \mathbb{R} \setminus \{0\}$, we have*

1. *If $a(t) = b(t) \pm c(t)$, then $a^{[n]}(t) = b^{[n]}(t) \pm c^{[n]}(t)$.*
2. *If $a(t) = b(t)c(t)$, then $a^{[n]}(t) = \displaystyle\sum_{j=0}^{n} b^{[n-j]}(t) c^{[j]}(t)$.*
3. *If $a(t) = \dfrac{b(t)}{c(t)}$, then $a^{[n]}(t) = \dfrac{1}{c^{[0]}(t)} \left[b^{[n]}(t) - \displaystyle\sum_{j=1}^{n} c^{[j]}(t) a^{[n-j]}(t) \right]$.*
4. *If $a(t) = b(t)^\alpha$, then $a^{[n]}(t) = \dfrac{1}{n b^{[0]}(t)} \displaystyle\sum_{j=0}^{n-1} \left((n-j)\alpha - j\right) b^{[n-j]}(t) a^{[j]}(t)$.*
5. *If $a(t) = e^{b(t)}$, then $a^{[n]}(t) = \dfrac{1}{n} \displaystyle\sum_{j=0}^{n-1} (n-j)\, a^{[j]}(t) b^{[n-j]}(t)$.*

6. *If* $a(t) = \ln b(t)$*, then* $a^{[n]}(t) = \dfrac{1}{b^{[0]}(t)}\left[b^{[n]}(t) - \dfrac{1}{n}\displaystyle\sum_{j=1}^{n-1}(n-j)b^{[j]}(t)a^{[n-j]}(t)\right].$

7. *If* $a(t) = \cos c(t)$ *and* $b(t) = \sin c(t)$*, then*

$$a^{[n]}(t) = -\frac{1}{n}\sum_{j=1}^{n} jb^{[n-j]}(t)c^{[j]}(t), \quad b^{[n]}(t) = \frac{1}{n}\sum_{j=1}^{n} ja^{[n-j]}(t)c^{[j]}(t).$$

It is possible to derive similar formulas for other functions, like inverse trigonometric functions.

We note that the number of arithmetic operations to evaluate the normalised derivatives of a function up to order n is $O(n^2)$. We will come back to this point later on.

As an example, we can apply them to the Van der Pol equation,

$$\left.\begin{array}{rcl} x' &=& y, \\ y' &=& (1-x^2)y - x. \end{array}\right\} \tag{1.3}$$

To this end we decompose the right-hand side of these equations in a sequence of simple operations:

$$\left.\begin{array}{rcl} u_1 &=& x, \\ u_2 &=& y, \\ u_3 &=& u_1u_1, \\ u_4 &=& 1-u_3, \\ u_5 &=& u_4u_2, \\ u_6 &=& u_5-u_1, \\ x' &=& u_2, \\ y' &=& u_6. \end{array}\right\} \tag{1.4}$$

Then, we can apply the formulas given in Proposition 1.2.1 (items 1 and 2) to each of the equations in (1.4) to derive recursive formulas for $u_j^{[n]}$, $j = 1, \ldots, 6$,

$$\begin{array}{rcl} u_1^{[n]}(t) &=& x^{[n]}(t), \\ u_2^{[n]}(t) &=& y^{[n]}(t), \\ u_3^{[n]}(t) &=& \displaystyle\sum_{i=0}^{n} u_1^{[n-i]}(t)u_1^{[i]}(t), \\ u_4^{[n]}(t) &=& -u_3^{[n]}(t), n > 0 \\ u_5^{[n]}(t) &=& \displaystyle\sum_{i=0}^{n} u_4^{[n-i]}(t)u_2^{[i]}(t), \\ u_6^{[n]}(t) &=& u_5^{[n]}(t) - u_1^{[n]}(t), \\ x^{[n+1]}(t) &=& \dfrac{1}{n+1}u_2^{[n]}(t), \\ y^{[n+1]}(t) &=& \dfrac{1}{n+1}u_6^{[n]}(t). \end{array}$$

The factor $\frac{1}{n+1}$ in the last two formulas is because we are computing normalised derivatives. Then, we can apply recursively these formulas for $n = 0, 1, \ldots,$ up to a suitable degree p, to obtain the jet of normalised derivatives for the solution at a given point of the ODE. Note that is not necessary to select the value of p in advance.

1.2.2 Estimation of order and step size

There are several possibilities to estimate an order and step size for the Taylor method. When Taylor is used in a non-validated way, these estimates come from the asymptotic behaviour of the error. The following result can be found in [Sim01].

Proposition 1.2.2 *Assume that the function $z \mapsto x(t_m + z)$ is analytic on a disk of radius ρ_m. Let A_m be a positive constant such that*

$$|x_m^{[j]}| \leq \frac{A_m}{\rho_m^j}, \qquad \forall j \in \mathbb{N}, \tag{1.5}$$

and assume that the dominant part in the computational cost is proportional to the square of the order up to which the Taylor series is computed. Then, if the required accuracy ε tends to 0, the values of h_m (local step) and p_m (local order) that give the required accuracy and minimise the global number of operations tend to

$$h_m = \frac{\rho_m}{e^2} \qquad \textit{and} \qquad p_m = -\frac{1}{2}\ln\left(\frac{\varepsilon}{A_m}\right) - 1. \tag{1.6}$$

Note that the optimal step size does not depend on the level of accuracy. The optimal order is, in fact, the order that guarantees the required precision once the step size has been selected.

It is important to note that the values (1.6) are optimal only when the bound (1.5) cannot be improved. If the value A_m can be reduced –or if the function $x(t)$ is entire– the previous values are not optimal in the sense that a larger h_m and/or a smaller p_m could still deliver the required accuracy.

1.2.3 Validated integration

Validated methods are numerical methods based on an “exact” arithmetic, combined with a rigorous estimate of the truncation errors. Hence, the results (usually values with bounds on the error) are rigorous. For an “exact” arithmetic we mean a *validated floating point arithmetic*, which is usually an interval arithmetic, that guarantees that the result is between the given bounds (see Section 1.3).

One of the advantages of validated methods is that they produce rigorous results, that can be used to derive computer-assisted proofs of theorems (see, for instance, [Lan82, EKW84]). If the computation requires the numerical integration of a non-stiff ODE, then Taylor method is probably the best choice to have a rigorous estimation of the error.

In these cases, it is quite common to use a high precision arithmetic to check the conditions needed to derive the computer assisted proof. For instance, [KS07] shows the linear stability of the Figure Eight solution of the 3-body problem, by using an extended precision arithmetic (100 digits) and computing this orbit and its first variationals up to a very high accuracy. This allows for a very precise validated integration which, in turn, allows to complete the proof.

The situation considered in this work is of a different nature. We are focusing on space applications (propagating the motion of an asteroid or a spacecraft), and this means that there is an error in the data that cannot be reduced by extending the accuracy of the arithmetic. Because of the relatively large size of these errors, it will be enough to use interval arithmetics based on the double precision of the computer.

1.3 Interval arithmetic

Interval arithmetic was first introduced by E. Moore [Moo66] in 1966. In this book the foundations of interval arithmetic where laid. In the past years, it has been developed being now a very important tool for the computation of rigorous bounds for many problems in numerical analysis. See, for instance, [Zgl07] for a description of different applications.

We define the set of intervals on the real line as:

$$\mathbb{IR} = \{[a] = [a_l, a_u] \quad | \quad a_l, a_u \in \mathbb{R}, \quad a_l \leq a_u\}.$$

An interval number $[a]$ is the set of real numbers x that satisfy $a_l \leq x \leq a_u$. If $a_l = a_u$ we say that we have a point interval and if $a_l = -a_u$ we say that we have a symmetric interval. Two intervals $[a], [b]$ are equal iff $a_l = b_l$ and $a_u = b_u$.

We can define the classical inclusions as follows, we will say that $[a] \subseteq [b]$ iff $a_l \leq b_l \leq b_u \leq a_u$. And we can also define the partial ordering as $[a] < [b]$ iff $a_u < b_l$.

We can also define the following quantities:

- $w([a]) = a_u - a_l$ is the *width* of the interval $[a]$.
- $m([a]) = (a_l + a_u)/2$ is the *mid point* of the interval $[a]$.
- $|[a]| = \max(|a_l|, |a_u|)$ is the *magnitude* of the interval $[a]$.

1.3.1 Real interval arithmetic

The interval arithmetic is an extension of the real arithmetic. We will first assume that the end points of the intervals can be computed in an exact way. Further on we will discuss which modifications must be done so that the round-off errors are taken into account.

Let $[a]$ and $[b] \in \mathbb{IR}$ and $\circ \in \{+, -, *, /\}$ be one of the arithmetic operations. We define the arithmetic operation on intervals as:

$$[a] \circ [b] = \{x \circ y \quad | \quad x \in [a], y \in [b]\}, \tag{1.7}$$

where, for $\circ = /$ the operation is not defined if $0 \in [b]$. It is immediate to see that:

- $[a] + [b] = [a_l + b_l, a_u + b_u]$,
- $[a] - [b] = [a_l - b_u, a_u - b_l]$,
- $[a] * [b] = [\min(a_l b_l, a_l b_u, a_u b_l, a_u b_u), \max(a_l b_l, a_l b_u, a_u b_l, a_u b_u)]$,
- $[a]/[b] = [a_l, a_u] * [1/b_u, 1/b_l]$, if $b_l > 0$.

Properties:

1. The interval addition and multiplication are both associative and commutative, i.e. if $[a], [b]$ and $[c] \in \mathbb{IR}$, then:

$$\begin{aligned} [a] + ([b] + [c]) &= ([a] + [b]) + [c], \\ [a] + [b] &= [b] + [a], \\ [a] * ([b] * [c]) &= ([a] * [b]) * [c], \\ [a] * [b] &= [b] * [a]. \end{aligned}$$

2. The real numbers 0 and 1 are identities for the interval addition and multiplication respectively, i.e. if $[a]$ is an interval then:

$$0 + [a] = [a] + 0 = [a],$$
$$1 * [a] = [a] * 1 = [a].$$

3. The inverse elements for addition and for multiplication only exist in the degenerate case, i.e. for point intervals.

4. The arithmetic operations are monotonic inclusions, i.e. if $[a_1] \subset [a_2]$ and $[b_1] \subset [b_2]$ then:

$$[a_1] \circ [b_1] \subset [a_2] \circ [b_2], \text{ where } 0 \notin [b_2] \text{ if } \circ = /.$$

5. The distributive law does not always hold for interval arithmetic, but we do have the sub distributive law, i.e. if $[a], [b]$ and $[c] \in \mathbb{IR}$, then

$$[a] * ([b] + [c]) \subseteq [a] * [b] + [a] * [c].$$

   It can be seen that the distributive law holds if $[b] * [c] \geq 0$, if $[a]$ is a point interval or if both $[b]$ and $[c]$ are symmetric.

We will refer to an interval vector and interval matrix as a vector or matrix where all their components are intervals. And we will define the set of n-dimensional real interval vectors as $\mathbb{IR}^n$ and the set of $n \times m$-dimensional real interval matrices as $\mathbb{IR}^{n\times m}$. The interval arithmetic operations involving interval vectors and interval matrices are defined as in the scalar case but substituting the scalars by intervals and using the definitions seen above. Further properties of the interval arithmetic on interval vectors and matrices and the extension of the interval arithmetic to other functions can be seen in [Moo66, Moo79, AH83].

1.3.2 Rounded interval arithmetic

The interest in interval arithmetic has aroused, specially due to the limitations of the representation of floating point numbers on a computer. Every time that a real number x is stored on a computer or that computations with such numbers are made, round-off errors occur. Rounded interval arithmetic provides a tool to bound the roundoff error in an automatic way.

Now, instead of representing the real number x by a machine number, it will be represented by an interval $[x] = [x_\blacktriangledown, x_\blacktriangle]$, where $x_\blacktriangledown$ is the negative rounding of x and $x_\blacktriangle$ is the positive rounding of x. Then we can use the real interval arithmetic described before but rounding positive on the right end points and rounding negative on the left end points, i.e. if $[a] = [a_l, a_u]$ and $[b] = [b_l, b_u]$ are intervals, with the endpoints previously rounded positive and negative as needed, and $\circ = \{+, -, *, /\}$, then:

- $[a] + [b] = [(a_l + b_l)_\blacktriangledown, (a_u + b_u)_\blacktriangle]$,
- $[a] - [b] = [(a_l - b_u)_\blacktriangledown, (a_u - b_l)_\blacktriangle]$,
- $[a] * [b] = [\min(a_l b_l, a_l b_u, a_u b_l, a_u b_u)_\blacktriangledown, \max(a_l b_l, a_l b_u, a_u b_l, a_u b_u)_\blacktriangle]$,
- $[a]/[b] = [a_l, a_u] * [1/b_u, 1/b_l]$, if $b_l > 0$.

On our validated computations we will use a rounded interval arithmetic instead of the real interval arithmetic. In this way, after all the computations, we will have an interval that contains the exact result.

1.3.3 The dependency problem and the wrapping effect

As it was observed by E. Moore [Moo66], interval arithmetic is sometimes affected by overestimation due to the dependency problem, which happens as interval arithmetic cannot detect the different occurrences on the same variable.

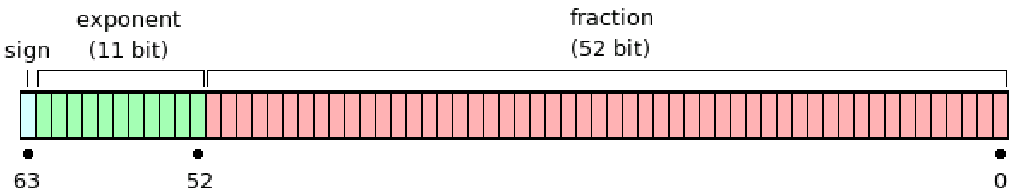

Figure 1.1: The fields in a double variable, according to the standard IEEE 754

For instance, $x - x = 0$ for all $x \in [1, 2]$, but using interval arithmetic we have that $[1, 2] - [1, 2] = [-1, 1]$. It is true that the true solution $[0, 0]$ is contained in $[-1, 1]$, but we have a big overestimation. Another example would be $x^2 - x$ for $x \in [0, 1]$. It can be seen that for all $x \in [0, 1]$, $(x^2 - x) \in [-1/4, 0]$. Note that using interval arithmetic we have $[0, 1] * [0, 1] - [0, 1] = [-1, 1]$. Notice that $x^2 - x = x(x - 1)$ and if we evaluate this second expression: $[0, 1] * ([0, 1] - 1) = [-1, 0]$, we get a better estimation of the real interval. So here we can see two clear examples of the dependency on the order of the operations that occur in interval arithmetic. This is a situation that must be taken into account in order to get accurate estimations of the real result. A correct estimate of $f([a, b])$ for continuous functions would require to find extrema of f on $[a, b]$. In general, this cannot be achieved quickly, specially in the case of several variables. Anyway, subdivision algorithms can improve the result. In the $x \mapsto x^2 - x$ example, subdivision in 10 equal parts gives the result $[-0.35, 0.1]$.

Another source of error in interval arithmetics is the wrapping effect. Let us consider the function: $f(x, y) = \frac{\sqrt{2}}{2}(x + y, y - x)$. The image of the square: $[0, \sqrt{2}] \times [0, \sqrt{2}]$ is the rotated square of corners $(0, 0), (1, -1), (2, 0)$ and $(1, 1)$, but interval arithmetic gives $[0, 2] \times [-1, 1]$, a square that doubles the size.

These and other examples have been studied by E. Moore [Moo66] and other authors, for instance [NJ01]. In this project, we will describe techniques that can be developed in order to deal with these two effects in an accurate way, and to get good approximations of the real result.

1.3.4 Software implementations for interval arithmetic

There are several possibilities to construct an interval arithmetic. Usually, an interval arithmetic is defined as a structure with two members, the upper and the lower bounds of the interval. These members are floating point types. Depending on the considered application, they can be double precision numbers, or to have extended precision.

In this work, due to the fairy large amount of error in the initial data, and to the relatively moderate number of floating point operations in the experiments done so far, we have chosen to use an arithmetic based on the standard double precision of the computer: using extended arithmetic makes the programs slower without any significant gain in accuracy.

There are several libraries in the public domain for interval arithmetic. For a comprehensive list, see `http://www.cs.utep.edu/interval-comp/intsoft.html`. As it has been mentioned above, we will only focus on arithmetics based on the standard double precision. In the next sections we discuss several possibilities to implement such arithmetic.

Accessing the internal representation

There is a widely accepted standard to store floating point numbers, the *IEEE Standard for Binary Floating-Point Arithmetic* (IEEE 754). In short, floating point numbers are stored using one bit for the sign, 11 bits for the exponent, and 52 bits for the mantissa (for a discussion see, for instance, [Gol91]). This has been represented schematically in Figure 1.1

The idea is, after each arithmetic operation, to access the bit representation of the result and to add (or subtract) 1 to the less significant bit to produce an enclosure for the true result. This technique has been used in the libraries `FI_LIB` (ANSI C) and `FILIB++` (C++), freely accessible from `http://www.math.uni-wuppertal.de/~xsc/software/filib.html`. More details can be found in [LTWVG+06].

Using the rounding modes of the processor

A second possibility is provided by the rounding modes specified by the standard IEEE 754: round to nearest (the default), round toward 0, round toward $+\infty$, and round toward $-\infty$. For instance, if we are adding two intervals it is enough to add the lower bounds with rounding toward $-\infty$ and the upper bounds with rounding toward $+\infty$.

The GCC compiler on Intel platforms provides library functions to access and modify the rounding modes of the processor. We have implemented a first version of an interval arithmetic based on the use of these rounding modes. The main drawback is that every time the rounding mode is altered, the floating point pipeline is restarted which means a considerable loss of performance of the floating point unit.

Correcting the output of the operations

We have considered a third possibility: assuming that the result of each arithmetic operation is correct up to rounding error, we can modify the result of the operation up and down to obtain an enclosure for the true result (as it is done in the `FI_LIB` and `FILIB++` libraries). The difference is that, instead of accessing the bits of the mantissa of the floating point variable (which is costly in time), we multiply by a correcting factor. For instance, if $a = b + c > 0$, we can construct the upper and lower bounds by doing $\underline{a} = a(1-\delta)$ and $\overline{a} = a(1+\varepsilon)$, for suitable positive constants ε and δ. It is clear that ε and δ have to be as small as possible to minimise the growing of the size of the intervals, but if they are too small the rounding of the computer arithmetic makes that $\underline{a} = a = \overline{a}$, and the result will be wrong.

We have selected the values $\varepsilon = 2^{-52}$ (the so-called machine epsilon) and $\delta = 2^{-53}$ (half of the machine epsilon). The reason for using this δ comes from the way the subtraction (and the subsequent rounding) is done.

In the implementation (in C++) we have added a flag to optionally ask for verification of the result after each operation. This verification consists in checking the condition $\underline{a} < a < \overline{a}$, which means that we have obtained an enclosure for the result. A particular point that has to be considered here is the effect of the extra digits that the Intel CPU has in its floating point registers (80 bits each). Think of the following situation:

a) the code generated by the compiler is keeping the variables in these extra precision registers so that the relation $\underline{a} < a < \overline{a}$ is verified inside those registers;

b) once these numbers are stored in memory (only 64 bits), the roundoff makes that $\underline{a} = a$ or $a = \overline{a}$.

This checking is done by accessing the bit representation of the floating point number (8 bytes), using a `union` type variable to overlap the number with two integer variables (4 bytes each). In this way, the checking is done in a very fast way: the check slows down the arithmetic by a 30%. We have done this check on a set of 10^{10} randomly chosen double precision numbers and the condition has been satisfied in all the cases. A second test has been to use the flag `-ffloat-store` of the GCC compiler when compiling the code that checks for the condition $\underline{a} < a < \overline{a}$: this flag forces that, after each arithmetic operation, the result is stored in memory (64 bits). In this way, we have ruled out the previous effect when checking the previous inequalities. This second test slows down a lot the arithmetic.

Comparisons

We have done several comparison bewtween these arithmetics. We only explain one of the tests, which is enough to give an idea of how they compare. This test is the scalar product of two arrays of `interval` type, of 1000 components each. To have a measurable time, this product is done 10^6 times. As a reference, this test takes about 4.4 seconds using the double precision arithmetic of the computer (a Linux workstation with a 3.2GHz Intel Xeon CPU). The same test using our library takes 24.8 seconds, and using `FI_LIB` it takes 108.4 seconds. For this test we have compiled the `FI_LIB` library with the `-O3` option, although the makefile that comes with the source of the library does not use any optimization flag.

Other options

There are other options to construct a fast interval arithmetic. For instance, instead of storing the upper and lower bountd for the interval, we can store the central point and the radius. Then, the operations are done using the central point of the intervals (using the standard floating point arithmetic) and the resulting radius is modified to account for the roundoff of the operation. We have not explored this possibility in this work, but it is an interesting option that we plan to consider in the future.

Chapter 2

Applications

The aim of this work is to develop validated methods that can be applied to space related problems. We have chosen two particular examples where we will apply these methods. The first one is the propagation of the orbit of a Near Earth Object (NEO) and the second one is a low thrust transfer of a probe from a neighbourhood of the Earth to a neighbourhood of the Moon. In both problems we have an uncertainty in the initial condition and we want to know the effect of these uncertainties after a long time. They also require a long time integration that must be done in an accurate way. In this section, we will briefly describe these two applications.

2.1 The two-body problem

A first approximation of the motion of an asteroid orbiting around the Sun or of a probe that orbits around the Earth is the well-known two-body problem (details can be found in almost any textbook on Classical or Celestial Mechanics, e.g. [Mou14, Pol76, Dan88]). We will use this problem as a toy model to test the validated techniques that have been developed. This is done due to the simplicity of this model and because we have a complete knowledge of the solutions of the problem.

The two-body problem describes the motion of two point masses X_1 and $X_2 \in \mathbb{R}^3$ of masses m_1 and m_2, respectively, that are evolving under their mutual gravitational attraction. This problem can be reformulated by taking $r = X_1 - X_2$, namely,

$$\ddot{r} = -\mu \frac{r}{\|r\|^3}, \tag{2.1}$$

where $\mu = G(m_1 + m_2)$ is the mass parameter and G is the gravitational constant.

The above system is super integrable, that is, it has more first integrals than degrees of freedom. In particular, the first integral of the energy can be defined as

$$E = \frac{v^2}{2} - \frac{\mu}{r}, \tag{2.2}$$

where v represents the velocity of the second body with respect to the first one.

In our tests, we will consider that one of the particles is massless. This is the most natural assumption when one considers the motion of an asteroid around the Sun or of a low-thrust probe in the Earth-Moon system. We notice that this hypothesis reduces the problem to a central vector field in the asteroid case.

2.1.1 Local behaviour around an elliptic orbit

We are interested in the numerical propagation of an uncertainty in position and, as first approximation, Kepler's third law gives how such error is propagated.

We recall that this reads

$$n^2a^3 = \mu, \tag{2.3}$$

where n and a are, respectively, the mean motion and the semi-major axis associated with a given orbit. Roughly speaking, if we consider two orbits characterised by different values of a, say $a_1 > a_2$, then Eq. (2.3) yields $T_1 > T_2$, where $T_i = 2\pi/n_i$ $(i = 1, 2)$ is the corresponding period. This means that on the first orbit a particle will move slower than on the second one.

We can apply the above argument to predict the behaviour of a set of initial conditions which are displaced one from the other in position. This displacement results in an uncertainty on the semi–major axis, say Δa, which makes the mean motion n to vary in a quantity Δn. More precisely, expanding Eq. (2.3) up to first order, we have

$$2na^3\Delta n + 3a^2n^2\Delta a = 0. \tag{2.4}$$

We can always rescale the variables of the problem in such a way that $\mu = 1$, $n_0 = 1$ (i.e. $T_0 = 2\pi$) and $a_0 = 1$, where the subindex 0 refers to a nominal initial condition. In this way, any initial condition associated with $a_i = a_0 + \Delta a = 1 + \Delta a$ will be characterised by

$$n_i = n_0 + \Delta n = 1 - \frac{3}{2}\Delta a.$$

After m revolutions of the nominal orbit, the others will be delayed by an angle $\Delta\psi$ given by

$$\Delta\psi = -\frac{3}{2}mT_0\Delta a = -3m\pi\Delta a \approx -10m\Delta a.$$

As a consequence, if we integrate numerically a random set of initial conditions, we will see the box stretching out along the orbit, that is, at a given time different points displaced one ahead the other.

Let us consider two vectors, one tangent to the orbit and the other orthogonal to it at the initial time. After m revolutions, the angle α between them will have changed according to

$$\tan(\alpha) = \frac{\Delta a}{\Delta\psi} \approx \frac{1}{10m}.$$

For m large enough, we have

$$\alpha \approx \frac{1}{10m}.$$

Moreover, the stretching of the box behaves linearly, in a first approximation, with respect to the time t. This can be proved noting that in action-angle variables, I and φ, the angle φ evolves linearly with time, depending only on the constant action. Since this change of variables is a diffeomorphism, the same behaviour is observed approximately in Cartesian coordinates if we approximate locally the diffeomorphism by its differential map.

2.2 Solar System models

As we want to describe the motion of an object in space, we first need to have a good model of the motion of the Solar System. It is well known that the motion of the planets can be approximated by the N-body problem.

Body/Planet	$G \cdot m$		
Mercury	0.4912547451450812	×	10^{-10}
Venus	0.7243452486162703	×	10^{-9}
Earth	8.8876923901135099	×	10^{-10}
Mars	0.9549535105779258	×	10^{-10}
Jupiter	0.2825345909524226	×	10^{-6}
Saturn	0.8459715185680659	×	10^{-7}
Uranus	0.1292024916781969	×	10^{-7}
Neptune	0.1524358900784276	×	10^{-7}
Pluto	0.2188699765425970	×	10^{-11}
Moon	1.0931895659898909	×	10^{-11}
Sun	0.2959122082855911	×	10^{-3}

Table 2.1: Values of $G \cdot m$ in AU^3/day^2 for the bodies considered.

2.2.1 The N-body problem

We suppose that we have N bodies in space that are evolving under the effect of their mutual gravitational attraction. This problem has been highly studied (see [Mou14, Dan88, MH92]). In our particular integrations we have taken into account the following bodies: Sun, Mercury, Venus, Earth, Moon, Mars, Jupiter, Saturn, Neptune, Uranus and Pluto. As we will describe objects that have close approaches with the Earth's orbit, the gravitational effect due to the Moon must be considered.

The equations of motion are:

$$\ddot{X}_i = \sum_{j=1, j\neq i}^{11} \frac{Gm_j(X_j - X_i)}{r_{ij}^3}, \qquad \text{for } i = 1, \ldots, 11 \tag{2.5}$$

where $X_1, \ldots, X_{11} \in \mathbb{R}^3$ are the position of the 11 bodies, $m_1, \ldots, m_{11}$ are their masses, r_{ij} is the distances between the bodies X_i and X_j $(i, j = 1, \ldots, 11)$ and $G = 6.67259 \times 10^{-11}\text{m}^3/(\text{s}^2\text{kg})$ is the gravitational constant. In this notation, each body is related to a number as follows: 1=Mercury, 2=Venus, 3=Earth, 4=Mars, 5=Jupiter System, 6=Saturn System, 7=Uranus System, 8=Neptune System, 9=Pluto, 10=Moon and 11=Sun.

We have taken as units of mass, distance and time: 1 kg, 1 AU and 1 day, respectively. In Table 2.1, we can see the values of $G \cdot m$ in these units. This and other astronomical constants can be found in [Sei92] or [JPL].

We must mention that, to obtain a full understanding of the dynamics of a body in the Solar System, other effects should be taken into account. Among them, the relativistic correction, the forces due to other natural satellites and asteroids, the J_2 (and higher order harmonics of the potential) effect of the Earth and other bodies. However, these terms can be considered negligible for our purpose. In the next section we will see some tests that have been made to verify the accuracy of our model.

2.2.2 The JPL model

The JPL Solar System Ephemerides are computer files that store information to derive the positions of Sun, Earth, Moon and the planets in three-dimensional, Cartesian coordinates.

In this report we have used the ephemerides DE405 of Caltech's Jet Propulsion Laboratory (JPL). They have been obtained from a least-square fitting of previously existing ephemerides to the available observation data, followed by a numerical integration of a suitable set of equations that describe the motion of the Solar system.

A detailed description about how these ephemerides are obtained can be found in [SW]. In short, we will only mention that the equations of motion used for the creation of DE405 include contributions from: (a) point mass interactions among the Moon, planets, and Sun; (b) general relativity (isotropic, parametrised post-Newtonian); (c) Newtonian perturbations of selected asteroids; (d) action upon the figure of the Earth from the Moon and Sun; (e) action upon the figure of the Moon from the Earth and Sun; (f) physical libration of the Moon, modelled as a solid body with tidal and rotational distortion, including both elastic and dissipational effects, (g) the effect upon the Moon's motion caused by tides raised upon the Earth by the Moon and Sun, and (h) the perturbations of 300 asteroids upon the motions of Mars, the Earth, and the Moon.

The numerical integrations were carried out using a variable step-size, variable order Adams method. The result of the integration is stored in form of interpolatory data (Chebyshev polynomials, each block of them covers an interval of 32 days). The DE405 ephemerides is defined from Dec 9, 1599 to Feb 1, 2200 (there are other ephemerides covering longer time spans, with a slightly lower accuracy).

The internal reference system is the so-called J2000 coordinates. This is a Cartesian frame, with origin at the Solar system barycentre, the XY plane is parallel to the mean Earth Equatorial plane, the Z axis is orthogonal to this plane, the X axis points to the vernal point and the Y axis is selected to have a positive oriented reference system. All these references are taken at 2000.0 (Jan 1st, 2000, at 12:00 UT).

For numerical integrations, we access the file DE405 to obtain the positions of the bodies of the Solar system. We have coded a few routines to interface our programs with the JPL programs for the ephemerides, in particular we have added the option of changing from the equatorial coordinates of the ephemeris to ecliptic coordinates, which are more natural to deal with asteroids.

To obtain initial conditions for Apophis, we have used [GBO$^+$08] and the JPL Horizons system ([JPL]). The Horizons system provides a very simple and convenient web interface method to access for the initial conditions of an asteroid (or any body in the Solar system, in a variety of coordinates).

2.3 The orbit of a Near Earth Object (NEO)

A Near Earth Object is an asteroid, a comet or a meteoroid whose orbit can get significantly close to the Earth's one, ranging from zero (collision) to a few Earth–Moon distances. In particular, the perihelion distance can assume values less than 1.3 AU.

To model the motion of a NEO we will take a restricted $(N+1)$-body problem. As before, we will suppose that we have N bodies, the 9 planets, Moon and Sun (N = 11), that are evolving under their mutual gravitational attraction and that we have a massless particle, the NEO, which is affected by the gravitational attraction of the N bodies but that has no gravitational effect on them. The equations of motion are:

$$\begin{aligned} \ddot{X}_i &= \sum_{j=1, j\neq i}^{11} \frac{Gm_j(X_j - X_i)}{r_{ji}^3}, \qquad \text{for } i = 1, \dots, 11 \\ \ddot{X}_a &= \sum_{j=1}^{11} \frac{Gm_j(X_j - X_a)}{r_{ja}^3}, \end{aligned} \tag{2.6}$$

where again $X_1, \dots, X_{11} \in \mathbb{R}^3$ are the position of the 11 bodies, $m_1, \dots, m_{11}$ are their masses, $X_a \in \mathbb{R}^3$ is the position of the asteroid, r_{ij} denotes the distance between the main body i and the main body j $(i, j = 1, \dots, 11)$ and r_{ia} the one between the major bodies i $(i = 1, \dots, 11)$ and the asteroid. Just mention that we have taken the same units of mass, distance and time as before.

In our simulations we have taken the asteroid (99942) Apophis. This asteroid was discovered in 2004 when it presented a high probability of a close approach to the Earth and a possible collision with it. The trajectory of this asteroid has recently been studied by various research groups (see [GBO+08] and [MCS+05]). Information about the physical data, close passages of Apophis with the Earth, among others, can also be found in [NEO]. It seems that Apophis will exhibit a close approach with the Earth on Friday 13 April 2029 and another in 2037. For this reason, it is necessary to have a rigorous methodology to clarify if a collision might take place.

2.4 The transfer of a probe using low-thrust

The second example considered is the transfer of a spacecraft from the Earth to the Moon using a low-thrust propulsion system. This problem can be approached in several ways. In our formulation, we will consider that the probe is accelerated (or decelerated) by a constant low thrust in a given direction.

Also in this case, the model adopted is a restricted $(N+1)$-body problem where now the massless particle will be the low-thrust probe and the N bodies will be the 11 bodies already mentioned. The probe will be affected by the gravitational effect of the N bodies and by an extra force due to the thruster. Under these assumptions, the equations are:

$$\begin{aligned} \ddot{X}_i &= \sum_{j=1, j\neq i}^{11} \frac{G m_j (X_j - X_i)}{r_{ji}^3}, \qquad \text{for } i = 1, \dots, 11 \\ \ddot{X}_{sat} &= \sum_{j=1}^{11} \frac{G m_j (X_j - X_{sat})}{r_{jsat}^3} + F_T \frac{V_{sat} - V_c}{\|V_{sat} - V_c\|}, \end{aligned} \tag{2.7}$$

where $X_1, \dots, X_{11} \in \mathbb{R}^3$ are the position of the 11 bodies, $m_1, \dots, m_{11}$ are the masses of these bodies and r_{ij} $(i, j = 1, \dots, 11)$ their mutual distances, $X_{sat}, V_{sat} \in \mathbb{R}^3$ are, respectively, the position and velocity of our probe, $V_c \in \mathbb{R}^3$ is the velocity of the body with respect to which we are accelerating or braking and F_T is the thrust magnitude.

We will consider two strategies to build up the mission. In the first case, the probe will depart from a circular orbit at an altitude of 650 km around the Earth and it will be accelerated in order to gain energy with respect to the Earth and thus to move away from it. When the probe will reach a region where the Moon's influence becomes significant, approximately 67000 km from the Moon [Ron05], we will change the direction of the thruster in order to lose energy with respect to the Moon and to come closer to it. Finally, we will turn the thruster off when the distance from the probe to the Moon is less than $R_{Moon} + 1000$ km, where $R_{Moon} = 1737.5$ km is the Moon's radius [Ron05]. This can be resumed as:

Stage 1: $F_T > 0$ and $V_c = V_{Earth}$.
Stage 2: $F_T < 0$ and $V_c = V_{Moon}$.
Stage 3: $F_T = 0$.

In the above procedure, the most crucial point concerns the instant of thrust direction's change.

Because of this, we will also look at a different approach, which takes advantage of the dynamics associated with the L_1 point. It is well known that in the Circular Restricted Three–Body Problem (CR3BP), there exist five equilibrium points, L_i $(i = 1, \dots, 5)$ and that L_1 is the one which lies between the primaries on the axis joining them. Because of its unstable character, there exist stable and unstable manifolds associated with L_1. Departing from L_1 forwards in time on a trajectory belonging to the unstable manifold, it is possible to get to the Moon or to an orbit close to the Earth, depending on the branch of the manifold chosen. The same result can be obtained by means of the stable manifold going backwards in time.

From these considerations, the second idea applied is to construct a trajectory which passes right through L_1 with null velocity and to carry out the transition between acceleration and braking at that moment. This is, we will consider two trajectories starting from L_1, one moving away from it forwards in time with $F_T < 0$, the other approaching the Earth backwards in time with $F_T > 0$. This can be summarised as

$$\begin{array}{ll} L_1\text{-Earth leg:} & F_T > 0,\ V_c = V_{Earth},\ t < 0, \\ L_1\text{-Moon leg:} & F_T < 0,\ V_c = V_{Moon},\ t > 0. \end{array}$$

However, in the real problem L_1 does not exist, since it is approximately replaced by a quasi–periodic curve whose properties depend on the model adopted ([JV97]). We can translate the above design, by considering a point at the same distance from the Earth in physical units moon-ward side. In the inertial reference frame, this point is no longer characterised by null velocity, but it moves with the same angular velocity as the Moon. This means

$$V_{L_1} = (V_{Moon} - V_{Earth})\frac{r_{EL}}{r_{EM}},$$

where r_{EL} is the distance between Earth and the fictitious L_1 and r_{EM} is the distance between Earth and Moon. In this case, we will consider as initial time $t = JD2454607.1034722$, which corresponds to 20 May 2008 14.29h, when the Moon will be at the apogee with respect to the Earth.

As final remark, according to the data offered by the SMART–1 mission (see [SMA]), we will set the low-thrust magnitude as 5×10^{-6} AU/day^2 (about 0.1 mm/s^2) in both simulations.

Chapter 3

Non validated methods

As a previous step to the so-called validated methods we use non-validated methods with two main purposes: first to get familiarised with the orbits of the problems we are going to deal and second to get an idea of what should expected to obtain when using validated methods.

Within this framework, we have implemented a non-validated Taylor method for ODEs adapted to each of the problems we have considered. The main details of the implementation are commented below.

3.1 The N-body integrator

As it has been mentioned in Section 2, we have modelled the motion of the Solar System by the N-body problem with $N = 11$, where each body represents one of the 9 planets, the Sun or the Moon.

We have coded a Taylor integrator with variable step size and arbitrary order for the jet with respect to time. The user can modify the number of main bodies to consider N, the local tolerance of each step and the order of the method. In most of our computations we have used a local tolerance of $\epsilon = 10^{-20}$ and order $p = 28$.

Remark. According to Proposition 1.2.2 of Chapter 1, assuming the computational effort to be quadratic in p, the optimal order is $p_m \approx -\frac{1}{2}\log\epsilon$ ([Sim01, JZ05]), which gives $p_m = 23$ if $\epsilon = 10^{-20}$. If the computational effort is assumed to be linear, then $p_m \approx -\log\epsilon$. In our case we must consider both effects, $c_1(p+1)^2 + c_2(p+1)$, then one can see that the optimal order for $\epsilon = 10^{-20}$ is $p_m = 28$.

We must mention that for the numerical integration of (2.5) we have taken into account the symmetries of the problem and other properties in order to save computations and speed up the integrator. It is a known fact that the N-body problem conserves the centre of mass. We have fixed the centre of mass at the origin of coordinates, and therefore we compute the position and velocity of the Sun from the position of the other planets. With this we avoid to integrate the Sun's orbit and so save some computational time. Moreover, let us notice that the force exerted from the body X_i to X_j is the same in magnitude but with the opposite direction than the one exerted from the body X_j to X_i. We have also taken this into consideration in order to reduce the number of operations at each integration step.

3.1.1 Comparing the results with JPL 405 ephemerides

In order to verify the results obtained by our Taylor integrator, we have compared the results of integrating the Solar System using our integrator and the JPL 405 ephemerides.

Planet	$\|x_p - x_{jpl}\|$	$\|y_p - y_{jpl}\|$	$\|z_p - z_{jpl}\|$
Mercury	0.81012770E-04	0.72433510E-04	0.13317051E-04
Venus	0.34691978E-04	0.65331954E-04	0.28509546E-08
Earth	0.28927662E-05	0.55188831E-05	0.99721224E-07
Mars	0.20693680E-06	0.28612678E-07	0.22556628E-08
Jupiter	0.62007402E-06	0.17336213E-05	0.36908137E-06
Saturn	0.37777570E-05	0.79455660E-05	0.30570073E-06
Uranus	0.10436420E-05	0.66091594E-06	0.14739934E-08
Neptune	0.78729302E-08	0.48713875E-08	0.17239279E-09
Pluto	0.81168067E-11	0.46916092E-10	0.70336794E-12
Moon	0.89781622E-10	0.90764759E-10	0.17574056E-10

Table 3.1: Difference between the planets position given by our Taylor integrator and by the JPL ephemerides after ≈ 200 years (Final Day: $JD2524466.5$). Units AU.

Planet	$\|vx_p - vx_{jpl}\|$	$\|vy_p - vy_{jpl}\|$	$\|vz_p - vz_{jpl}\|$
Mercury	0.23435797E-04	0.11450443E-03	0.29769501E-05
Venus	0.27469289E-04	0.22077168E-04	0.11440552E-05
Earth	0.11208050E-05	0.20475347E-05	0.82068533E-07
Mars	0.31470357E-07	0.19719165E-07	0.18005013E-08
Jupiter	0.20419279E-04	0.87025726E-04	0.35358497E-05
Saturn	0.31999577E-05	0.61769777E-06	0.17548453E-06
Uranus	0.15234875E-06	0.28771651E-06	0.21109683E-08
Neptune	0.10852279E-08	0.82065225E-09	0.30393440E-10
Pluto	0.86420376E-11	0.31223743E-11	0.46450213E-12
Moon	0.61859111E-05	0.27738011E-05	0.13349501E-06

Table 3.2: Difference between the planets velocity given by our Taylor integrator and by the JPL ephemerides after ≈ 200 years (Final Day: $JD2524466.5$). Units AU/day.

We must mention that the JPL coordinates are given assuming as origin of the reference frame the Solar System Barycentre. In our code, we assume that the origin is set at the centre of masses of the N-body system and use it to compute the Sun's position. As we are neglecting other bodies this centre is different from the complete Solar System Barycentre. So in order to have good initial conditions we must, before starting the simulations, recompute the centre of mass of our system and make a translation in the initial conditions, to place our centre of mass at the origin.

We have made simulations integrating the 11 bodies up to approximately 600 years. In Tables 3.1 and 3.2, we can see the difference in position and velocity, respectively, after 200 years.

Looking at the discrepancy on the planets coordinates, one source of error resides in the relativistic correction which should be applied to the Mercury's orbit. In Figure 3.1, we can see as a function of time the difference between the Mercury's true longitude taking the data from our integration and the data obtained by JPL. In average, the difference $\Delta\omega$, is around 0.000208 radians $\approx 42.972''$ per century.

This quantity is in agreement with the precession of the perihelion of Mercury, due to the

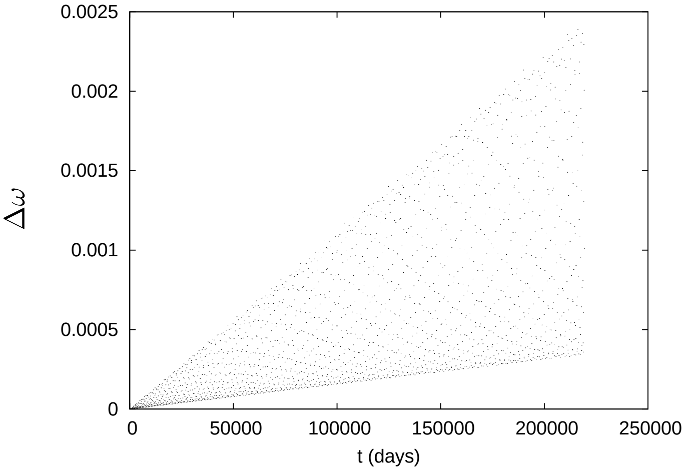

Figure 3.1: Perihelion precession of Mercury obtained integrating the N-bodies problem ($N = 11$). The whole interval of time considered is of 600 years, we can see that in average $\Delta\omega \sim 0.000208$ rad per century.

relativistic effects. After one revolution, that is, 0.2408467 years, it can be estimated as

$$\Delta\omega = \frac{6\pi GM_{Sun}}{a(1-e^2)c^2}\ rad, \tag{3.1}$$

where $a = 57909176000$ m and $e = 0.20563069$ are, respectively, Mercury's semi-major axis and eccentricity, $c = 299792458$ m/s is the speed of light and $GM_{Sun} = 1.32712440018 \times 10^{20}$ m^3/s^2. This is, $\Delta\omega = 42.978''$ per century.

Another considerable effect which has been neglected is the J_2 Earth's gravitational potential harmonic. Indeed, the non-sphericity of the Earth affects the Moon's motion causing the perihelion to advance of

$$\frac{d\omega}{dt} = 3nJ_2R_E^2\frac{1-5\cos^2 i}{4(1-e^2)R^2}, \tag{3.2}$$

where $J_2 = 1.0827 \times 10^{-3}$, $e = 0.0549$ is the Moon's eccentricity, $R_E = 6378.14$ km is the Earth mean radius, $R = 384400$ km is the Earth-Moon distance, $n = 2.65 \times 10^{-6}$ is the Moon's mean motion and $i = 23.45°$ is the Moon's mean inclination with respect to the Earth's equator. With these values, equation (3.2) gives $\Delta\omega \approx -6 \times 10^{-5}$ rad per year. As we can see in Figure 3.2, our simulations (that do not take into account the J_2 effect) give an advance in the perihelion of -5.8×10^{-5} rad per year.

A classical reference for perturbation theory is [Dan88].

3.2 Precision and speed of the integrator

Beyond the accuracy of the model one also must consider the errors done during the integration of the N-body problem with a Taylor integrator.

It is well known that the N-body problem has 10 first integrals, such as, the energy level, the angular momentum and the conservation of the centre of mass. We have already used the conservation of the centre of mass to derive the position of the Sun and save computing time, but we can check the conservation of the other first integrals.

As before, we have done simulations up to 600 years, and we check the variation of these first integrals at every step. In Figure 3.3, we can see that this variation is very small, up to the

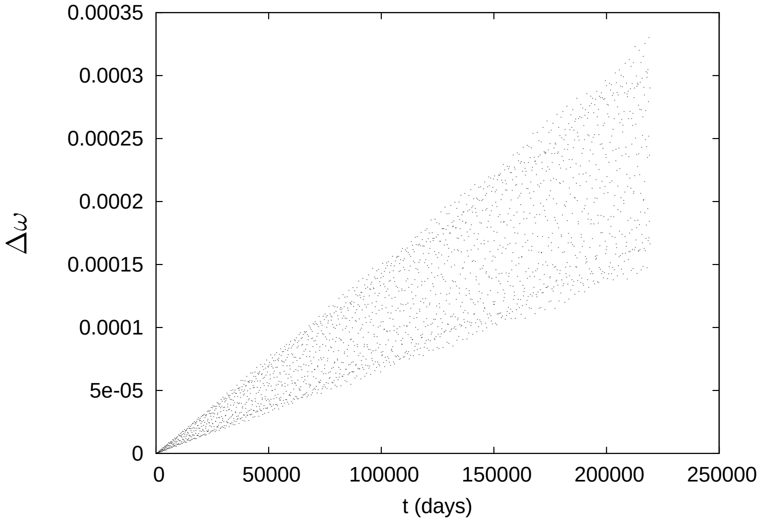

Figure 3.2: Perihelion advance of the Moon obtained integrating the N-bodies problem ($N = 11$). The whole interval of time considered is of 600 years, we can see that in average $\Delta\omega \sim -5.8 \times 10^{-5}$ rad per year.

computer accuracy, and behaves as a random walk. This shows that the only sources of error during our computations is due to the roundoff of the computer as the truncations errors are almost negligible.

We also want to comment on the computational time. At present, our code takes 29.52 seconds of CPU time to integrate the 11 bodies up to 600 years using an Intel Xeon CPU at 3.40GHz.

3.3 Including Apophis

As we have already explained in the previous section, we have modelled the motion of a NEO asteroid (99942 Apophis) by a restricted $(N + 1)$-body problem. We consider 11 main bodies affected by their mutual gravitational attraction but not by the asteroid, which is a massless particle evolving under the gravitational attraction due to the planets, the Sun and the Moon.

With respect to the numerical simulation, we have used the N-body integrator described above to integrate the N massive bodies and used a similar algorithm for the equations concerning the asteroid.

To deal with the imprecision on the position and velocity of the main bodies observed previously (recall Tables 3.1 and 3.2), we propose an alternative method to integrate the motion of Apophis. The main idea is to take advantage of the JPL ephemerides for the main bodies: we use our integrator for Apophis and the JPL integrator for the other bodies. In this way, the position of the major bodies is taken in a more realistic way, as the JPL files takes into account some additional effects which become relevant over long time intervals and which can make the dynamics of our asteroid to vary, specially in close approaches.

We recall that the equation of motion for Apophis can be written as

$$\ddot{X}_a = \sum_{j=1}^{11} \frac{Gm_j(X_j - X_a)}{r_{ja}^3}.$$

Note that to compute the jet of derivatives for X_a we need the jet of derivatives of the bodies X_j. We assume the step of integration to be small enough to take into account just the mutual

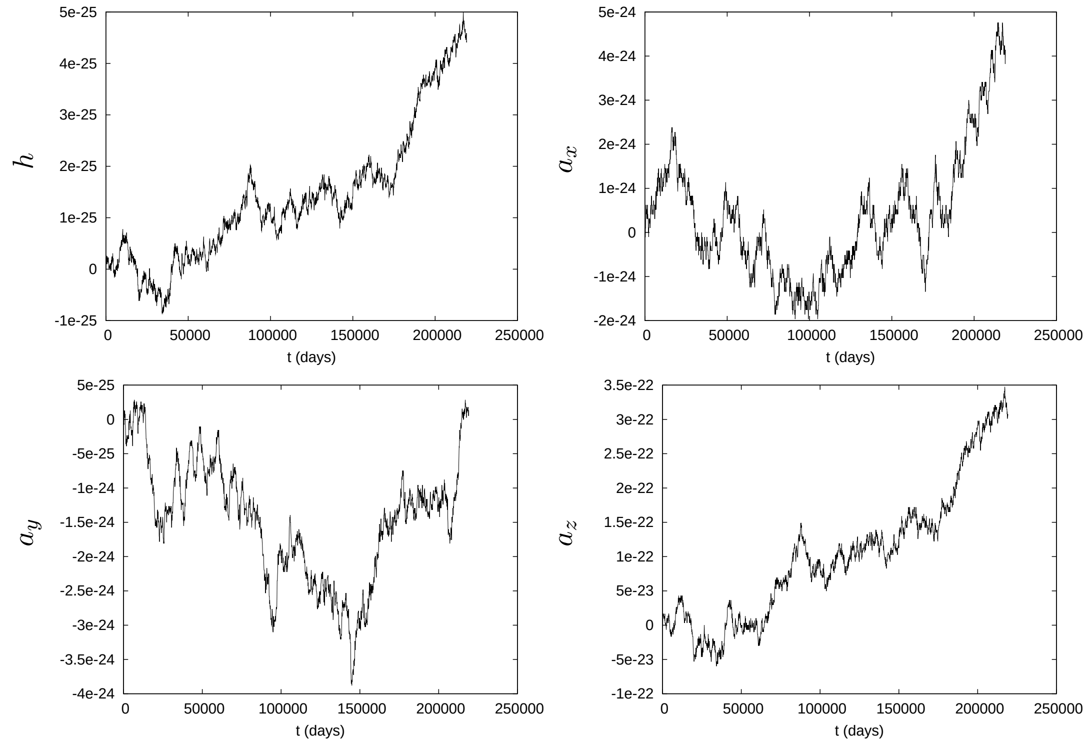

Figure 3.3: From left to right, up and down the variation of the energy h and the angular momentum a_x, a_y and a_z, respectively, for the 11-body problem for a time span of 600 years.

gravitational attraction for the main bodies and to neglect the other effects in JPL. With this we mean, that the jet of derivatives of the major bodies will be computed using the N–body approximation, taking at each step of integration the position given by JPL ephemerides.

3.3.1 Comparing the results with JPL 405 ephemerides

We have first done some computations and compared the results with the JPL Horizons system [JPL]. We have taken the initial data for Apophis and the other 11 massive bodies on 1 September 2006 00:00h ($t = JD2453979.5$) from the JPL Horizons system (see data in Table 3.5). We have done the simulations using both of the schemes mentioned before, with and without corrections on the massive bodies, just before the first close approach of Apophis with the Earth on 13 April 2029 00:00h ($t = JD2462239.5$). In Tables 3.3 and 3.4 we can see the results for these simulations for the two procedures and the results given by the JPL ephemerides.

Method	x	y	z
Integ without JPL	-0.9246754234693604E+00	-0.3936319665713030E+00	-0.8053392702252214E-03
Integ with JPL	-0.9246768454408323E+00	-0.3936299644684824E+00	-0.8055027261187782E-03
JPL ephemerides	-0.9246839460779299E+00	-0.3936206025875185E+00	-0.8061487939761827E-03

Table 3.3: Position of Apophis on 13 April 2029 00:00h doing the integration with the JPL, without JPL and the data given by JPL ephemerides. Units AU.

As we can see, if we take the corrections of the planets by JPL at every time step we get a better approximation than without taking them into account.

In Figure 3.4, we show the results concerning the numerical simulation of the Apophis'

Method	vx	vy	vz
Integ without JPL	0.8889495406390923E-02	-0.1368219560142266E-01	0.9635559643139474E-03
Integ with JPL	0.8889374524479276E-02	-0.1368219167106837E-01	0.9635357380844921E-03
JPL ephemerides	0.8889257217928338E-02	-0.1368237615492765E-01	0.9635446437798482E-03

Table 3.4: Velocity of Apophis on 13 April 2029 00:00h doing the integration with the JPL, without JPL and the data given by JPL ephemerides. Units AU/day.

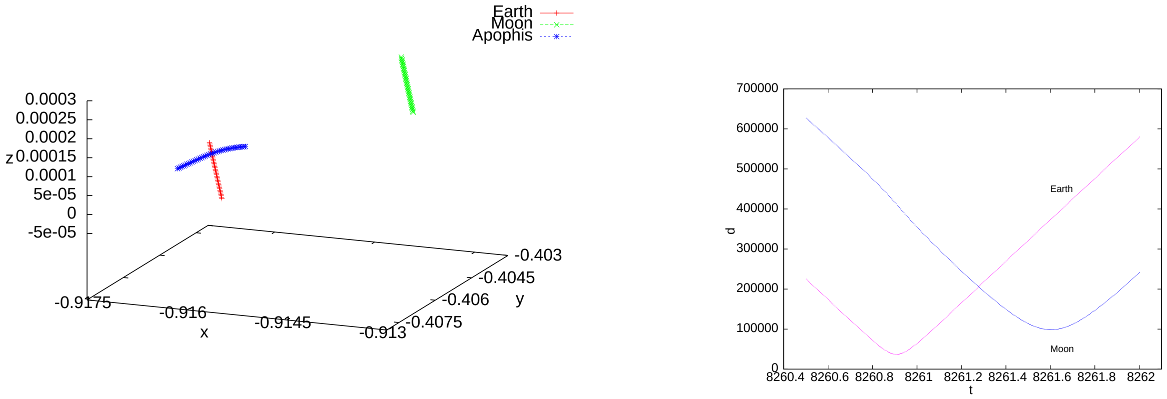

Figure 3.4: On the left, for $t \in (JD2462239.5, JD2462240.5)$ the orbit of the Earth (in red), of the Moon (in green) and of Apophis. The reference system displayed is centred at the Solar System Barycentre. Unit AU. On the right, the distance between Apophis and the Earth (in pink) and the distance between Apophis and the Moon (in blue) as function of time for $t \in (JD2462240, JD2462241.5)$. Units km and days.

trajectory, taking as initial condition the one given in Table 3.5. On the left, we display the close approach with the Earth corresponding to $t \in (JD2462239.5, JD2462240.5)$; on the right, the distance between Apophis and the Earth and the one between Apophis and the Moon as function of time in the interval $t \in (JD2462240, JD2462241.5)$.

3.3.2 The non-rigorous propagation of an initial box

In order to have an idea of what should be expected by a validated method we have propagated an initial box using our non-validated code. The idea is to iterate a mesh of points on a box. In this particular case the initial box represents the uncertainty in the determination of the position of the asteroid.

As we have already said, information on the physical and orbital data of Apophis can be found in the NEODYS website ([NEO]). According to recent observations of Apophis [GBO$^+$08], we have a standard deviation on the semi-major axis, a, $\sigma_a \approx 9.6 \times 10^{-9}$ AU and a standard deviation on the mean anomaly, M, $\sigma_M \approx 1.08 \times 10^{-6}$ degrees. Hence, σ_a gives an uncertainty of about 1.44 km in the determination of the position of the asteroid, and σ_M gives an uncertainty of about 2.59 km on the velocity's direction, as $a \approx 0.92$ AU.

According to these data, we have chosen an initial box centred at the initial condition given in Table 3.5, of 7 km long on the tangent to the orbit direction and 3 km long on two other given orthogonal directions.

In our code, the user can choose the mesh of points on the initial box, the total interval of time and the time step to print out the results. For instance, we have propagated $10 \times 4 \times 4$

Data from [GBO$^+$08]			
	x	y	z
position	0.5166128258669076E+00	0.6961955810635310E+00	-0.2443608670809208E-01
velocity	-0.1295180180760195E-01	0.1388132695417834E-01	-0.1047646475022484E-02
Horizons			
	x	y	z
position	5.166129167886292E-01	6.961955223318413E-01	-2.443608650998807E-02
velocity	-1.295180048414146E-02	1.388132804750336E-02	-1.047646730942868E-03

Table 3.5: Initial position and velocity for Apophis on 1 September 2006 00:00h, given by [GBO$^+$08] and by the Horizons system.

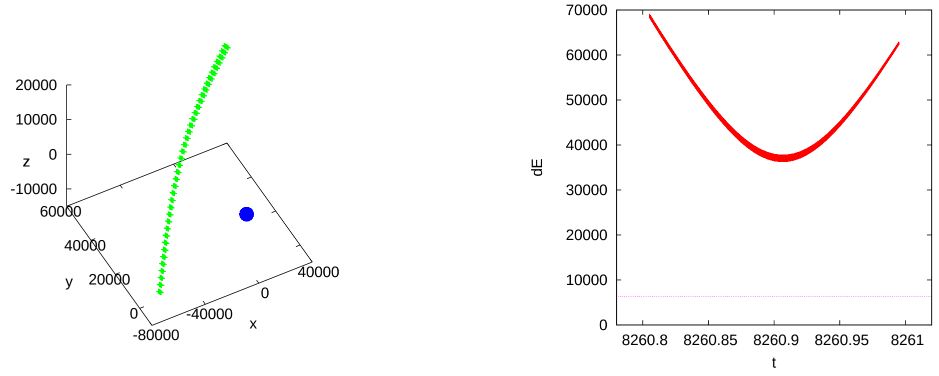

Figure 3.5: On the left, for $t \in (JD2462240.3, JD2462240.5)$ the orbit corresponding to the box representing the uncertainty in space for Apophis. The reference system displayed is centred at the Earth. The blue point is the Earth. On the right, the distance between the same box and the Earth as function of time for $t \in (JD2462240.3, JD2462240.5)$. Units km and days.

points over about 23 years plotting every 0.5 days when Apophis has a close approach with the Earth.

In Figure 3.5, we show, on the left, the orbit of the given initial box propagated up to the first close approach with the Earth. On the right, the distance between the box and the Earth: the closest point to the centre of the Earth reaches a distance of about 36000 km.

In the simulations, we have observed that the boxes stretch out along the direction of the orbit as time goes by. This is something that can be expected. As already mentioned, as a first approximation the motion of the asteroid can be seen as a two-body problem, just considering Sun and asteroid. Thus its trajectory can be approximated by a Keplerian orbit and the estimates given in Section 2.1.1 hold.

3.3.3 The non-rigorous propagation of an initial box using variational equations

As mentioned in the introduction, a validated method requires information at least on the first order variational equations to describe how an initial uncertainty evolves in time. In some cases, when the error grows we need to consider a higher order approximation for the dynamics. One can consider a k order jet of derivatives with respect to time and a j order jet of derivatives with respect to spatial coordinates. It is then relevant to know which (k, j)-jet of Taylor method provides an accurate information of the system.

On the other hand, when we deal with a non-validated method, the behaviour of a random box can be described by the variational equations. In general, they provide accurate information on the evolution over time of close initial conditions with a lower computational effort (see Appendix A).

In particular, we have considered the first and second order variational equations to approximate the random box along time. We are interested in knowing if the first order one gives enough information or if we need to take into account the second order one.

First, this has been done starting from $t = JD2453979.5$ (01 September 2006 00:00h) up to $t = JD2462240.5$ (14 April 2029 00:00h). The results show that we need the quadratic approximation furnished by the second variational equations to describe the first estimated close approach of Apophis with the Earth. In particular, this is needed in the interval of time $t \in (JD2462240.42, JD2462240.5)$, which covers about 2 hours. Apart from that, the linear information gives a very good estimation of the box. Looking to Figures 3.6 and 3.7, we can appreciate when the Earth's attraction starts bending the box, that is, when the linear approximation provided by the first variational equations is no longer enough to describe the dynamics.

As a further step, we have tried to figure out if the same approach could be considered for the second estimated close approach, that is, up to $t = JD2465323.5$ (22 September 2037 00:00h). It turns out that, after the first passage, the dynamics becomes very sensitive to the initial conditions and thus there exists an instant of time from which we are no longer able to predict the behaviour of the box. In our simulation, we have found this discrepancy after approximately 6 years from $JD2462240.5$.

Concerning the CPU computational time, our code spends 1.169 s to integrate the vector field, 1.572 s to integrate the first variational equations and 11.129 s to integrate the second order ones from 1 September 2006 to 13 April 2029. This has been done using an Intel Xeon CPU with 2.66GHz. For more details on the computation of the variational equations see Appendix A.

3.4 Low-Thrust

With respect to the low-thrust transfer, we have first integrated the system of equations (2.7) in Section 2.4 defined in the previous chapter by a Taylor non-validated integrator of order 22 and variable step size, setting as local tolerance 10^{-20}. Given the results of the above sections, we consider the planets' position and velocity to be determined by the JPL ephemerides.

In these simulations, we have observed a significant loss of digits, both departing from the Earth or from the instantaneous point L_1. This fact can mainly be explained by two considerations. System (2.7) faces the problem in an inertial reference system with origin at the Solar System Barycentre, but the motion of the spacecraft takes place in the Earth–Moon neighbourhood. This means that we are dealing with an initial condition which keeps few information about the dynamics we are interested in. On the other hand, in both strategies (see Section 2.4) the orbit obtained consists of several ellipses characterised by different values of semi–major axis. Departing from the Earth, the extra force introduced is not as big as to gain altitude soon and thus the spacecraft moves very fast performing thousands of revolutions before getting to the target. The same argument holds when we depart from the instantaneous L_1, as the probe gets closer to the Earth backwards in time.

To overcome this problem, we rewrite the equation of motion in order to take into account the relative position and velocity of the probe with respect to the Earth (or the Moon). If

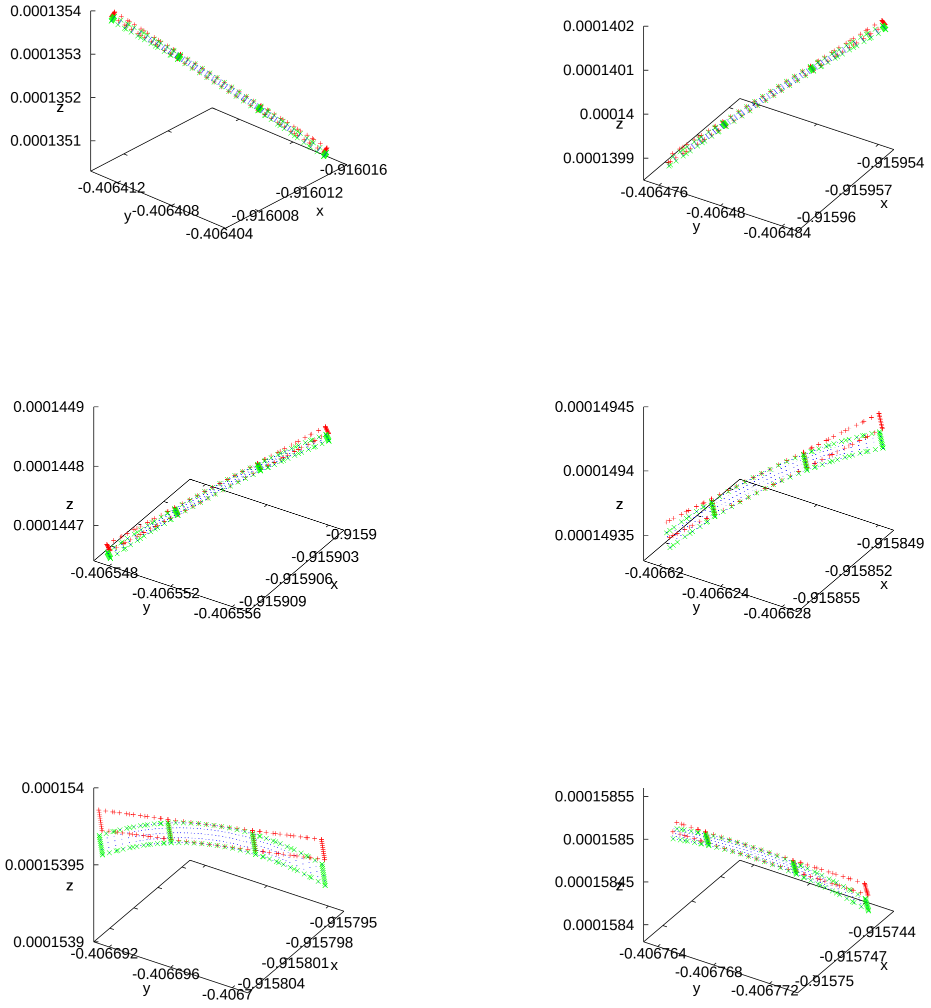

Figure 3.6: In blue, the box of uncertainty in space corresponding to Apophis, in red and in green the information provided by the variational equations of first and second order, respectively. The interval of time is $t = [JD2462240.435, JD2462240.46]$.

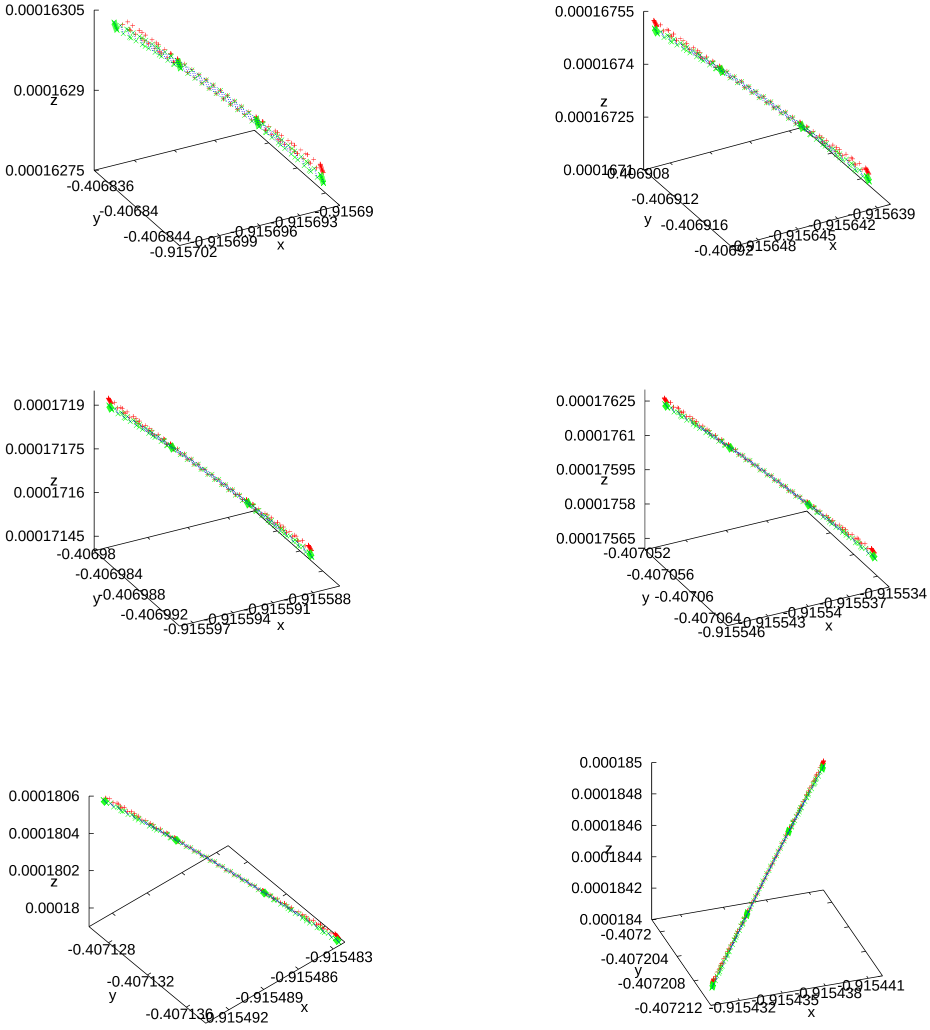

Figure 3.7: In blue, the box of uncertainty in space corresponding to Apophis, in red and in green the information provided by the variational equations of first and second order, respectively. The interval of time is $t = [JD2462240.465, JD2462240.495]$.

$Y = X_{sat} - X_c$ and $VY = V_{sat} - V_c$, then the new system of equations will be

$$\begin{aligned} \ddot{X}_i &= \sum_{j=1, j\neq i}^{11} \frac{Gm_j(X_j - X_i)}{r_{ji}^3}, \qquad \text{for } i = 1, \ldots, 11 \\ \ddot{Y} &= \sum_{j=1}^{11} \frac{Gm_j(X_j - X_c - Y)}{r_{jcY}^3} - \ddot{X}_c + F_T \frac{VY}{\|VY\|}, \end{aligned} \tag{3.3}$$

where r_{jcY} is the distance between the planet j ($j = 1, \ldots, 11$) and the spacecraft.

At each integration step we keep on taking the data relative to the planets from the JPL model. On the other hand, we notice that the change of variables just introduced does not affect the computation of the jet of derivatives associated with the planets. Indeed, we are not modifying the reference frame and thus the only force acting on the main bodies is still their mutual gravitational attraction.

To illustrate the improvement resulting from this choice, in Tables 3.6 and 3.7 we show the relative and absolute error obtained by integrating equation (2.7) and equation (3.3) starting from a same initial condition up to 730.5 days. We have checked how many digits we were losing in double precision taking the solution computed in quadruple precision as the exact one.

Eq.	relative error	absolute error
(2.7)	1.56e-9	1.53e-9
(3.3)	7.16e-13	7.e-13

Table 3.6: Relative and absolute error in position obtained by integrating equations (2.7) and (3.3) starting from a same initial condition up to 730.5 days. The errors refer to the results obtained in double precision and considering the solution obtained with quadruple precision as the exact one.

Eq.	relative error	absolute error
(2.7)	1.03e-5	1.41e-7
(3.3)	1.45e-9	2.e-11

Table 3.7: Relative and absolute error in velocity obtained by integrating equations (2.7) and (3.3) starting from a same initial condition up to 730.5 days. The errors refer to the results obtained in double precision and considering the solution obtained with quadruple precision as the exact one.

3.4.1 Nominal orbits

In order to get two nominal orbits to work with, we have applied the mission designs mentioned in Section 2.4 by means of the new system of equations (3.3).

In the first case, we set as departure orbit a circular orbit around the Earth with radius $r = 650$ km and inclination $i = 23.5°$ with respect to the Earth's equator. To find a nominal trajectory which gets to the Moon, we integrate forwards in time equation (3.3) setting $F_T > 0$ and $V_c = V_{Earth}$ and taking as initial conditions $1257 \approx 2\pi/(5 \times 10^{-3})$ equally spaced points on the given circular orbit. For each of these trajectories, we change the thrust direction, i.e. $F_T < 0$ and $V_c = V_{Moon}$, when the distance to the Moon becomes smaller than 67000 km. Finally, when

$x_{sat} - x_{Earth}$	$y_{sat} - y_{Earth}$	$z_{sat} - z_{Earth}$
-1.4666642042013140e-05	4.0598584396571580e-05	1.8541442146629470e-05

Table 3.8: Initial condition in position for the first nominal orbit of the low-thrust mission considered. Initial time $JD2454371.6034722$ (28 September 2007 02:29h). Unit AU.

$\dot{x}_{sat} - \dot{x}_{Earth}$	$\dot{y}_{sat} - \dot{y}_{Earth}$	$\dot{z}_{sat} - \dot{z}_{Earth}$
-4.0476276575951110e-03	-1.5733606852764890e-03	2.4329879836049721e-04

Table 3.9: Initial condition in velocity for the first nominal orbit of the low-thrust mission considered. Initial time $JD2454371.6034722$ (28 September 2007 02:29h). Unit AU/day.

$x_{sat} - x_{Earth}$	$y_{sat} - y_{Earth}$	$z_{sat} - z_{Earth}$
-9.5667391869508616e-04	-1.8395563106682602e-03	-1.0105503932953232e-03

Table 3.10: Initial condition in position for the second nominal orbit of the low-thrust mission considered. Initial time $JD2454607.1034722$ (20 May 2008 14.29h). Unit AU.

$\dot{x}_{sat} - \dot{x}_{Earth}$	$\dot{y}_{sat} - \dot{y}_{Earth}$	$\dot{z}_{sat} - \dot{z}_{Earth}$
4.3272284589505872e-04	-1.8632719222831515e-04	-7.0465113441901608e-05

Table 3.11: Initial condition in velocity for the second nominal orbit of the low-thrust mission considered. Initial time $JD2454607.1034722$ (20 May 2008 14.29h). Unit AU/day.

the spacecraft reaches a distance less than 1000 km from the Moon's surface (considered as a sphere), we turn the engine off.

In this way, we have found as nominal orbit the one shown in Figure 3.8. The corresponding initial condition is given in Tables 3.8 and 3.9, the initial time $t = 0$ corresponds to $JD2454371.6034722$ (28 September 2007 02:29h) and the manoeuvre is performed after approximately 741 days. One day after we switch the engine off; the probe gets to the Moon after about $t_{tot} = 765$ days.

We would like to point out that not all the initial conditions considered accomplish the condition of getting to a distance less of 67000 km from the Moon, neither this requirement is enough to guarantee to be captured by the Moon. In the two–body problem, the gravitational capture by the Moon (or the Earth) could be defined as the moment at which the energy of the probe with respect to the main body changes from positive to negative. Unfortunately, our problem cannot be approximated by such a model, since we need to consider at least three main bodies, Earth, Moon and Sun, to affect the spacecraft. As a consequence, we cannot apply the above criterion to carry out the manoeuvre.

To illustrate this, in Figure 3.9 we show the two–body energy, see equation (2.2), with respect to the Earth and to the Moon as function of time up to the manoeuvre and after that. We can note that the thrust direction's change takes place when the energy with respect to the Moon is still positive and the one with respect to the Earth is still negative. At the same time,

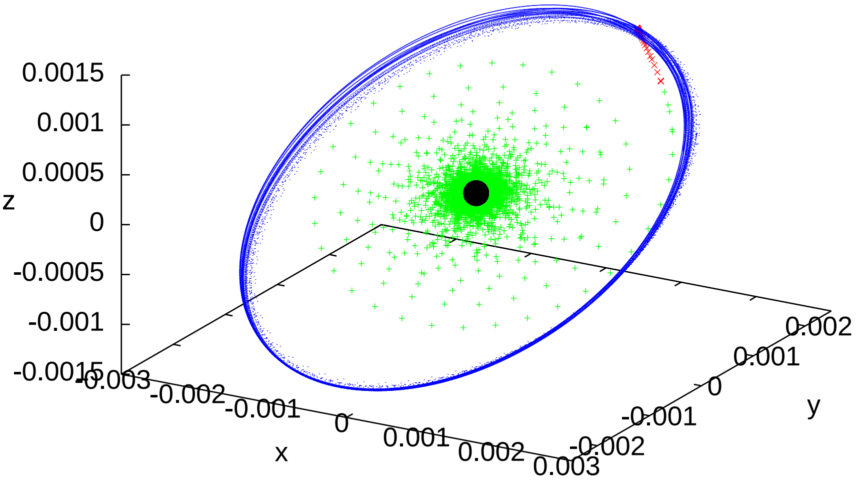

Figure 3.8: First nominal orbit considered for the low-thrust mission. Inertial reference frame centred at the Earth. In green, trajectory's leg characterised by $F_T > 0$; in red, by $F_T < 0$. In black, the Earth; in blue, the orbit of the Moon. Unit AU.

when we are braking towards the Moon, the energy with respect to the Earth and the one with respect the Moon oscillate from positive to negative. This behaviour is due to the perturbations introduced on the probe by the planets, which make the dynamics corresponding to the probe very sensitive to the initial condition at the time of performing the manoeuvre.

We have decided to turn the thruster off when the distance to the Moon is less than 2737.5 km for the same reason as above. By changing a little this value, the spacecraft could not achieve the goal of the mission, i.e. to get to the Moon.

We have also considered another nominal trajectory, starting from the instantaneous L_1 corresponding to $JD2454607.1034722$ and integrating equation (3.3) backwards in time with $F_T > 0$ and $V_c = V_{Earth}$ and forwards in time with $F_T < 0$ and $V_c = V_{Moon}$. The initial condition corresponding to L_1 is given in Tables 3.10 and 3.11. The trajectory is shown in Figure 3.10. In this case, the total transfer time is about $t_{tot} = t_{Moon} + t_{Earth} \approx 45 + 245$ days.

Comparing the two trajectories above, we note that in the first case we need more than two years to get to the Moon, while in the second one less than one. On the other hand, the trajectory passing right through L_1 flows from a departure and an arrival orbit characterised by a great value of eccentricity ($e \approx 0.8$), which is actually not a realistic parameter.

3.4.2 The non-rigorous propagation of an initial box using variational equations

With respect to the non-rigorous propagation of an initial box, we follow the same procedure as done in the case of Apophis. This is, we take an orthonormal basis, composed by the tangent to the orbit vector and by two vectors orthogonal to this one and we consider how an uncertainty in position is reflected on these directions. With the actual technologies, the position of a probe can be measured with a high level of accuracy: we set the box to be 30 cm long on the three

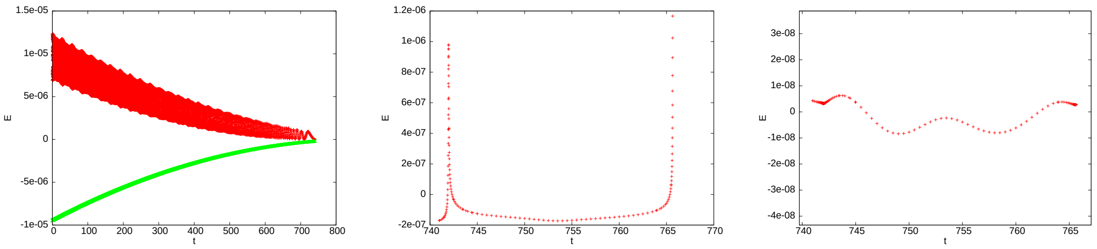

Figure 3.9: On the left, two–body energy, see equation (2.2), with respect to the Earth (green) and to the Moon (red) as function of time up to the manoeuvre. On the middle, two–body energy with respect to the Earth after the manoeuvre. On the right, two–body energy with respect to the Moon after the manoeuvre. We can note that the thrust direction's change takes place when the energy with respect to the Moon is still positive and the one with respect to the Earth is still negative. At the same time, when we are braking towards the Moon, the energy with respect to the Earth and the one with respect the Moon oscillate from positive to negative.

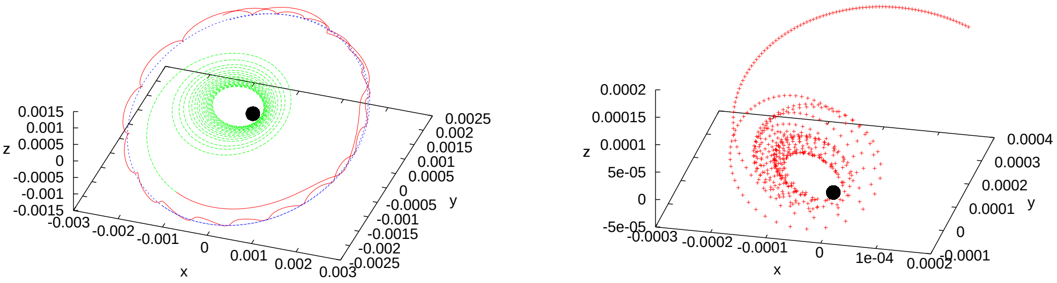

Figure 3.10: Second nominal orbit considered for the low-thrust mission. On the left, in the inertial reference frame centred at the Earth we show the trajectory's leg characterised by $F_T > 0$ (green) and the one corresponding to $F_T < 0$ (red). In black, the Earth; in blue, the orbit of the Moon. On the right, in the inertial reference frame centred at the Moon the part of the trajectory associated with $F_T < 0$ in red and the Moon in black. Unit AU.

directions.

Applying the variational terms of first and second order to the orthonormal basis at each step of the integration and to both the nominal trajectories, we can approximate very well the dynamics obtained. In particular, the quadratic approximation is needed when the probe is approaching the Moon or when a considerable interval of time has elapsed. In Figure 3.11, we show some results associated with the L_1-Moon leg.

It is worth to point out that a difficulty arises at the time of performing the manoeuvre. As said before, for the nominal orbit we have decided to change the thrust direction when the probe reaches a distance to the Moon less than 67000 km, which corresponds to a well–defined instant of time. However, each initial condition included in the given box fulfils the above requirement at a different value of time. We can decide to fix as instant of force's change the one associated with the nominal trajectory, that is, the same for all the points of the initial box. Or, to carry out the manoeuvre in a independent way one point to the other, that is, at different times according to the trajectory considered.

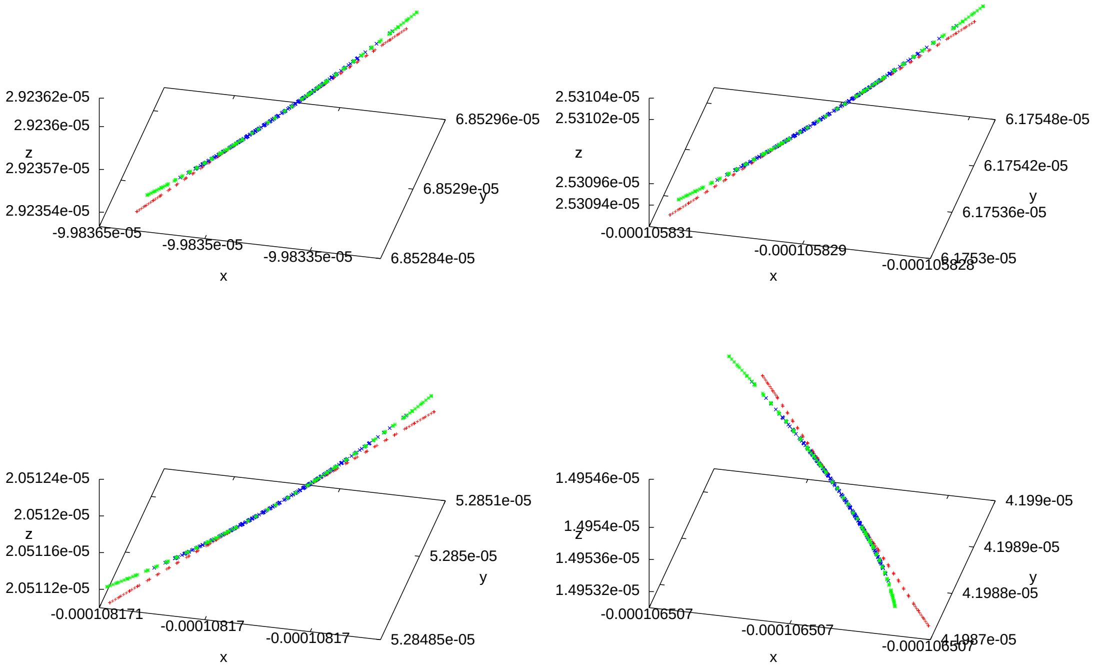

Figure 3.11: In blue, the box of uncertainty in space corresponding to the probe, in red and in green the information provided by the variational equations of first and second order, respectively. The initial condition considered corresponds to the instantaneous L_1 at $t_0 = JD2454607.1034722$, see Tables 3.10 and 3.11. The leg of trajectory is characterised by $F_T < 0$ and $V_c = V_{Moon}$. The interval of time displayed is $t = [t_0 + 39.609375, t_0 + 39.84375]$.

As explained before for the Apophis case, a random box stretches out along the tangent to the orbit direction. Hence the first option gives values of distance to the Moon (and of Keplerian energy) which are significantly different for one point to the other. We have already mentioned that the dynamics is quite sensitive to the initial condition when we are approaching the Moon and how this can produce an effect on getting there. On the other hand, to apply the second choice means that we are considering not one mission, but as many as the number of initial conditions in the box. These considerations hold even more for the validated integration.

Chapter 4

Validated methods: theory

Since R. Moore started the study of validated methods for ODEs, different methods have been suggested to improve the original Moore's algorithm (an implementation of the original algorithm by Lohner is in the AWA code [Loh] which also contains implementations of the parallelepiped method and the C^0 QR-Lohner method). Mainly, the problem of the initial algorithm is the so-called wrapping effect as R. Moore already stated in his celebrated example of a rotation (a brief explanation is given in Section 1.3.3). Below we describe the main lines of improvement of the classical algorithm.

To fix ideas, let $u(t; u_0)$ be the solution of the initial set value problem (ISVP)

$$u' = f(u), \quad u(t_0) = u_0 \in \{u_0\}, \tag{4.1}$$

where $f : \mathbb{R}^m \to \mathbb{R}^m$ defines an analytic vector field, $u_0 \in \mathbb{R}^m$ and $\{u_0\}$ is a set of $\mathbb{R}^m$. The ISVP should be understood in the following way: given $h > 0$ we look for a set $\{u_1\}$ of $\mathbb{R}^m$ such that $u(t_0 + h; u_0) \in \{u_1\}$ for all $u_0 \in \{u_0\}$. We have then a map T mapping a region $\{u_0\}$ to a region $T(\{u_0\}) \subset \{u_1\}$. The idea is to implement the evaluation of T in such a way that the set $\{u_1\}$ is as close as possible to the set $T(\{u_0\})$. The difference between the two sets is mainly due to the wrapping effect, but the so-called dependency problem also plays a role.

Two different approaches can be found in the literature: the so-called Interval methods and the Taylor-based methods. Both approaches try to reduce the wrapping and the dependency effect by modifying the original algorithm in an effective way. A precise explanation of both methods mainly for linear ODEs as well as a comparison between them for specific linear examples can be found in [NJN07]. For the sake of completeness, in the following we give an explanation of the main ideas of both type of methods.

In this report, we will concentrate in examining the Interval methods although they are not so accurate as the Taylor-based methods. The main reason to explore these methods is that Interval methods for rigorous integration of an ODE are faster than Taylor-based methods. A comparison of the two methods is given in [HB03] where the integration of the asteroid 1997 XF11 using a Kepler model and using a model of the full Solar System is carried out by using both types of methods. In this paper the authors note that although the Taylor-based integration is more stable and accurate the method is around 50 times slower that an Interval method. Some results concerning the same example can also be found in [BMH01].

In both strategies of integration several problems related to the interval representation of the sets appear as will be explained now.

4.1 Interval based methods

This family of methods contains the Moore's original algorithm and the later modifications: the parallelepiped method, the Lohner algorithm and the C^r-Lohner algorithms. The main idea of these methods is to represent sets in terms of interval boxes (product of intervals of $\mathbb{R}$) in a suitable form. There are many ways to represent sets in terms of intervals. In [MZ00] there is a list of some useful representations. Using the notation of [MZ00] we are mainly going to deal with the interval set representation, parallelepiped representation and the cuboid representation. We also propose a modification of the cuboid representation which uses the fact that the vector field is a second order one.

4.1.1 Moore's direct algorithm

We start describing the ideas of the Moore's direct algorithm. In Moore's algorithm sets are represented using interval set representation, that is, a set $\{u_0\}$ is represented by an interval box $[u_0]$ which is a direct product of intervals of $\mathbb{R}$. Let $\mathbb{IR}^m$ the set of intervals of $\mathbb{R}^m$. From now on $[x] \in \mathbb{R}^m$ will denote an interval set representation of a given set $\{x\}$. We have then $\{x\} \subset [x]$ and we expect the difference between both sets to be not very large (although it can be large enough to produce the failure of the method as is going to be shown).

For $u_0 \in \{u_0\}$, Taylor expansion of $u(t_0+h;u_0)$ up to order n around $t=t_0$ gives

$$u(t_0+h;u_0) = T(u_0) + R(\xi;u_0),$$

where

$$T(u_0) = u_0 + f(u_0)h + \cdots + \frac{d^{n-1}}{dt^{n-1}} f(u_0)\frac{h^n}{n!},$$

and

$$R(\xi;u_0) = \frac{d^n}{dt^n} f(u(\xi;u_0))\frac{h^{n+1}}{(n+1)!},$$

with $\xi \in [t_0, t_0+h]$.

To make this evaluation rigorous it is important to note that for all $u_0 \in \{u_0\}$ it is $T(u_0) \subset T([u_0])$, where

$$T([u_0]) = [u_0] + f([u_0])h + \cdots + \frac{d^{n-1}}{dt^{n-1}} f([u_0])\frac{h^n}{n!},$$

and where an interval extension of f is considered, that for convenience we denote by the same symbol f. We recall that a map $\hat{f} : \mathbb{IR}^m \to \mathbb{IR}^m$ is said to be an interval extension of $f : \mathbb{R} \to \mathbb{R}$ if $\hat{f}\,|_{\mathbb{R}} = f$, where by $\hat{f}\,|_{\mathbb{R}}$ we denote the restriction of the map $\hat{f}$ to the point interval sets.

To have a rigorous way to evaluate $T(\{u_0\})$, it remains to obtain an interval containing $R(\xi;u_0)$ for all $u_0 \in \{u_0\}$. Note that if $[\hat{u}_0]$ is an interval box such that $u(t;u_0) \subset [\hat{u}_0]$ for all $t \in [t_0, t_0+h]$ and for all $u_0 \in [u_0]$, then

$$R(\xi, u_0) \subset R([\hat{u}_0]),$$

and, as a consequence,

$$[u_1] = T([u_0]) + R([\hat{u}_0]),$$

which gives a solution of the ISVP at time $t+h$.

To obtain the interval $[\hat{u}_0]$ it is proposed the iteration

$$\begin{aligned} [\hat{u}_0^0] &= [u_0] + [\epsilon, \epsilon], \\ [\hat{u}_0^{k+1}] &= [u_0] + [0,h] f([\hat{u}_0^k]), \end{aligned} \tag{4.2}$$

whose convergence can be justified if h is small enough. This provides the interval box $[\hat{u}_0]$ with the required properties. A theoretical justification of this iteration can be found, for instance, in [Ned99]. This interval is denoted as *rough enclosure*.

This method provides a rigorous integrator of an ODE. Nonetheless, among the inconveniences of this method we note

1. The evaluation of $T([u_0])$ depends on the interval extension of f. This is the so-called dependency problem and can be reduced by a suitable implementation of the algorithm.
2. The set $\{u_1\} = \{T(u_0), u_0 \in [u_0]\}$ is not an interval box and hence we need to include it in $T([u_0])$ giving rise to the so-called wrapping effect.
3. The iteration (4.2) implies the use of a small step size (mainly reduced to the size of the Euler method). To have convergence of the method requires h to be less than L^{-1} where L is the Lipschitz constant of f. We note that other approaches as polynomial enclosures provide large step size for verified integration (see [Ned99]).

The most relevant inconvenience of the described method is the wrapping effect which implies the rigorous integration of solutions that do not come from the initial set $[u_0]$ making the interval enclosures growing fast and invalidating the method. The wrapping effect is a consequence of the way we represent the sets. In any representation we choose we will have wrapping effect although the choice of a suitable representation can reduce it.

4.1.2 The parallelepiped method, the QR-Lohner method and new modifications

As mentioned above, the inclusion $T(u_0) \subset T[u_0]$ used in the Moore's algorithm is the point where the wrapping plays a role. In order to analyse this evaluation carefully we rewrite it in centred form way. Note that

$$T[u_0] = T(m(u_0)) + DT([u_0])([r_0]),$$

where $m(u_0)$ denotes the mid point of the interval set and $[r_0] = [u_0] - m(u_0)$. Hence, Moore's algorithm in centred form reads

$$[u_1] = T(m(u_0)) + DT([u_0])([r_0]) + [z_1], \tag{4.3}$$

where $[z_1] = R([\hat{u}_0])$.

Remark. In general, it is better to use centred forms rather than direct forms. The idea behind this is to reduce the effect of the interval arithmetic to high order terms in the evaluation. As a general rule the interval evaluation should be the latest and at the highest possible order.

Parallelepiped method

The basic idea of the parallelepiped method is to represent a set in terms of the parallelepiped representation (see [MZ00]), that is, the set $\{u\}$ is included in a set of the form $p + A[r]$, where $p \in \mathbb{R}^n$ is a point, $A \in \mathbb{R}^{n\times n}$ is a matrix and $[r] \subset \mathbb{R}^n$ is an interval box set.

To be precise, assume $\{u_0\} = m(u_0) + A_0[\hat{r}_0]$. Then,

$$T(\{u_0\}) \subset T(m(u_0)) + DT([u_0])A_0([\hat{r}_0]) + [z_1] \subset \{u_1\}. \tag{4.4}$$

where we want the set $\{u_1\}$ to be of the form $\{u_1\} = m(u_1) + A_1[\hat{r}_1]$.

Note that $m[u_1] = T(m(u_0)) + m(z_1)$. Then, it turns out that $DT([u_0])A_0([\hat{r}_0]) + [z_1] - m(z_1)$ should be $A_1[\hat{r}_1]$ for a suitable A_1. Setting $A_1 = m(DT([u_0]A_0))$ (or $A_1 = m(DT([u_0]))A_0$) giving the same numerical results) and

$$[\hat{r_1}] = [B_1][\hat{r}_0] + [A_1^{-1}]([z_1] - m(z_1)), \tag{4.5}$$

then

$$DT([u_0])A_0([\hat{r}_0]) + [z_1] - m(z_1) = A_1[\hat{r_1}],$$

where the factorisation $DT([u_0])A_0 = A_1[B_1]$ is considered. Note that in this evaluation mainly the second term is affected by wrapping effect and it can be considered small (increasing the order of the Taylor expansion for instance).

Remark.

1. To get the factorisation $DT([u_0])A_0 = A_1[B_1]$ we proceed in the following way. Put
$$DT([u_0])A_0 = m(DT[u_0]A_0) + DT[u_0] - m(DT[u_0]A_0) = A_1 + [\tilde{B}_1],$$
where $[\tilde{B}_1] = DT[u_0] - m(DT[u_0]A_0)$. Then,
$$DT([u_0])A_0 = A_1[B_1],$$
with $[B_1] = Id + [A_1^{-1}][\tilde{B}_1]$.

2. To compute the rigorous inverse $[A^{-1}]$ of a point matrix A we start by computing an approximate inverse $B = A^{-1}$ (computed non-rigorously). Then, it is $[B][A] = Id + [\delta]$ which implies
$$A^{-1} = (Id + [\delta])^{-1}B \subset B\left(Id \pm \frac{\|[\delta]\|_\infty}{1 - \|[\delta]\|_\infty}\right) = [A^{-1}],$$
where $\|[\delta]\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{n} |[\delta_{ij}]|$.

Note that instead of computing $DT(\{u_0\})$ we evaluate $DT[u_0]$ as was noted in the formula (4.4). This is because to evaluate the jet of a set of the form $p + A[r]$ using interval arithmetic needs an interval as input. On the other hand, the change of variables $x = A\xi$, where ξ are the standard variables we use in computations, gives an initial set which is the interval $A^{-1}p + [r]$ but becomes necessary to compute the jet of derivatives of $A^{-1} \circ f \circ A$ which is not an obvious task. This evaluation on an overestimation of the set $\{u_0\}$ will be analysed in detail later when explaining a method (KT method) which allows the computation on the exact interval $A^{-1}p+[r]$ for a suitable choice of the matrix A in the representation of the sets.

It is remarkable that, in the evaluation of a step of the parallelepiped method, the interval evaluation is done in the first order differential $DT[u_0]$. This suggests to use centred form again modifying the method in order to use the interval evaluation in higher derivatives. For instance, instead of using the expression (4.3) it is possible to use the following
$$[u_1] = T(m(u_0)) + DT(m(u_0))A_0[\hat{r}_0] + \frac{1}{2}(A_0[\hat{r}_0])^t D^2T([u_0])(A_0[\hat{r}_0]). \tag{4.6}$$
One expects this modification to be useful when there is a relevant difference of the first variational in different points of the set $\{u_0\}$ and that this difference is well-approximated by the second variational effect. In this case, we choose $A_1 = DT(m(u_0))A_0$ and we proceed in a similar way as in the first order case.

In general, the parallelepiped method is almost as inefficient as the direct method. The main inconvenience is that the wrapping effect is reduced by a change of variables A_1 which should be rigorously inverted to compute $\hat{r}_1$. As was observed in [NJ01] for the particular case of a planar linear system with different magnitude eigenvalues, after some time steps the matrix A_1 becomes singular. Although in the case of a linear ODE with suitable linear part the parallelepiped method is able to reduce the wrapping effect, its effectiveness for non linear systems depends on the concrete case considered.

We have observed in Section 2.1.1 that the stretching that a box suffers along the orbit behaves linearly (not exponentially) with respect to the time. However, after enough revolutions the stretching becomes a problem as the matrix becomes singular.

QR-Lohner algorithm

To solve the problem of inverting a singular matrix in the parallelepiped method Lohner proposed a modification of the above method consisting in just inverting the orthogonal matrix Q_1 of the QR factorisation of the matrix A_1.

As before, we put $\{u_1\} = m(u_1) + A_1[\hat{r_1}]$ then $A_1[\hat{r_1}] = DT[u_0][A_0][\hat{r_0}] + [z_1] - m(z_1)$ and $DT[u_0][A_0] = A_1[B_1]$. Consider the QR-factorisation of $[A_1]$, $[A_1] = Q_1[R_1]$. Hence,

$$[\hat{r_1}] = [R_1][B_1][\hat{r_0}] + Q_1^{-1}([z_1] - m(z_1)),$$

and $\{u_1\} = m(u_1) + Q_1[\hat{r_0}]$ and we proceed as in the parallelepiped method.

Lohner modification of the original algorithm seems to be an efficient way to deal with the ISVP. Note, however, that we compute again the term $DT[u_0]$ in an over estimated interval set containing $\{u_0\}$. The effect of wrapping appears not only in the term $DT[u_0]$, but it is also partially propagated in different ways in the term $[R_1][B_1][r_0]$.

KT method

Note that, when evaluating $DT(\{u_0\})$ on the interval set $[u_0]$, the wrapping effect can produce a big overestimation of the result. In order to solve this problem we propose the following modification of the QR-Lohner algorithm. We observe that any vector field of the form $\ddot{x} = f(x)$ is equivariant by matrices of the form

$$\begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix},$$

where M is such that $f(x)$ is equivariant by M (that is, $f(M(x)) = Mf(x)$). Moreover, as shown in the appendix B, the Kepler force and, in general, the N-body problem, is equivariant by any orthogonal matrix $Q \in \mathcal{SO}(3)$. We conclude that the Kepler vector field is equivariant by any matrix of the set

$$\mathcal{K} = \left\{ K \in \mathcal{M}_{6\times 6} \text{ s.t. } K = \begin{pmatrix} Q & 0 \\ 0 & Q \end{pmatrix}, Q \in \mathcal{SO}(3) \right\}.$$

This fact suggests the representation of the sets in the form $p+K[r]$ where $K \in \mathcal{K}$. The validated algorithm is similar to the QR algorithm but instead of computing the QR factorisation of matrix A_1 we compute a KT factorisation of the matrix, that is, a factorisation where $K \in \mathcal{K}$ and $T \in \mathcal{M}_{6\times 6}$. A possible choice of the matrix Q of K is the corresponding orthogonal matrix of the QR factorisation of an order three minor of A_1. For instance, if

$$A_1 = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \; A_{ij} \in \mathcal{M}_{3\times 3},$$

put $A_{11} = Q_1 R_1$ and choose $Q = Q_1 \in \mathcal{SO}(3)$. If T is expressed as a block matrix of the form

$$T = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix},$$

then, according to the choice of K above, $T_{ij} = Q^t A_{ij}$, $i, j = 1, 2$ (in particular, $T_{11} = R_1$).

Remark. It will be relevant to know which choice of K is suitable to carry on the computations. The problem with this method is that K is related with the “true” change of variables, given essentially by the first variational equations, but it does not coincide with the change neither with the orthogonal part of the change itself. It is interesting to study if another choice of matrices produces better results. For instance, if one considers matrices that respect only the equivariance of the force f, the method can be adapted just by changing the initial point at each step but the jet of derivatives necessary to compute the Taylor step remains invariant.

The important fact of the KT factorisation is that it allows us to make the change of variables $x = K\xi$ and integrate the equations in the new variables x, because by equivariance this is equivalent to a suitable integration in the ξ variables. In particular, the differential DT can be evaluated in the set $K^{-1}(p) + [r]$ without computing an interval set $[u_0]$ which exceeds the real set $\{u_0\}$. We explain in detail the evaluations in what follows.

In the ξ coordinates, a set $\{u_0\} = p_0 + K_0[r_0]$ is expressed as $\{u_0\} = K_0^{-1}(p_0) + [x_0] = [\hat{x}_0]$, with $[x_0] = [r_0]$. The system of equations has the same expression in the new coordinates due to the equivariance of the vector field with respect to $K \in \mathcal{K}$. So, we have to integrate the same system but with different initial condition (the initial condition now is given by $[\hat{x}_0]$ which is an interval with respect to the ξ coordinates). In particular, the jet of derivatives of the first variational equations is the same, so we can compute $DT[\hat{x}_0]$ directly. The method reads then,

$$T(\{u_0\}) \subset T(m(\hat{x}_0)) + DT[\hat{x}_0][x_0] + [z_1],$$

and, as before, we require the right hand of the last expression to be of the form $p_1 + K_1[x_1]$. Then, in the new coordinates x the set is described by $K_0 p_1 + K_0 K_1[x_1]$.

We note that all the methods explained can be generalised to higher order centred form like the one given by (4.6). In the next Chapter we provide examples of different computations carried out using each of these methods. However, other methods and generalisations can be found in the literature as we proceed to briefly explain.

4.1.3 $\mathbb{C}^r$-Lohner algorithms

Although our interest in this work is to propagate a box under an N-body flow, and this should be done by integration of the vector field, we want to stress that for other applications (for instance [KS07]) it is important to compute rigorously the solution of the variational equations.

It turns out that the direct rigorous integration by Lohner algorithm of the system of equations together with the variational equations is not an efficient way to integrate the problem.

The interval methods above described are slightly modified to compute the variational equations. Note that the variational equation is already used to get the term $DT[u_0]$, remaining only a rigorous bound of the error of this evaluation. This gives rise to the so-called C^1-Lohner methods (see [Zgl02]) which provide a rigorous integrator for the system of equations and variational equations with a few more effort.

The idea used can be generalised to higher derivatives giving rise to the C^r-Lohner algorithms (see [WZ08]) . It becomes necessary an examination of the errors using this type of integrators for the variational equations.

4.2 Taylor-based methods

In the numerical examples using validated methods we will see that the dependency on the initial conditions up to first, second or higher order plays a role in the computations allowing to reduce the wrapping effect. Also in the non-rigorous computations of Apophis (Section 3.3), it was observed that the box of random points is well-approximated using the first variational equations for quite a long time becoming necessary the second variational equations when trying to approximate the random box close to the 2029 approach of Apophis with the Earth.

Using validated integrators, as the boxes increase due to wrapping and dependency effects, it becomes necessary to use higher order approximation after a smaller interval of time than in the non-validated case. There is another type of methods which take care of this fact and compute the dependency of the final set with respect to the initial conditions up to higher order.

When computing the set $\{u_1\}$ at time $t = t_0 + h$ of the ISVP with initial condition $\{u_0\}$ the interval computations give rise to wrapping and dependency phenomena. The idea behind the Taylor-based models is to represent the set $\{u_1\}$ as a Taylor series with respect to the initial conditions. More precisely, they represent the set $\{u_0\}$ as

$$\{u_0\} = \hat{u}_0 + [\Delta u_0],$$

where $[\Delta u_0] \in \mathbb{I}^m$ is an interval box describing the uncertainty in the initial conditions. Then $(\delta_1, ..., \delta_m) \in \{\Delta u_0\}$ denotes any point in the uncertainty region. The idea is to represent the solution at time $t = t_0 + h$ as a Taylor series with respect to the set of variables $(t, \delta) = (t, \delta_1, ..., \delta_m)$. The Taylor series is then truncated at order n with respect to the full set of variables and the remainder is estimated in a suitable way. Mainly, the idea to estimate the errors is to compute the Taylor step using polynomials in a non-rigorous way and to obtain an interval I bounding the errors of computations. In the next step, the input is a polynomial plus the interval I. This is called a Taylor model which consists in the polynomial approximation p plus an interval I. The Taylor step produces a Taylor model of the function at time $t + h$.

To compute the Taylor series with respect to the set of variables (t, δ) a symbolic manipulator is used. The input parameter for the Taylor time stepper is a polynomial with respect to the δ variables and the output is again a polynomial with respect to the δ variables, which is the Taylor expansion of a set close to the set $\{u_1\}$ (they differ by terms that are added to the remainder in a suitable way). In the time stepper, the polynomials are considered with respect to the (t, δ) variables (see [NJN07]).

In order to compute a rigorous remainder an algebra on Taylor models is necessary. That is, at least we need to know how to compute the sum, product, inverse, powers, etc. of Taylor models. The algebraic manipulator allows to do these operations for the polynomial part of the Taylor model. On the other hand, for the interval remainder part it is necessary a careful bound of the Taylor remainder of the Taylor series with respect to the full set of parameters of the expansion. In this way in [Mak98] it is established how to do the computations with Taylor models. For instance, let (p_1, I_1) and (p_2, I_2) be two Taylor models up to order n of two functions. Then $(p_1 + p_2, I_1 + I_2)$ is a Taylor model up to order n for the sum of the functions, where $p_1 + p_2$ is computed symbolically and $I_1 + I_2$ is computed using interval arithmetic. Other algebraic operations are not so simple, but for each operation one can compute a suitable upper bound for the interval part of the Taylor models (see [Mak98]).

Among the advantages of this method, we note the ability to deal with sets that are not convex, mainly because at any step it is not necessary to embed the approximation of the solution in an interval set. However, the wrapping effect still plays a role: it is included in the remainder estimations which at each step is an interval containing all the errors of the computations. In this way, different strategies are also proposed to confront this problem. Mainly, the kind of strategies are the same as those explained in the interval methods, that is, a parallelepiped representation of the remainder set, QR representation when the matrix of the parallelepiped method becomes a singular matrix, or different kinds of representations according to the dynamics of the flow to be integrated rigorously. We want to emphasise that the preconditioning strategy, which consists in taking suitable coordinates to integrate the system, can be included in the Taylor-based method. This is accomplished by composition of the Taylor series of the change of variables with the Taylor series of the integration with respect to the new variables (see [MB05]).

In general, the Taylor-based methods produce better results than the first order Interval methods (see [NJN07], [HB03] for comparison of the results). From a theoretical point of view, the higher order Interval methods are equivalent to this polynomial approach. However, using this idea one can compute in an easy (but expensive) way the dependency with respect to the initial conditions up to higher order.

Remark.

1. In the Taylor time stepper, the jet with respect to the time t is considered mixed with the jet with respect to the space coordinates δ. It is not clear to us if considering separate jet expansions produce different results. Moreover, it is possible to perform the integration using different orders for different variables according to the precision required. However, if both jets are considered together, the Taylor model algebra used for computations gives upper bounds of the solution in a direct way. When dealing with the jets in a separate way, one should bound the time expansion using different techniques (for instance, via the computation of a rough enclosure for the solution or via analytical estimates).

2. A suitable rescale of variables and time to get similar rates of increase of the different variables could provide sharper bounds of the error of computations.

4.3 Subdivision strategy

As we have already mentioned, in the validated integration of an ODE the problem to deal with is mainly the wrapping effect and, more precisely, the wrapping produced in the evaluation of $DT(\{u\})$ (in the case of a first order interval method). In the case of Taylor-based methods, the wrapping is included in the remainder and in the final evaluation. As a general procedure to reduce the wrapping, one can consider the subdivision strategy which consists in dividing the interval in different intervals and evaluating each of them separately.

We consider two strategies of subdivision. A first procedure consists in subdividing the set in different intervals and integrating each of them separately. Note that this subdivision can be done at the beginning or when considered necessary along the computations. This reduces the problem of integrating a big domain into many problems of integrating not so big intervals.

On the other hand, for the Interval methods, another possibility is to divide the interval just when necessary for the evaluation of the variational part, which produces wrapping and then, to continue with the integration by considering the interval hull of the set of intervals obtained after the evaluation.

The major problem with subdivision strategy is that the computational time needed to obtain the result increases linearly with the number of subdivisions. On the other hand, we are dealing with six dimensional interval boxes and it is not clear which can be the optimal way to divide them. Nevertheless, we note that the process of evaluation of the different subintervals can be parallelised easily, allowing the computations to be done in a relatively short time.

An advantage of this subdivision procedure is that it can be implemented in any of the methods explained in this section, and in particular for all the Interval methods of first and higher order. In the next section, we will illustrate the effect of this subdivision in some examples.

Chapter 5

Validated methods: results

In this chapter, we compare the results obtained using some of the numerical methods explained in the last chapter. Before dealing with the concrete examples of the motion of an asteroid and the low-thrust transfer, we focus on the Kepler's problem which we shall use as a paradigmatic example to illustrate the advantages and inconveniences of the methods. We will see that, as was also observed in [HB03], the results obtained from the integration of the asteroid in the Kepler's problem and in the N-body code do not differ too much. We recall that the two body problem in the asteroid case is reduced to a centre vector field because the asteroid is assumed to be of negligible mass. On the other hand, the solution of the Kepler problem can be computed explicitly, though we do not use this information neither any other properties because we want to confront this problem in the same way as we can confront any other general problem.

5.1 Validated integration of Kepler's problem

In the examples of this section the initial condition described in Table 5.1 are used. This initial condition corresponds to the data of JD 2453979.5 (1 September 2006 00:00h) obtained from [GBO$^+$08], which also gives information about uncertainties.

Component	Initial data	Uncertainty
x_0	0.5166128258669076	$\pm 5 \times 10^{-8}$
y_0	0.6961955810635310	$\pm 5 \times 10^{-8}$
z_0	$-2.443608670809208 \times 10^{-2}$	$\pm 5 \times 10^{-8}$
v_x	$-1.295180180760195 \times 10^{-2}$	0
v_y	$1.388132695417834 \times 10^{-2}$	0
v_z	$-1.047646475022484 \times 10^{-3}$	0

Table 5.1: Initial condition and uncertainties for the Kepler's problem.

On the other hand, for some examples we have used uncertainties of 10^{-6} in position instead of 5×10^{-8}, the reason being to analyse the overestimations of the interval boxes with more evidence. We will specify each uncertainty for the intervals when convenient. The uncertainties are always considered in position because the uncertainties in velocity at the initial point can be translated in terms of the uncertainties in position in a similar manner that we have done to generate the random points in the non-validated integration.

We have implemented different Interval methods. The algorithms described in the last chapter provide rigorous integration schemes for the ISVP. In the implementation, all the computations are done using interval arithmetic in order to obtain rigorous set enclosures for the flow. These are the set which contain the numerical errors of the computations and hence validate the procedure.

We start by describing the results obtained using our codes in the case of Interval methods of first order. Table 5.2 specifies the time at which the computations stop due to the overestimation of the sets involved and to the impossibility of finding a rough enclosure for the validating step of the Taylor method. The results have been obtained using the parallelepiped method, the QR-Lohner one and the KT modification. To get these results it has been used a Taylor method of order 28 and fixed step size $h = 0.625$ day. It should be noted that to decrease the order and/or to change the step size (for instance dividing it by 10) produces exactly the same results.

Uncertainty	Parallelepiped	QR-Lohner	KT-equivariant
0	1612.5	1206.875	969.375
$\pm 5 \times 10^{-8}$	648.75	748.75	573.75
$\pm 10^{-6}$	268.75	493.75	473.75

Table 5.2: Maximum time of integration (in days) applying first order intervalar methods to the Kepler's problem. In all the computations, $h = 0.625$ day.

It seems that the integration using QR method gives better outcome when the initial interval box is not so small. However, in the parallelepiped method we have never detected problems with the inversion of the matrices. Hence it remains to clarify why the QR method is better.

Remark. We have already shown how an uncertainty in semi–major axis is reflected on the behaviour of the box and how the angle between two given vectors evolves in time. Recalling the results of Section 2.1.1, we find an heuristic explanation why there is no problem in the inversion of matrices in the parallelepiped method.

Before continue with our exploration of the interval methods, we would like to make some comments with respect to the step size h. The choice of $h = 0.625$ day is motivated by the fact that this number is representable in an exact form by the binary arithmetic that the computer uses. In this way, there is no error in time when integrating.

Nevertheless, although from a formal point of view this choice should be done, it has not been observed any difference in the computations when changing the step size.

On the other hand, we note that while in the non-validated integration the step size considered is around 30 days (except, of course, in the close approach to the Earth) in the validated integration we are restricted to the use of a step size less than one day. The main restriction for the usage of large step size is due to the computation of the rough enclosure which is necessary for the validation of the process. In Appendix B it is described an alternative to the rough enclosure procedure which allows much larger step sizes. The proposed procedure has been implemented and the results are described in Section 5.2.

To continue with the analysis of the results obtained using the first order Interval Taylor methods, we examine which one of the considered methods provide more accurate interval boxes. Figure 5.1 on the left shows the distance between the box obtained by integration of 500 random conditions in the initial data box by the non-rigorous Taylor method and the interval box obtained by validated integration using the parallelepiped method. The points of the random box are translated inside the interval box $[\hat{r}_k]$ via the inverse of the change of coordinates A_k. In these coordinates, the distance is measured. In the figure, on the y-axis it is represented the decimal logarithm of the distance in each component and on the x-axis the number of iterations (each unit in the x-axis means 6.25 days of integration).

We note that formula (4.5) implies that no one of the coordinates is able to decrease although the real dynamics makes the box to decrease in some directions (indeed, the differential of the flow is symplectic and the volume of the boxes is preserved in phase space). To solve this, we have modified the method by factorising the matrix A_1 in the form $A_1 = \tilde{A}_1 D$, where D is a diagonal matrix which considers the expansion and contraction in each one of the directions. The corresponding results are illustrated in figure 5.1 on the right. We note that there is no advantage using this modification. In all these considerations, the initial uncertainty in position is of 10^{-6}.

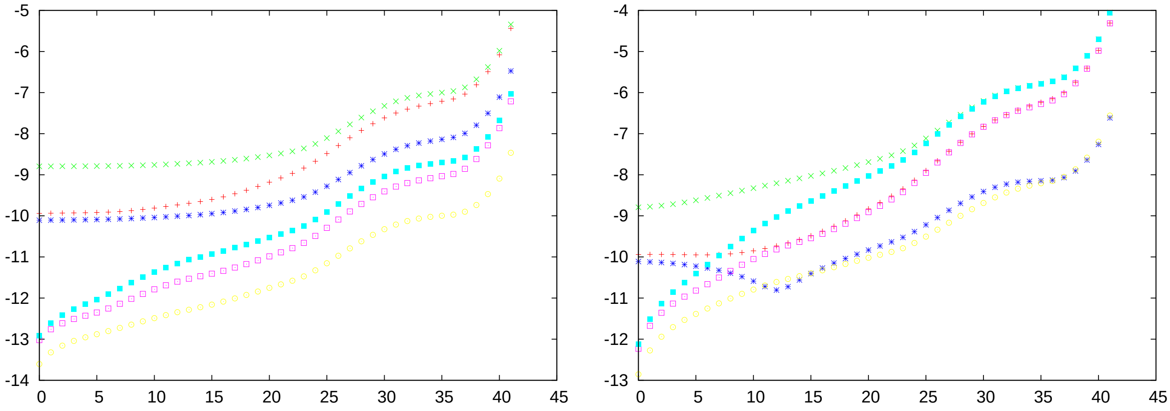

Figure 5.1: Overestimation of the interval boxes with respect to the random boxes in the parallelepiped method (left) and the modified parallelepiped method (right), see text for details. Each colour corresponds to the difference of the measure in each of the components: x corresponds to red colour, y to green, z to dark blue, v_x to magenta, v_y to blue and v_z to yellow colour.

Figure 5.2 shows the same box overestimations as in the Figure 5.1, but for the QR-Lohner algorithm and the KT-modification.

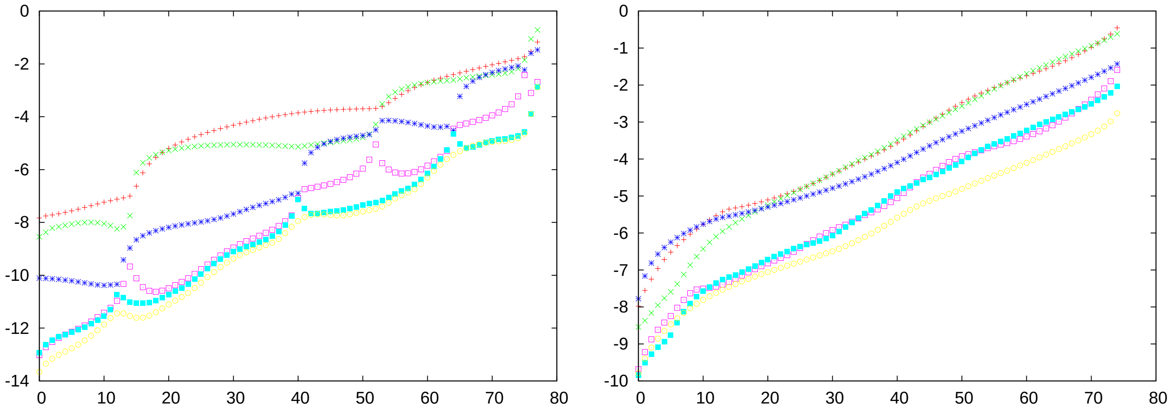

Figure 5.2: Overestimation of the interval boxes with respect to the random boxes using the QR-Lohner method (left) and the KT modification (right), see text for details. The pattern of colours used is the same as in figure 5.1.

Observing the Figures 5.1 and 5.2 we can conclude that the parallelepiped method produces better results, although the other methods allow to make computations for longer time. Note that in the QR and KT methods at each step the sets are included in a representation which does not follow so close the true shape of the interval. This happens in particular in the KT

case where the matrix K is not necessary close to the linear change of variables needed for the representation.

To better understand how the computed sets differ from the random points box, Figures 5.3 and 5.4 represent the interval boxes obtained using parallelepiped and KT methods as well as the random box for $t = k\ 6.25$ with $k = 0, 5, 10$ and 15. The computed interval box is just the section corresponding to zero initial uncertainty in the velocity components. We remark that, at the beginning, this section contains the box of random points, but that this does not necessarily take place when the uncertainty in the velocity components increases.

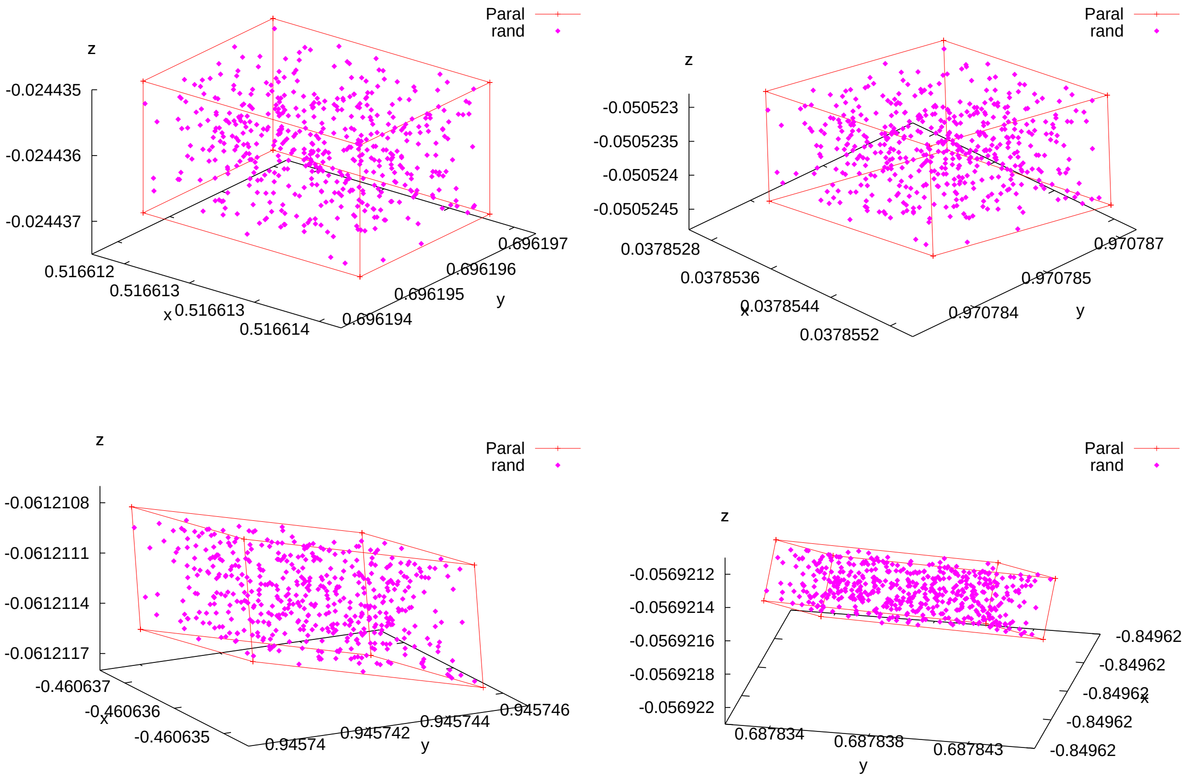

Figure 5.3: For $t = k\ 6.25$ with $k = 0, 5, 10,$ and 15 we represent the boxes obtained using the parallelepiped method and the position of the random initial points.

In a more precise way, the random box should be contained in the projection of the six dimensional interval box computed by the method. Note that the projection in a plane of a six dimensional box is, generically, a dodecagon. The corresponding projection containing the position of the points of the initial random box is depicted in Figure 5.5.

In all of the methods, we observe the same type of behaviour. The interval boxes predicted by the validated algorithm are quite precise during the first steps but then start to grow. This implies that the first order variational matrix in the set, that is $DT(\{u_0\})$ and the centre of this variational matrix $m(DT(\{u_0\}))$ cannot longer be close enough. In other words, it seems necessary at least a second order variational approximation in the set $\{u_0\}$. This makes necessary the implementation and analysis of the second order Interval Taylor methods.

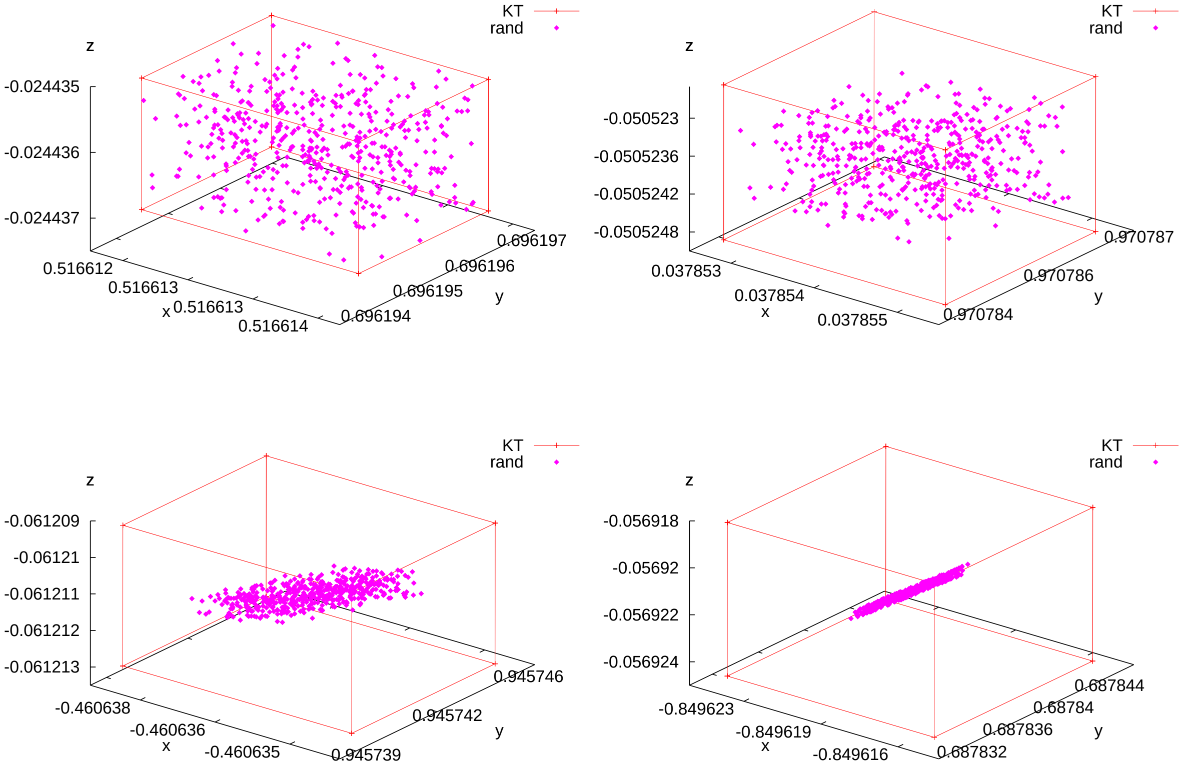

Figure 5.4: For $t = k\ 6.25$ with $k = 0, 5, 10,$ and 15 we represent the boxes obtained using the KT method and the position of the random initial points.

Parallelepiped	QR-Lohner	KT-equivariant
893.75	681.25	481.25

Table 5.3: Maximum time of integration (in days) obtained using second order interval methods. In all the computations, $h = 0.625$ day and the initial uncertainty in position is $\pm 10^{-6}$.

5.1.1 Second order interval methods

As explained in the sketch of validated methods of the Chapter before, all the Interval methods can be generalised to second order Interval methods. We have implemented the second order methods for the three types of algorithms. The more satisfactory results have been obtained with the parallelepiped method. All the results are described in Table 5.3.

Now, if we apply the parallelepiped second order method to the Kepler's problem with the real initial uncertainty of $\pm 5 \times 10^{-8}$ in position, we obtain the results of Table 5.4.

To obtain these results it was necessary to use $h = 0.0625$ day. Using $h = 0.625$ day the program breaks down because at some step (depending on the method) it is not possible to find rough enclosure. Once more it becomes clear the need to suppress the rough enclosure step (see B where an idea is provided to suppress this step of validation and to justify the use of larger step size).

Remark. At this point we can comment on the fast increase of the size of the boxes near the

1st. order	636.875 days
2nd. order	1575. days

Table 5.4: Maximum time of integration (in days) obtained using first and second order parallelepiped interval methods. In all the computations, $h = 0.0625$ day and the initial uncertainty in position is $\pm 5 \times 10^{-8}$.

No Div.	268
Div. 1 ($k = 2^6$)	330

Table 5.5: Comparison between the maximum interval of time obtained without applying subdivision and applying the first strategy subdivision. Unit days.

end of the admissible computations. Let ℓ be a typical size of the box at one step. At the next step it is of the order $\ell(1 + \alpha\ell^p)$ if a method of order p is used. Here α is a constant, probably large, which accounts for bounds of higher order derivatives, dependency, etc. We can set up the recurrence $\ell_{k+1} = \ell_k + \alpha\ell^{p+1}$ which, for small ℓ_k and smaller α can be approximated by the solution of $\frac{d\ell}{dk} = \alpha\ell^{p+1}$, which gives $\ell_k = (\ell_0^{-p} - p\alpha k)^{-1/p}$, with a singularity for $k = \frac{1}{p\alpha}\ell_o^{-p}$.

5.1.2 Subdivision strategy

We have implemented the subdivision process for the parallelepiped method of order one. Tables 5.5 and 5.6 contain the details of the computations. The initial uncertainty in position is 10^{-6} and the step size is $h = 0.625$ day. As explained before, we have considered two different subdivision strategies:

1. Div. 1: We divide the initial interval box into k subintervals and then we propagate each one of them separately.

2. Div. 2: We divide the interval box into k subintervals just when we compute the variational matrix, $DT(\{u_0\})$. After dividing the interval box, we can proceed in two different ways:

   (i) We compute the variational matrix for each of the intervals, obtaining $DT\{u_0^i\}$ for $i = 0, ..., k$, and we take the union of the interval components of the matrices to produce the final $DT\{u_0\}$. Then, we continue with the validated algorithm using the variational matrix $DT\{u_0\}$.

   (ii) For each of the k interval matrices we compute the product $DT\{u_0^i\}A_0$, for $i = 0, ..., k$, obtaining k matrices and then consider the union of each interval components to construct the matrix $DT\{u_0\}A_0$.

As we can see, as the number of subdivisions increases we are able to integrate more time. The main drawback is the computational time. In particular, we observe that the strategy 2 (*ii*) is better than the 2 (*i*) one because the former is able to reduce dependency and wrapping problems. Figure 5.6 illustrates the boxes obtained using the first strategy. As before, the plot contains a three dimensional representation and just the points with no uncertainty in velocity should be contained in the boxes.

Alternative approaches, taking directly an interval which contains $A[u]$, subdividing in intervals and taking only the ones which intersect $A[u]$ can also be considered, but have not been implemented yet.

Div. 2	$k = 2^6$	$k = 3^6$	$k = 4^6$
(i)	313	338	365.625
(ii)	441.25	446.875	448.75

Table 5.6: Comparison between the subdivisions methods (i) and (ii) of the second subdivision strategy in terms of the maximum integrated time in days. The boxes are subdivided the same number of times in each direction. So $k = 2^6$ means that two subdivisions are made in each direction, $k = 3^6$ means three subdivisions in any of the six directions, and $k = 4^6$ four subdivisions.

5.2 Validated alternative to the rough enclosure

As we have already commented, the computation of a rough enclosure in any validated algorithm strongly restricts the time step size. In particular, for the Kepler problem in the Apophis case we have to use step sizes less than one day although the period is close to one year. To solve this inconvenience, in Appendix B we describe a way to bound the remainder without using the rough enclosure set in a validated way. This strategy also provides a way to choose an appropriate variable step size which is much larger than the one provided by the rough enclosure method.

We recall that the Kepler equations, after normalising $\mu = 1$, can be written as

$$\ddot{x} = f_x = -xr^{-3}, \quad \ddot{y} = f_y = -yr^{-3}, \quad \ddot{z} = f_z = -zr^{-3}.$$

If the instantaneous distance to the body is assumed to be one, then the remainder of the Taylor expansion of the Kepler equation can be bounded by

$$c^{-1}\left(\frac{h}{t_f}\right)^{n+1} \frac{1}{1-\frac{h}{t_f}}, \tag{5.1}$$

where $t_f = \frac{1}{vc-\sqrt{c}}$ is the convergence radius of the solution with respect to the time variable, $c = \sqrt{2}+1$ is the rate of increase of the coefficients of f_x, f_y and f_z, and h is the step size (see Appendix B for details).

We will bound the remainder by the roundoff error representation of the arithmetics (which is equal to $\varepsilon = 2^{-52}$) and find for a given order n the appropriate step size h. That is, we look for h such that (5.1) is less than ε. We solve the equality

$$c^{-1}(\xi)^{n+1}\frac{1}{1-\xi} = \varepsilon,$$

or equivalently,

$$\xi^{n+1} + (\xi - 1)\varepsilon c = 0,$$

where $\xi = \frac{h}{t_f}$ is solved iterating a Newton method using as initial guess $\xi = 1$. It is easy to see that when n tends to ∞, ξ goes to 1 (i.e. $h \to t_f$).

Then we define the new step size $h = \text{fact}\ \xi t_f$, where fact is a constant. For instance, we have chosen fact $= 0.95$. Before using the new step size we must check that the inequality

$$c^{-1}(\xi)^{n+1}\frac{1}{1-\xi} < \varepsilon,$$

is rigorously satisfied (using interval arithmetics). If it is not satisfied we can reduce fact .

Figure 5.7 shows the step size h predicted by the method explained above as a function of the order n. We see that it tends to a value close to the convergence radius t_f, precisely, as we use fact $= 0.95$ to compute h, it converges to $0.95t_f$.

1st. order	632.5 days
2nd. order	1529.375 days

Table 5.7: Maximum integration time obtained for Apophis in the $(N+1)$-problem using first and second order parallelepiped validated methods.

To apply this procedure to the general Kepler problem is necessary to rescale distance and time at each step of integration. In the Apophis case, with initial uncertainty in position equal to 10^{-6} and using order 28 in the Taylor expansion, the evolution of the step size along the integration can be seen in Figure 5.8.

We note that the time of integration is almost the same as the one achieved using rough enclosure. However, the number of iterates of integration is 64 instead of 429 iterates using $h = 0.625$ and the rough enclosure procedure.

5.3 Validated integration of the Apophis orbit

This section describes the results obtained in the validated propagation of the orbit of Apophis.

We take as model the restricted $(N+1)$-body problem, as initial box the one described in Table 5.1 and as step size $h = 0.0625$ days. We have set uncertainty in position only in the Apophis coordinates, not in the planets' ones. Indeed, at each integration step we consider the position and velocity of the planets as point intervals. This is because we need intervals to computes the jet of derivatives associated with Apophis, but we read the coordinates of the major bodies from the JPL ephemerides.

As mentioned before, the results offered by the Kepler model and those obtained with the $(N+1)$-body problem are very similar. Comparing Tables 5.7 and 5.4, we can appreciate how the easiest model represents a good approximation for the full one.

5.4 Validated integration of the low-thrust problem

As explained previously, for the low-thrust mission we take as nominal orbits two trajectories which require a manoeuvre in the course of the transfer. With a validated method we will likely perform the manoeuvre when the box has become considerably large but this will not apply to an actual mission. Rather, we can think to consider the trajectory as composed by different branches and to treat each of them separately. In the real framework, we are given with ground and on board instruments as reliable as to control and correct the orbit of the probe, as soon as it deviates from the nominal one. To fix ideas, we assume that orbit determination is available every day. In other words, a validated analysis of the low-thrust problem can make sense if we take into account short intervals of time.

The first case analysed takes as initial condition a validated box centred at the point given in Tables 3.10 and 3.11 setting an uncertainty in space of 10^{-12} AU. We will put uncertainty not only in position but also in the thrust magnitude, in order to consider the error relative to the engine's power. In particular, we have considered three values of uncertainty for the thrust: 0, 5×10^{-9} and 5×10^{-8} AU/day^2, respectively.

We apply the parallelepiped method of first and second order without subdivision to propagate the box forwards in time with $F_T < 0$ and $V_c = V_{Moon}$ and also backwards in time with $F_T > 0$ and $V_c = V_{Earth}$, that is, when we are braking towards the Moon or accelerating from the Earth, respectively. We have used different values for the integration step to see how the result might change according to it. In Tables 5.8 and 5.9, we show the maximum interval of

thrust error	first order	second order
0	6.7875	6.8375
5×10^{-9}	4.41875	4.5625
5×10^{-8}	3.63125	3.7875

Table 5.8: Maximum interval of time obtained until the first and the second order parallelepiped method break down. The initial box considered is centred at the point given in Tables 3.10 and 3.11 with an uncertainty in space of 10^{-12} AU. The integration has been performed forwards in time with $F_T < 0$ and $V_c = V_{Moon}$, which corresponds to the trajectory's leg going from L_1 to the Moon. The integration step is $h = 0.00625$ days. The first column refers to the error assumed for the thrust. Units AU/day^2 and days.

thrust error	first order	second order
0	9.6125	9.8125
5×10^{-9}	4.6875	4.76875
5×10^{-8}	3.4875	3.575

Table 5.9: Maximum interval of time obtained until the first and the second order parallelepiped method break down. The initial box considered is centred at the point given in Tables 3.10 and 3.11 with an uncertainty in space of 10^{-12} AU. The integration has been performed backwards in time with $F_T > 0$ and $V_c = V_{Earth}$, which corresponds to the trajectory's leg going from L_1 to the Earth. The integration step is $h = 0.00625$ days. The first column refers to the error assumed for the thrust. Units AU/day^2 and days.

time obtained until the given method breaks down using $h = 0.00625$ days. In Tables 5.10 and 5.11, we display the results referring to $h = 0.0078125$ days. Comparing a first order validated method with a second order one, we note that the greater computational effort needed in the second case is not worth, since it does not result in a significant improvement. The same can be observed comparing different integration steps. Apart from those showed, we have tried with $h = 0.0625$ and $h = 0.000244140625$ days.

A similar effort has also been devoted to test QR–Lohner methods of first and second order on the low–thrust transfers. Though increasing the order does not provide any meaningful contribution, in this case we obtain much better results in the L_1–Earth's leg of trajectory comparing with those offered by the parallelepiped method. In Figure 5.9, we show the orbit integrated with the two methods, considering null the error for the thrust.

With respect to the other nominal trajectory, departing from a box centred at the point given in Tables 3.8 and 3.9 and setting an uncertainty in space of 10^{-12} AU, with any of the above methods we are able to launch the validated integration only with $h <= 10^{-6}$ days, which is not an useful parameter. This difficulty arises because of the short period associated with the initial condition considered and because of the small distance with respect to the Earth.

thrust error	first order	second order
0	6.9140625	6.875
5×10^{-9}	4.3984375	4.5625
5×10^{-8}	3.6328125	3.7890625

Table 5.10: Maximum interval of time obtained until the first and the second order parallelepiped method break down. The initial box considered is centred at the point given in Tables 3.10 and 3.11 with an uncertainty in space of 10^{-12} AU. The integration has been performed forwards in time with $F_T < 0$ and $V_c = V_{Moon}$, which corresponds to the trajectory's leg going from L_1 to the Moon. The integration step is $h = 0.0078125$ days. The first column refers to the error assumed for the thrust. Units AU/day^2 and days.

thrust error	first order	second order
0	8.8203125	9.8828125
5×10^{-9}	4.4765625	4.7734375
5×10^{-8}	3.4921875	3.578125

Table 5.11: Maximum interval of time obtained until the first and the second order parallelepiped method break down. The initial box considered is centred at the point given in Tables 3.10 and 3.11 with an uncertainty in space of 10^{-12} AU. The integration has been performed backwards in time with $F_T > 0$ and $V_c = V_{Earth}$, which corresponds to the trajectory's leg going from L_1 to the Moon. The integration step is $h = 0.0078125$ days. The first column refers to the error assumed for the thrust. Units AU/day^2 and days.

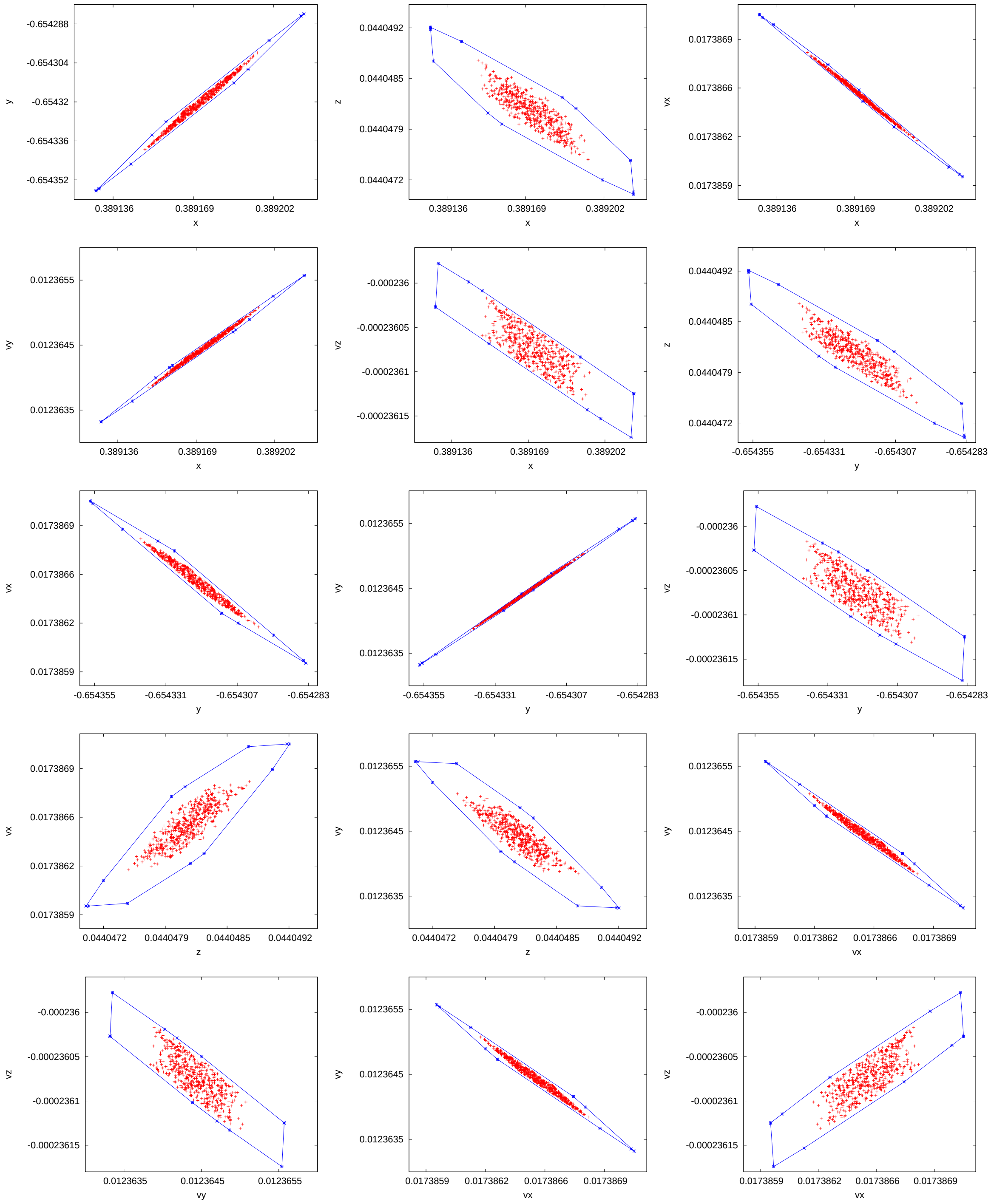

Figure 5.5: Two dimensional projection of the 6D interval box computed using the parallelepiped method at $t = 250$ days. From left to right and from top to bottom the corresponding plane of projection is xy, xz, xv_x, xv_y, xv_z, yz, yv_x, yv_y, yv_z, zv_x, zv_y, zv_z, v_xv_y, v_xv_z, and v_yv_z.

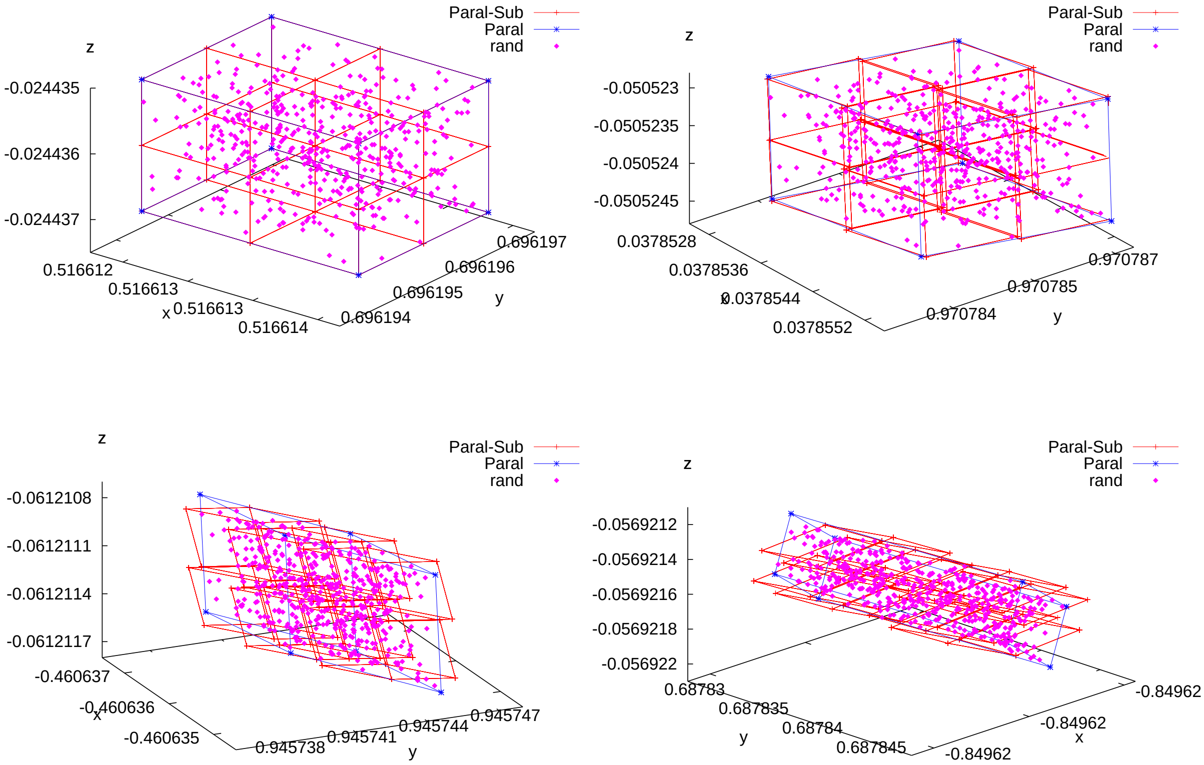

Figure 5.6: Propagation of the 64 boxes contained in the initial box with uncertainties 10^{-6} in position. From left to right and top to bottom $t = 0,\ 31.25,\ 62.5,\ 93.75$ days.

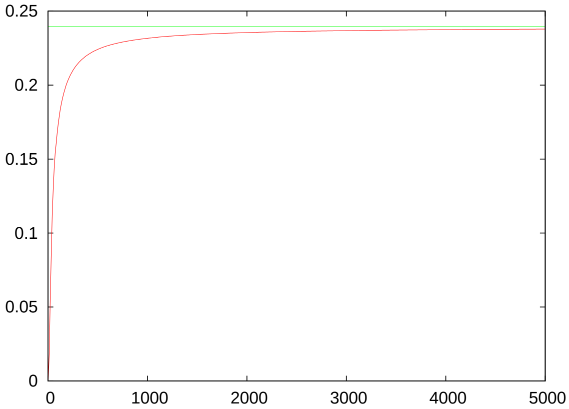

Figure 5.7: Step size h (y-axis) vs. order n (x-axis) for the normalised Kepler problem. Green line is the value of $0.95t_f$.

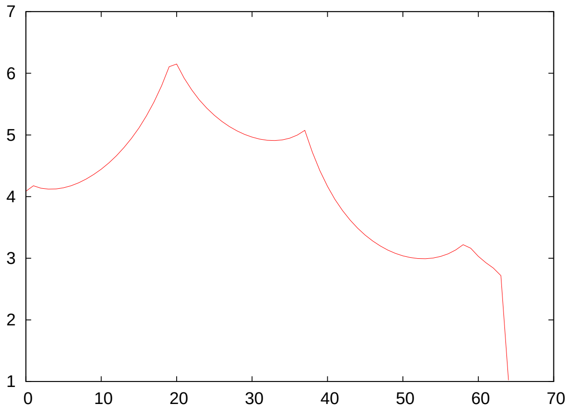

Figure 5.8: Evolution of the step size along the integration, for the alternative method proposed in Section 5.2. Step size h in days (y-axis) vs. iterate of validated integration (x-axis) for the Apophis Kepler problem.

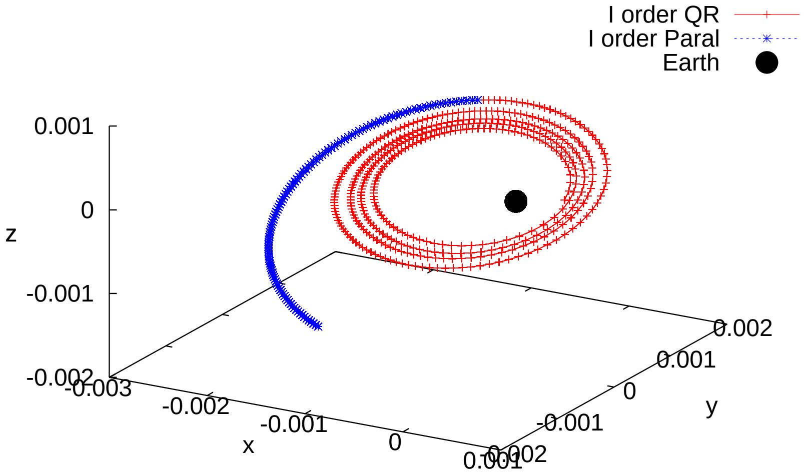

Figure 5.9: In red, the part of trajectory integrated with a first order QR–Lohner method, about 43 days. In blue, with a first order parallelepiped one, about 9 days. In black, the Earth. The initial box considered is the one centred at the point given in Tables 3.10 and 3.11 and setting an uncertainty in space of 10^{-12} AU and null uncertainty for the thrust. Unit AU.

5.5 Summary

Here we summarise the results obtained in this chapter. We have developed three different interval based methods:

- Parallelepiped,
- QR - Lohner,
- KT - equivariant.

First, we have tested them for the Kepler problem using a first and second order approach, following the ideas of the original Moore's algorithm. In Table 5.12 we can see these results for all these methods taking an initial uncertainty of $\pm 10^{-6}$ and an integration step size $h = 0.625$.

	Parallelepiped	QR - Lohner	KT - equivariant
1st order	268.75	493.75	473.75
2nd order	893.75	681.25	481.25

Table 5.12: Comparison of the maximum time of integration (in days) obtained for the 3 different methods using 1st and 2nd order approaches. In all the computations we have taken a step size $h = 0.625$ days and an initial uncertainty of $\pm 10^{-6}$

As the step size provided by the classical rough enclosure method is very small, we have developed a new strategy (see Section 5.2), adapted for the Kepler problem, that can be generalized to N-body problems). This allows to use step sizes about 7 times larger (see Figure 5.8).

We have also developed and tested three different subdivision strategies (see Section 5.1.2 for more details on the differences) and we have tested them using the parallelepiped method. In Table 5.13 we can see the results of applying the different subdivision strategies with the first and second order methods. In Table 5.6 we can see the results of considering the two subdivision strategies and applying them for a different number of subdivisions.

1st order	2nd order	Subdiv 1	Subdiv 2i	Subdiv 2ii
268.75	893.75	340	313	441.25

Table 5.13: Comparison of the maximum time of integration (in days) obtained for the parallelepiped method using 1st and 2nd order approaches and different subdivision strategies. In all the computations a step size $h = 0.625$ days and an initial uncertainty of $\pm 10^{-6}$ is used.

According to the previous results and the accuracy of the computed interval boxes it seems that the parallelepiped method is the best choice. We have applied this method, using the first and second order approaches, to our two applications: the motion of a NEO, taking the particular case of Apophis, and the motion of a low-thrust probe in the Earth-Moon system.

Table 5.7 shows the results for the Apophis case, taking and integration step size $h = 0.0625$. We can see that, in this case, taking the second order parallelepiped method is much better. We notice that the results are very similar to the results obtained for the Kepler problem.

In Tables 5.8, 5.9, 5.10 and 5.11 we can see the results for the motion of a low-thrust probe. In this case we can see that the gain obtained when we consider second order methods is almost negligible.

Chapter 6

Conclusions and future work

6.1 Conclusions

After looking at the topics presented in this report and to the available literature on Self Validated Integrators, the following conclusions have been reached:

1) For non-validated integrations, the Taylor method has been widely tested. In the case of non-stiff equations, for analytic vector fields or, at least, for sufficiently regular ones, it seems to be a good choice.

   Main points are: 1. Recurrent computation of higher order coefficients by automatic differentiation; 2. Truncation error negligible in front of round off errors; 3. Suitable order for optimal performance, and 4. Optimal step size depending only of the local radius of convergence of the expansion and, in particular, independent of the number of digits used in the computations.

   The propagation of the round off errors in the first integrals displays the characteristics of a random walk. In some sense, one cannot do better. Furthermore, for given initial conditions and final time one easily obtains that the cost is proportional to d^4 when working with an arithmetic with d digits.

   It also allows to implement the integration of variational equations. This, in turn, gives a way to propagate a box of initial data.

2) When dealing with intervals the key problems to face are dependency and wrapping. Their influence in the integration of ODE is crucial, specially for relatively long time intervals.

   It is important to note that we can consider two different kinds of problems. First one appears if our purpose is to give a Computer Assisted Proof of some fact like existence of a periodic solution or a transversal homoclinic solution. Second one concerns with the propagation of a given initial set of data.

   In the first case it is possible to use a large number of digits. This allows to "delay" the apparition of problems due to dependency and wrapping. A good and efficient implementation of interval arithmetic is essential to speed up the computations.

   In the second case, if the initial box is relatively large, then the proper dynamics can largely increase the size of the box. This occurs, typically, at a linear rate, for integrable Hamiltonian systems, like Kepler's problem. The behaviour is similar for an asteroid, except during the periods of close approaches. Then, as in the general case of chaotic systems, the rate of increase is exponential.

   In this last case the influence of the errors introduced by the arithmetic is almost negligible. The bad problem is that all estimates must be done in large boxes. As mentioned at the end of Section 5.1.1, even in a simple model describing the rate of increase of the size of the boxes, the procedures lead to a singularity in the size of the boxes.

3) Our efforts have been devoted to the implementation of algorithms based of Lohner algorithm and variants. We have found three, non independent, main sources of difficulties. We describe them shortly and how we have tried to solve them.

    a) First one concerns with rough enclosure. Starting with a current box at some moment t^* of the integration one should find another box which contains the image of that box for all t in the interval $[t^*, t^* + h]$, where h is the current time step. This is done to obtain existence of the solution and to bound the errors of the Taylor method used to do one time step. The procedure is iterative and the value of h has to be very small, of the order of the inverse of the Lipschitz constant of the vector field, to have convergence. This constant can increase in a non-admissible way when the size of the boxes increase far beyond the true size of the propagation of the starting box, due to dependency and wrapping.

       The fact that the rough enclosure is small is not critical concerning the final result about the size of the boxes at a given final time, but it affects, definitely, the computing time.

       An approach which largely improves the computations skips the computation of rough enclosure and replaces it by analytical estimates, based on the domain of analyticity of the vector field. Using tools from analysis we have been able to produce lower estimates of the radius of convergence and upper estimates of the remainder. Appendix B contains a sketch of the method for Kepler's problem. It can be extended to the perturbations of other bodies and to general analytic vector fields.

    b) Next difficulty is dependency. At every operation one takes the worst case among all the possible choices of elements contained in the boxes involved in the operation. But, in general, they are not independent because they can come from different real number computations done with the same elements. To compute $f(I)$ for all elements x in a set I is an almost impossible task if the computation of f involves many individual operations and we want to produce sharp estimates.

       A mild remedy we have tried is to change the interval arithmetic in the elementary case of computing squares of real intervals. But the improvement is minor. A better approach is based on subdivision, to be commented in next item.

    c) Last problem is wrapping. Every time an operation with sets is done, we must put the result inside a set of a given class, using some suitable representation. To mimic the behaviour of the true dynamics a convenient method consists in representing the sets as a central point plus a matrix applied to a product of intervals in $\mathbb{R}$. This is how boxes propagate under the first variational equations. The size of the boxes has to be increased to account for the remainder and errors in the operations.

       An important difficulty is that the differential matrix, even keeping the symplectic character in the integration of Hamiltonian systems, can have a strong increase in the condition number. Some representations like QR or similar decompositions can improve the results in a moderate way.

       In the case of Kepler problem (integrated without taking into account any special property, like passing to action angle variables) both dependency and wrapping produce an increase of the size of the box across the orbit. This implies changes in the energy, hence in the frequency and, therefore, in the size along the orbit.

       A suitable method to cope with these difficulties is subdivision. This can be done at two different places:

A. On the initial box of data.

B. When an initial box is propagated under the algorithm, instead of putting the image of the box inside a product of intervals, it can be covered by small boxes of the same kind, with a minimal covering.

Anyway, subdivision has the drawback of a strong increase in the computing time. This is specially dramatic when the dimension of the phase space is high. Dimension 6 gives not so many chances to subdivide in small elements with a reasonable computing time.

4) As a result of the experiments carried out inside this project we conclude that Lohner-like base methods are suitable when the following conditions are satisfied:

A. The size of the initial set of data is very small.

B. The time interval during which the system has to be integrated is moderate.

C. The rate of increase of the size of the boxes is moderate.

Even if these condition hold, it is quite convenient to try to avoid the computation of rough enclosures and replace them by analytical estimates. This can decrease in a significative way the computing time.

5) Another possibility, yet to be explored, is the use of Taylor based methods, not only for the time integration but also for the propagation of an initial set of points. That is, at every time step one should produce a Taylor expansion with respect to the variation of the initial conditions plus a (hopefully) small box containing the propagated effect of remainders (in the time step and in the phase space variables) plus the effect of the errors in the operations.

From a mathematical point of view what is done is a propagation of the solution as a truncated jet (a polynomial) representation in terms of the initial conditions, with control on all errors.

This method allows to decrease most of the problems due to dependency and wrapping. The price to be payed is the computational cost. In some sense it can be considered as a Lohner-like method of high order in which boxes are kept to a minimal size. The implementation also differ. While $\mathcal{C}^k$ Lohner methods integrate variational equations, the propagation of the jet in Taylor based methods is done symbolically, using routines for all the required operations as polynomials in several variables.

Anyway, even these methods require subdivision when the size of the propagated box is too large and the intervals bounding the remainder exceed a given size.

6) An important point is the optimality of the algorithms used, not only considering sharpness of the results, but also computing time. Preliminary attempts are included in Appendices A and B.

The fact that both time integration and dependence with respect to initial conditions are obtained by Taylor methods (or, equivalently, integration of variational equations) raises the problem of optimal orders with respect to each one of the variables, both time step and phase space variables. If the sizes of an initial box are quite different in different directions the orders of the expansion must adapt to that fact.

Careful a priori operation counts, which strongly depend on the algorithms used, allow to discuss the optimal strategy concerning subdivision. This also gives the optimal step size to be used to achieve optimal performance keeping an admissible bound on errors.

6.2 Future work

The study of the literature on the topic, the experience gained in this project, the difficulties encountered during its realisation and the analysis we have done of the problem suggest several lines of future work. We consider that they are quite promising.

i) Definitely gain also experience by implementing Taylor based methods. The dependence and wrapping effects seem to influence too strongly Lohner-like algorithms for the transport of big boxes during long time intervals.

ii) Items 4) and 5) in the Conclusions give a first indication about when it is suitable to use one of the two main approaches to deal with a given problem. After i) above, explicit comparisons of the relative performance should be done in a battery of problems.

iii) Analytical methods to obtain estimates for radius of convergence in the time integration, in bounding the remainder and in the domain of validity and remainder in the phase space variables should provide an alternative to other estimates. This requires location of singularities of the vector field in the complex phase space, Cauchy estimates for bound of the rates of increase of the coefficients in a polydisc, etc. Even for entire vector fields (e.g., a polynomial vectorfield) the solutions display generically singularities, perhaps for some complex time, which give the local radius of convergence.

iv) A delicate point is the optimality of the expansions used concerning efficiency. The vector field can be the sum of different contributions, like in the asteroid case. The main effect is the solar attraction and the effect of the other bodies is a perturbation, except in close approaches to a planet or moon, where the roles exchange.

   Taylor series for time integration can be computed to a different order for each one of the contributions, taking into account, of course, the couplings between the different terms (the so-called indirect effects in Celestial Mechanics). Optimality can then require s subtle strategy. We consider worth to explore this.

   A similar thing appears with the suitable orders to be used in the expansions in the phase space variables, with respect to each one of the variables.

v) Final topic of future work can be the inclusion of random terms in the vector field, such as uncertainties in some forces, errors in the execution of manoeuvres, etc. This is easily done in Lohner-like algorithms, while it requires the inclusion of additional variables in Taylor based methods.

   Uncertainties and errors in manoeuvres, as mentioned, have typically a statistical distribution, according to a given probability law. In the case of compact support this can be approximated by a distribution of probabilities in a finite set of small boxes. This is also true, for instance, in the propagation of a set of initial conditions for an asteroid, where orbit determination provides a statistical distribution of errors.

   Rigorous methods should be able to deliver a (discrete) approximation of the propagated distribution of probability. This is really important but still seems to be far in the future and will require a strong use of parallelism. But one should keep the point in mind.

6.3 Acknowledgements

We thank Gerard Gómez, Tomasz Kapela, María Eugenia Sansaturio, Daniel Wilczak and Piotr Zgliczynski for technical help. Computer facilities were provided by the Spanish Ministery of Education and Generalitat de Catalunya.

Bibliography

[AH83] G. Alefeld and J. Herzberger. *Introduction to interval computations.* Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1983. Translated from the German by Jon Rokne.

[BBCG96] M. Berz, C. Bischof, G.F. Corliss, and A. Griewank, editors. *Computational Differentiation: Techniques, Applications, and Tools.* SIAM, Philadelphia, Penn., 1996.

[BCCG92] C.H. Bischof, A. Carle, G.F. Corliss, and A. Griewank. ADIFOR: Automatic differentiation in a source translation environment. In Paul S. Wang, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 294–302, New York, 1992. ACM Press.

[BKSF59] L.M. Beda, L.N. Korolev, N.V. Sukkikh, and T.S. Frolova. Programs for automatic differentiation for the machine BESM. Technical Report, Institute for Precise Mechanics and Computation Techniques, Academy of Science, Moscow, USSR, 1959. (In Russian).

[BMH01] M. Berz, K. Makino, and J. Hoefkens. Verified integration of dynamics in the solar system. *Nonlinear Anal.*, 47(1):179–190, 2001.

[Bro71] R. Broucke. Solution of the N-Body Problem with recurrent power series. *Celestial Mech.*, 4(1):110–115, 1971.

[BWZ70] D. Barton, I.M. Willers, and R.V.M. Zahar. The automatic solution of ordinary differential equations by the method of Taylor series. *Computer J.*, 14(3):243–248, 1970.

[CC82] G.F. Corliss and Y.F. Chang. Solving ordinary differential equations using Taylor series. *ACM Trans. Math. Software*, 8(2):114–144, 1982.

[CC94] Y.F. Chang and G.F. Corliss. ATOMFT: Solving ODEs and DAEs using Taylor series. *Computers and Mathematics with Applications*, 28:209–233, 1994.

[Cor95] G.F. Corliss. Guaranteed error bounds for ordinary differential equations. In M. Ainsworth, J. Levesley, W. A. Light, and M. Marletta, editors, *Theory of Numerics in Ordinary and Partial Differential Equations*, pages 1–75. Oxford University Press, Oxford, 1995. Lecture notes for a sequence of five lectures at the VI-th SERC Numerical Analysis Summer School, Leicester University, 25 - 29 July, 1994.

[Dan88] J.M.A. Danby. *Fundamentals of celestial mechanics.* Richmond, Va., U.S.A.: Willmann-Bell, 1988.2nd ed., enl., 1988.

[EKW84] J.-P. Eckmann, H. Koch, and P. Wittwer. A computer-assisted proof of universality for area-preserving maps. *Mem. Amer. Math. Soc.*, 47(289):vi+122, 1984.

[GBO$^+$08] J.D. Giorgini, L.A.M. Benner, S.J. Ostro, M.C. Nolan, and M.W. Busch. Predicting the Earth encounters of (99942) Apophis. *Icarus*, 193:1–19, 2008.

[GC91] A. Griewank and G.F. Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application.* SIAM, Philadelphia, Penn., 1991.

[Gib60] A. Gibbons. A program for the automatic integration of differential equations using the method of Taylor series. *Comp. J.*, 3:108–111, 1960.

[Gol91] A. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

[Gri00] A. Griewank. *Evaluating Derivatives.* SIAM, Philadelphia, Penn., 2000.

[HB03] J. Hoefkens and M. Berz. Controlling the wrapping effect in the solution of ODEs for asteroids. *Reliab. Comput.*, 9(1):21–41, 2003.

[Hoe01] J. Hoefkens. *Rigorous numerical analysis with high order Taylor methods.* PhD thesis, Michigan State University, 2001.

[IS90] D.H. Irvine and M.A. Savageau. Efficient solution of nonlinear ordinary differential equations expressed in S-system canonical form. *SIAM J. Numer. Anal.*, 27(3):704–735, 1990.

[JPL] `http://ssd.jpl.nasa.gov/horizons.html`.

[JV97] À. Jorba and J. Villanueva. On the persistence of lower dimensional invariant tori under quasi-periodic perturbations. *J. Nonlinear Sci.*, 7:427–473, 1997.

[JZ05] À. Jorba and M. Zou. A software package for the numerical integration of ODEs by means of high-order Taylor methods. *Exp. Math.*, 14(1):99–117, 2005.

[KS07] T. Kapela and C. Simó. Computer assisted proofs for nonsymmetric planar choreographies and for stability of the Eight. *Nonlinearity*, 20(5):1241–1255, 2007.

[Lan82] Oscar E. Lanford, III. A computer-assisted proof of the Feigenbaum conjectures. *Bull. Amer. Math. Soc. (N.S.)*, 6(3):427–434, 1982.

[Loh] Lohner. http://www.math.uni-wuppertal.de/ xsc/xsc/pxsc_software.html#awa.

[LTWVG$^+$06] M. Lerch, G. Tischler, J. Wolff Von Gudenberg, W. Hofschuster, and W. Krämer. FILIB++, a fast interval library supporting containment computations. *ACM Trans. Math. Software*, 32(2):299–324, 2006.

[Mak98] K. Makino. *Rigorous analysis of nonlinear motion in particle accelerators.* PhD thesis, Michigan State University, 1998.

[MB05] K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model-based verified integrators: long-term stabilization by preconditioning. *Int. J. Differ. Equ. Appl.*, 10(4):353–384 (2006), 2005.

[MCS$^+$05] A. Milani, S.R. Chesley, M.E. Sansaturio, G. Tommei, and G.B. Valsecchi. Nonlinear impact monitoring: line of variation searches for impactors. *Icarus*, 173:362 – 384, 2005.

[MH92] K.R. Meyer and G.R. Hall. *Introduction to Hamiltonian Dynamical Systems and the N-Body Problem.* Springer, New York, 1992.

[Moo66] R.E. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1966.

[Moo79] R.E. Moore. *Methods and applications of interval analysis*, volume 2 of *SIAM Studies in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1979.

[Mou14] F.R. Moulton. *An introduction to celestial mechanics.* New York, The Macmillan company; [etc., etc.] 1914. 2d rev. ed., 1914.

[MZ00] M. Mrozek and P. Zgliczyński. Set arithmetic and the enclosing problem in dynamics. *Ann. Polon. Math.*, 74:237–259, 2000. Dedicated to the memory of Bogdan Ziemian.

[Ned99] N.S. Nedialkov. *Computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation.* PhD thesis, University of Toronto, 1999.

[NEO] `http://unicorn.eis.uva.es/cgi-bin/neodys/neoibo?objects:Apophis`.

[NJ01] N.S. Nedialkov and K.R. Jackson. A new perspective on the wrapping effect in interval methods for initial value problems for ordinary differential equations. In *Perspectives on enclosure methods (Karlsruhe, 2000)*, pages 219–264. Springer, Vienna, 2001.

[NJC99] N.S. Nedialkov, K.R. Jackson, and G.F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. Comput.*, 105(1):21–68, 1999.

[NJN07] M. Neher, K.R. Jackson, and N.S. Nedialkov. On Taylor model based integration of ODEs. *SIAM J. Numer. Anal.*, 45(1):236–262 (electronic), 2007.

[Pol76] H. Pollard. *Celestial mechanics.* Mathematical Association of America, Washington, D. C., 1976. Carus Mathematical Monographs, No. 18.

[Ral81] L.B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science.* Springer Verlag, Berlin, 1981.

[Ron05] R.B. Roncoli. Lunar constants and models document. JPL Technical Document D-32296, 2005.

[Sei92] P. K. Seidelmann. *Explanatory Supplement to the Astronomical Almanac.* Published by University Science Books, 648 Broadway, Suite 902, New York, NY 10012, 1992. Completely revised and rewritten, edited by Seidelmann, P. Kenneth, 1992.

[Sim01] C. Simó. Global dynamics and fast indicators. In H.W. Broer, B. Krauskopf, and G. Vegter, editors, *Global analysis of dynamical systems*, pages 373–389, Bristol, 2001. IOP Publishing.

[SMA] SMART-1. http://smart.esa.int/science-e/www/area/index.cfm?fareaid=10.

[SS71] E. L. Stiefel and G. Scheifele. *Linear and regular celestial mechanics. Perturbed two-body motion, numerical methods, canonical theory.* Springer-Verlag, New York, 1971.

[Ste56] J.F. Steffensen. On the restricted problem of three bodies. *Danske Vid. Selsk. Mat.-Fys. Medd.*, 30(18):17, 1956.

[Ste57] J.F. Steffensen. On the problem of three bodies in the plane. *Mat.-Fys. Medd. Danske Vid. Selsk.*, 31(3):18, 1957.

[SV87] M.A. Savageau and E.O. Voit. Recasting nonlinear differential equations as S-systems: a canonical nonlinear form. *Math. Biosci.*, 87(1):83–115, 1987.

[SW] M.E. Standish and J.G. Williams. Detailed description of the JPL planetary and lunar ephemerides, DE405/LE405. Electronically available as a PDF file, `http://iau-comm4.jpl.nasa.gov/XSChap8.pdf`.

[Wen64] R. E. Wengert. A simple automatic derivative evaluation program. *Comm. ACM*, 7(8):463–464, 1964.

[WZ08] D. Wilczak and P. Zgliczyński. $\mathcal{C}^r$-lohner algorithm. arXiv:0704.0720v1, 2008.

[Zgl02] P. Zgliczynski. C^1 Lohner algorithm. *Found. Comput. Math.*, 2(4):429–465, 2002.

[Zgl07] P. Zgliczyński. Advanced course on Computer Assisted Proofs in Dynamics, University of Barcelona, 2007. The notes of the course can be retrieved from `http://www.imub.ub.es/cap07/slides/`.

Appendix A

N-body variationals, asteroid case

A.1 Set up and variational equations

We want to integrate the variational equations for an asteroid, at the same time that the motion, correcting the position of the N main bodies by using the JPL 405 ephemerides (or not).

The equations of motion for the asteroid can be written as

$$\ddot{x}_A = f_x = \sum_{j=1}^{N} m_j(x_j - x_A)r_{j,A}^{-3}, \qquad r_{j,A}^2 = (x_j - x_A)^2 + (y_j - y_A)^2 + (z_j - z_A)^2,$$

and similar equations for the other coordinates: $\ddot{y}_A = f_y, \ddot{z}_A = f_z$.

Let ξ, η, ζ denote the first three components of the solution of the first order variational equations, written in vector form. It is clear that it is enough to obtain the Taylor solutions of these. Then, the first derivatives with respect to time, easily computable from the jets of ξ, η, ζ, give the remaining 3 components, and starting with the different columns of the identity matrix we recover the full first variational matrix.

Note that the effect of the changes of velocities appears due to the fact that the velocities give the terms of order 1 in the jet of the coordinates. This means that, instead of 6×6 equations we need only to integrate 3×6.

To this end we write

$$\ddot{\xi} = D_x f_x \xi + D_y f_x \eta + D_z f_x \zeta, \quad \ddot{\eta} = D_x f_y \xi + D_y f_y \eta + D_z f_y \zeta, \quad \ddot{\zeta} = D_x f_z \xi + D_y f_z \eta + D_z f_z \zeta,$$

which gives the recurrences to be used in Taylor's method. Obviously $D_x f_x$ denotes the derivative of f_x w.r.t. x_A and similar for the others. As the 3×3 matrix containing the $D_* f_*$ is minus the Hessian of the Hamiltonian w.r.t. the position coordinates of the asteroid, it is symmetric: $D_x f_y = D_y f_x$, etc.

A trivial computation gives

$$D_x f_x = \sum_{j=1}^{N} m_j \left(-r_{j,A}^{-3} + 3(x_j - x_A)^2 r_{j,A}^{-5}\right), \qquad D_y f_x = \sum_{j=1}^{N} m_j 3(x_j - x_A)(y_j - y_A) r_{j,A}^{-5},$$

and similar for the other cases.

To obtain the second (and higher) order variational equations it is better to use indices. Let $w = (w_1, w_2, w_3)^T = (x, y, z)^T$ and denote as w^0 the initial values. The values w_4, w_5, w_6 denote the respective time derivatives. As mentioned these velocities appear as initial conditions (i.e., in w^0) but in w only the position coordinates are necessary. We shall use the shorter notation

$w_{n,i}, w_{m,ij}, w_{m,ijk}$, etc to denote $\frac{\partial w_n}{\partial w_i^0}, \frac{\partial^2 w_n}{\partial w_i^0 \partial w_j^0}, \frac{\partial^3 w_n}{\partial w_i^0 \partial w_j^0 \partial w_k^0}$, etc. From the first variational equations

$$\frac{d^2}{dt^2}\frac{\partial w_m}{\partial w_i^0} = \sum_{k=1}^{3} \frac{\partial f_{w_m}}{\partial w_k}\frac{\partial w_k}{\partial w_i^0}, \qquad \text{or} \qquad \frac{d^2}{dt^2} w_{m,i} = \sum_{k=1}^{3} \frac{\partial f_{w_m}}{\partial w_k} w_{k,i},$$

we obtain immediately

$$\frac{d^2}{dt^2} w_{m,ij} = \sum_{n,p=1}^{3} \frac{\partial^2 f_{w_m}}{\partial w_n \partial w_p} w_{n,i} w_{p,j} + \sum_{n=1}^{3} \frac{\partial f_{w_m}}{\partial w_n} w_{n,ij} \,.$$

Recall that the initial conditions are $w_{m,ij} = 0$ and the same happens for the initial conditions of higher order variational equations. These solutions differ because the equations contain $w_{n,i}w_{p,j}$ or similar terms, which do depend on the initial changes in both positions and velocities. Also the symmetry w.r.t. the indices i, j allows to reduce dimension. Instead of 36 different sets of initial conditions for the initial variations of the positions and velocities w_i^0, w_j^0, it is enough to use 21. This has to be done for each one of the 3 variables w_m. These considerations reduce the total number of second order variationals from 216 to 63, that is $3 \times \begin{pmatrix} 7 \\ 5 \end{pmatrix}$.

The equations involve second derivatives $\frac{\partial^2 f_{w_m}}{\partial w_k \partial w_n}$, which can be computed as

$$\frac{\partial^2 f_x}{\partial x^2} = \sum_{j=1}^{N} m_j \left(-9(x_j - x_A) r_{j,A}^{-5} + 15(x_j - x_A)^3 r_{j,A}^{-7} \right),$$

$$\frac{\partial^2 f_x}{\partial x \partial y} = \sum_{j=1}^{N} m_j \left(-3(y_j - y_A) r_{j,A}^{-5} + 15(x_j - x_A)^2 (y_j - y_A) r_{j,A}^{-7} \right),$$

$$\frac{\partial^2 f_x}{\partial y \partial z} = \sum_{j=1}^{N} m_j 15(x_j - x_A)(y_j - y_A)(z_j - z_A) r_{j,A}^{-7},$$

the remaining ones being obtained from the symmetries or by cyclic permutation of indices. These formulas can be computed in different ways, to minimise the number of operations, as shall be discussed later.

If we go to third and fourth order variationals (we shall refrain to go farther explicitly) the equations become

$$\frac{d^2}{dt^2} w_{m,ijk} = \sum_{n,p,q=1}^{3} \frac{\partial^3 f_{w_m}}{\partial w_n \partial w_p \partial w_q} w_{n,i} w_{p,j} w_{q,k} +$$

$$\sum_{n,p=1}^{3} \frac{\partial^2 f_{w_m}}{\partial w_n \partial w_p} \left(w_{n,ij} w_{p,k} + w_{n,ik} w_{p,j} + w_{n,jk} w_{p,i} \right) + \sum_{n=1}^{3} \frac{\partial f_{w_m}}{\partial w_n} w_{n,ijk} \,,$$

$$\frac{d^2}{dt^2} w_{m,ijkl} = \sum_{n,p,q,r=1}^{3} \frac{\partial^4 f_{w_m}}{\partial w_n \partial w_p \partial w_q \partial w_r} w_{n,i} w_{p,j} w_{q,k} w_{r,l} + \sum_{n,p,q=1}^{3} \frac{\partial^3 f_{w_m}}{\partial w_n \partial w_p \partial w_q} \times$$

$$\left(w_{n,ij} w_{p,k} w_{q,l} + w_{n,ik} w_{p,j} w_{q,l} + w_{n,il} w_{p,j} w_{q,k} + w_{n,jk} w_{p,i} w_{q,l} + w_{n,jl} w_{p,i} w_{q,k} + w_{n,kl} w_{p,i} w_{q,j} \right) +$$

$$\sum_{n,p=1}^{3} \frac{\partial^2 f_{w_m}}{\partial w_n \partial w_p} \left(w_{n,ij} w_{p,kl} + w_{n,ik} w_{p,jl} + w_{n,jk} w_{p,il} + w_{n,ijk} w_{p,l} \right) + \sum_{n=1}^{3} \frac{\partial f_{w_m}}{\partial w_n} w_{n,ijkl} \,.$$

Furthermore the number of equations to be integrated, taking into account the symmetries, for the variational equations of order s is $3 \times \begin{pmatrix} s+5 \\ 5 \end{pmatrix}$. For orders $3, 4, 5, \ldots$ we obtain $168, 378, 756, \ldots$, respectively.

It has to be stressed that, while in the second and third variational equations a factor which comes from a derivative of f_{w_m}, like $\frac{\partial^2 f_{w_m}}{\partial w_n \partial w_p}$, multiplies products of the solutions of the previous variationals equations of the same type, like $w_{n,ij}w_{p,k}$ in the third order variational, for variationals of order greater than 3 this is not true in general. For instance, in the fourth order variationals $\frac{\partial^2 f_{w_m}}{\partial w_n \partial w_p}$ multiplies both terms of the forms $w_{n,ij}w_{p,kl}$ and $w_{n,ijk}w_{p,l}$. This comes from the fact that, in general, in the partitions of a number there exist several partitions in the same number of pieces. In the example $4 = 2 + 2 = 3 + 1$.

A.2 A count on operations

Let us start a count on operations. Assuming that we work up to order M in the Taylor series and that the order is kept fixed for all the computed variationals, we assume that the cost of a product, a square and a power are, respectively,

$$M^2, \qquad \frac{1}{2}M^2 \qquad \text{and} \qquad 2M^2.$$

A first part of the cost comes from the integration of the motion of the N massive bodies. Even if we use JPL ephemerides, the jet for the asteroid depends on the jets of the N bodies. For every term in the sums we have to compute: $\Delta x_{i,j} = x_j - x_i$, $\Delta y_{i,j} = y_j - y_i$, $\Delta z_{i,j} = z_j - z_i$, $\Delta x^2_{i,j}$, $\Delta y^2_{i,j}$, $\Delta z^2_{i,j}$, that is, 3 squares. Then we have immediately $r^2_{i,j}$ and it remains to compute a power $(r^2_{i,j})^{-3/2}$ and then its products by $\Delta x_{i,j}$, $\Delta y_{i,j}$, $\Delta z_{i,j}$, etc. In all we have an estimate

$$\begin{pmatrix} N \\ 2 \end{pmatrix} \times (3 \times \frac{1}{2} + 2 + 3) \times M^2 = \frac{13}{2} M^2 \begin{pmatrix} N \\ 2 \end{pmatrix}.$$

Next step is the computation of the jet of the asteroid. Instead of following the above method, we compute in a slightly different order which is a little bit better when including variationals. For each one of the N bodies we compute the contributions as:

$$\Delta x_{j,A},\ \Delta y_{j,A},\ \Delta z_{j,A},\ \Delta x^2_{j,A},\ \Delta y^2_{j,A},\ \Delta z^2_{j,A},\ r^2_{j,A},\ 1/r_{j,A},\ 1/r^2_{j,A},$$

$$\Delta x_{j,A}/r^2_{j,A},\ \Delta y_{j,A}/r^2_{j,A}, \Delta_{j,A}/r^2_{j,A},\ \Delta x_{j,A}/r^3_{j,A},\ \Delta y_{j,A}/r^3_{j,A},\ \Delta_{j,A}/r^3_{j,A}.$$

In all, 4 squares, 1 power and 6 products, with a cost $10NM^2$.

Then the successive derivatives, $\frac{\partial f_{w_m}}{\partial w_n}, \frac{\partial^2 f_{w_m}}{\partial w_n \partial w_p}, \frac{\partial^3 f_{w_m}}{\partial w_n \partial w_p \partial w_q}$ are computed, recurrently, by multiplication of jets already available by the jet of one of the expressions $1/r^2_{j,A}$, $\Delta x_{j,A}/r^2_{j,A}$, $\Delta y_{j,A}/r^2_{j,A}$, $\Delta_{j,A}/r^2_{j,A}$. The cost for the derivatives of increasing order of f which appear in the first, second, third, fourth, fifth, ... order variationals is, in multiples of NM^2, equal to $1 + 6 = 7$, $3 + 10 = 13$, $1 + 6 + 15 = 22$, $3 + 10 + 21 = 34$, $1 + 6 + 15 + 28 = 50$, etc. It will be clear in a moment that this cost becomes negligible, for a fixed N, when the order of the variationals increases.

The computation of the right hand sides of the variationals requires several steps:

1) Computing the derivatives of f_{w_m}, already discussed in last paragraph.

2) The computation of all the intermediate products, that is, terms of the form $w_{n,i}w_{n,j}$, $w_{n,i}w_{n,j}w_{p,j}$, $w_{n,ij}w_{n,k}$, $w_{n,i}w_{n,j}w_{p,j}w_{r,l}$, $w_{n,ij}w_{n,k}w_{p,l}$, $w_{n,ij}w_{n,kl}$, $w_{n,ijk}w_{n,l}$, etc. It is clear that all these products can be obtained recurrently from the previous ones by a single multiplication. Hence the cost is just the number of products of each one of the types, taking into account the symmetries. Some of them are squares, but the fraction is small and we shall not take advantage of that fact.

   Up to order five there are 13 types of products. Each type of product is identified as a partition. So, the partition $2+1+1$ identifies products of the form $w_{n,ij}w_{n,k}w_{p,l}$. The table summarises the count.

1+1	171
1+1+1	1140
2+1	$63 \times 18 = 1134$
1+1+1+1	5985
2+1+1	$63 \times 171 = 10772$
2+2	2016
3+1	$168 \times 18 = 3024$
1+1+1+1+1	26334
2+1+1+1	$63 \times 1140 = 71820$
2+2+1	$2016 \times 18 = 36288$
3+1+1	$168 \times 171 = 28728$
4+1	$378 \times 18 = 6804$
3+2	$168 \times 63 = 10584$

3) The products of the intermediate ones by the derivatives of f_{w_m}. It is clear that all the intermediate products that multiply the same derivative of f_{w_m}, taking into account the symmetries of these derivatives in each equation, can be added before doing the corresponding product. Hence, in each one of the first variationals one needs 3 such products, in each one of the second ones $6+3=9$, etc.

   In general, in each one of the variational equations of order s the number of these final products is $\binom{s+3}{3} - 1$. This has to be multiplied by the number of equations, as given in next table, which presents order, number of equations, final products per equation and total number of final products.

1	18	3	54
2	63	9	567
3	168	19	3192
4	378	34	12852
5	756	55	41580

4) As final step we have to add all costs found till now. This is given as a function of the order s of the variationals (from 0 to 5) and the cost is expressed in multiples of M^2, where we recall that M is the order of the Taylor method. One could also consider different Taylor orders for the successive variationals.

   In the third column of next table we produce the values for $N = 10$ massive bodies (rounded to next integer) and in the fourth one the accumulated values $C(s)$. One can check that the role of N is negligible for $s > 2$. Furthermore, for $s > 3$ most of the cost comes from the computation of the intermediate products.

0	$13N(N-1)/4+10N$	393	393
1	$7N+54$	124	517
2	$13N+738$	868	1375
3	$22N+5466$	5686	7061
4	$34N+34650$	34990	42051
5	$50N+222138$	222638	264689

A fit of the accumulate results shows that the rate of increase for $s > 1$ behaves roughly like $C(s) = c_1 \exp(c_2 s)$, but that value of c_2 increases slowly when only the last 3 or 2 points are taken. A working value of c_2 can be 1.85.

A.3 On the optimal variational order to propagate a box

Now we face the problem of the optimality of the order of the variationals when we want to propagate a given box. We start with a suitable set up and assumptions:

a) We start at some time $t_0 = 0$ and want to arrive at a time t_f. The goal is to transport a box with sides of typical size ℓ. The analysis can be easily modified if not all of the sides are equal. There is freedom in the selection of the order of the variationals used to transport boxes.

b) We assume that the error in the representation of the map time-t_f using up to order s in the variationals, is of the type $K_f(\ell/R_f)^{s+1}$, where R_f is the radius of convergence and K_f the factor in the Cauchy estimates. One desires a bound of the error of the form

$$K_f(\ell/R_f)^{s+1} \leq \varepsilon,$$

for some fixed value of ε.

c) Independently of the dimension of the problem we assume that the problem has mainly d unstable directions, either because of positive Lyapunov exponents or because there is a strong shear in one or more directions. For instance, in Kepler's problem or mildly perturbed Kepler's problems one has $d = 1$ because of the shear along the orbit.

d) To achieve the goal it is possible to subdivide the box of size ℓ in boxes of size ℓ/m in each one of the d directions. Of course, the values of m can be different in each direction, but we shall take them equal for simplicity. An increase of the order of the variationals allows to take less subboxes, at the price of a higher cost. We want to optimise.

e) We also assume that the number of steps S_f is independent of the order of the variationals used. Furthermore, for simplicity, we assume the subdivision is done at t_0. Other strategies are possible, using different subdivisions at different stages of the integration.

Then the value of m is selected as $m = \dfrac{\ell}{R_f}\left(\dfrac{\varepsilon}{K_f}\right)^{-1/(s+1)}$ and the total cost is

$$S_f C(s)\left[\frac{\ell}{R_f}\left(\frac{\varepsilon}{K_f}\right)^{-1/(s+1)}\right]^d.$$

We assume $C(s) = c_1 \exp(c_2 s)$, introduce $\hat{\varepsilon} = \varepsilon / K_f$ and take logarithms. Skipping all terms not depending on s, we obtain

$$c_2 s + \frac{d}{s+1} \log(\hat{\varepsilon}^{-1}),$$

for which the minimum is attained for $s + 1 = \sqrt{d \log(\hat{\varepsilon}^{-1})/c_2}$.

A rough application to the asteroid problem assuming $d = 1, c_2 = 1.85, \hat{\varepsilon} = 10^{-20}$ gives $s \approx 4$. One can check the behaviour of $C(s)\hat{\varepsilon}^{-1/(s+1)}$ using the available values of $C(s)$. The figures obtained for $s = 2, 3, 4, 5$ are, respectively, 6.38×10^9, 0.706×10^9, 0.420×10^9, 0.570×10^9. This shows that, indeed, $s = 4$ is a good choice but that also one could use $s = 3$ with a larger cost, still admissible, and easier to implement.

A.4 On the integration of a finite jet

Another possibility is to transport a jet. More concretely, assume that some equation $x' = f(x)$ starts at an initial point (t_0, x_0). At every step of the time integration, say, at time t, we can assume that if at time t_0 we start at point $x_0 + \xi$ one has a truncated Taylor expansion representation of the solution of the form

$$\varphi_t(x_0 + \xi) = \varphi_t(x_0) + P_k(\xi) + O(|\xi|^{k+1}),$$

where $P_k(\xi)$ denotes a polynomial of degree k in the ξ variables, without independent term.

Of course, the coefficients in P_k depend on time. We want to propagate this representation to a new time.

An alternative to the use of variational equations, much on the spirit of Taylor methods is as follows.

In the formulas for the integration of the differential equation one can start, not with initial conditions x_0 but with $x_0 + \xi$, keeping ξ as formal variables. Then all the computations are carried out as polynomials in ξ up to some prescribed order, say k. Hence, instead of computing a jet with respect to t having numbers as coefficients, we have polynomials in n variables up to order k. It is possible to assign weights to the ξ variables to do the computations up to a given weighted order. But we shall confine the description to the usual weights.

Starting with linear data $(x_0)_i + \xi_i, i = 1, \ldots, n$ the propagation up to a final time t produces the desired $\varphi_t(x_0 + \xi)$ up to order k.

A.4.1 Operation count

A key point is the count on the number of operations in contrast to the previous counts using variational equations. It is clear that now every arithmetic operation $(*, +)$ we had before, is replaced by operations with polynomials of degree k in n variables, with the result truncated to order k. The computation of sums can be neglected in front the one of the products. Let $c(n, k)$ be the cost of one product in n variables to order k. This cost denotes the number of arithmetic operations (1 sum and 1 product) to be done. The table gives, as an example, the values of $c(n, k)$ for $n = 6$ as a function of k.

The main task when integrating N-body problems is the computation of squares, power $-3/2$ and products. As the sums can be neglected, if we keep order M with respect to t the cost of every square is, roughly, $c(n, k)/2$ and the products and powers is $c(n, k)$. The total cost depends on M.

To fix ideas we start with the 3D Kepler's problem. The cost of one step will be $\frac{11}{2} M^2 c(6, k)$. For $k = 4, 5$ this amounts, respectively, to $10010M^2, 34034M^2$, which compares in a very favourable way to the previous table if we skip the contribution of the N bodies.

1	13	7	50388
2	91	8	125970
3	455	9	293930
4	1820	10	646646
5	6188	11	1352078
6	18564	12	2704156

However, if we integrate an asteroid under the influence of N bodies, even if the jets of the N bodies are only needed for their contribution to the one of the asteroid (and then we recover the "correct" data for the N from JPL ephemerides, the computation with the 6 variables ξ must be done in each one of the terms of the form $(x_j - x_A)r_{j,A}^{-3}$. Essentially this multiplies by N the cost of carrying out the computation, even taking into account that we only want the expansion in ξ for the motion of the asteroid.

Appendix B

Convergence and remainder

We consider the problem of rigorous integration of Kepler's problem. Of course everything is known analytically. But trying to extend this to the integration of a planetary system, it is convenient to derive estimates, even rough estimates, which can be extended to the general case.

The main idea is similar to the ideas used in the proof of Cauchy's theorem for analytic vector fields by majorant method.

The equations of motion can be written as

$$\ddot{x} = f_x = -xr^{-3}, \quad \ddot{y} = f_y = -yr^{-3}, \quad \ddot{z} = f_z = -zr^{-3},$$

where $r^2 = x^2 + y^2 + z^2$. We are assuming that the factor $\mu = GM = 1$. This can always be achieved by selection of units. We also assume that the initial distance, at $t = 0$ is 1 and from now on we replace x, y, z by $x_0 - x, y_0 - y, z_0 - z$, that is, the point (x_0, y_0, z_0) is the initial location (in fact, the location at the beginning of every time step) and x, y, z denote the changes with respect to that point. Furthermore, as the equations are invariant under the action of $SO(3)$, it is not restrictive to take $(x_0, y_0, z_0) = (1, 0, 0)$.

As a first thing we want to obtain estimates of the rate of increase of the coefficients in the equations of motion. We start with the term $r^{-3/2} = ((1-x)^2 + y^2 + z^2)^{-3/2}$. Assume that we take x, y, z as complex variables in a disc or radius R. It is clear that the worst situation consists in taking $x \in \mathbb{R}$, $y \in \mathrm{i}\,\mathbb{R}$, $z = \mathrm{i}\,\mathbb{R}$. If $x = R, y = \mathrm{i}\,R, z = \mathrm{i}\,R$ then the condition $|r^2| > 0$ implies implies $R < \sqrt{2}-1$. Hence, the rate of increase of the coefficients in f_x, f_y, f_z is $c = 1/R = \sqrt{2}+1$, that is f_w is majorated as follows:

$$|C_{m,n,p}| \leq Kc^{m+n+p},$$

where $C_{m,n,p}$ is the coefficient of $x^m y^n z^p$ and w denotes any of the variables x, y, z.

In fact we are not only interested in $r^{-3/2}$ but on $(1-x)r^{-3/2}, yr^{-3/2}, zr^{-3/2}$. It is easy to obtain the expansions of these functions and an elementary (but tedious) computation involving combinatorial numbers gives that the constant K above can be taken equal to 1 for the three functions. This proves, in particular

$$|f_w| \leq \sum_{m,n,p \geq 0} |x|^m |y|^n |z|^p c^{m+n+p} = \frac{1}{(1-c|x|)(1-c|y|)(1-c|z|)}.$$

Let $v \geq 0$ be the maximum of the absolute values of the components of the velocity at the initial time, say $t = 0$. The majorant method shows that the variables x, y, z have a common bound given by the solution of

$$\ddot{x} = \frac{1}{(1-cx)^3}, \quad x(0) = 0, \quad \dot{x}(0) = v.$$

This is a Hamilton equation with Hamiltonian $H(x,p) = \dfrac{1}{2}p^2 - \dfrac{1}{2c(1-cx)^2}$. It is immediate to find the explicit solution. Let $\gamma = v^2c - 1$. Then

$$x(t) = \frac{1}{c}\left[1 - \left(1 - 2cvt + \gamma ct^2\right)^{1/2}\right].$$

This has two applications: a lower (and quite rough) estimate of the radius of convergence of the solution of the initial system is given by value of t for which $1 - cx = 0$ in the expression above, that is $t = t_f = \dfrac{1}{vc + \sqrt{c}}$. We also introduce, for later use, the value $t_g = \dfrac{1}{vc - \sqrt{c}}$ and note that $x(t) = \frac{1}{c}\left[1 - ((1 - t/t_f)(1 - t/t_g))^{1/2}\right]$.

Second application is the estimate of the remainder. Expanding $\left(1 - 2cvt + \gamma ct^2\right)^{1/2} = (1 - t/t_f)^{1/2}(1 - t/t_g)^{1/2}$ one finds, for the coefficient of t^n, say a_n, the expression

$$|a_n| = c^{-1}\left|\sum_{k=0}^{n} \begin{pmatrix} 1/2 \\ n-k \end{pmatrix} \begin{pmatrix} 1/2 \\ k \end{pmatrix} t_g^{-k} t_f^{k-n}\right|.$$

In particular, one also obtains $|a_n| \le c^{-1}t_f^{-n}$.

As an example we can consider the case $v = 1$. Then the radius of convergence is bounded from below by $1/(\sqrt{c} + c) \approx 0.252$. The values $|a_n|t_f^n$ decrease from some point on. If one uses order 30, it is enough to take a step 0.1 to have a bound of the error below 10^{-15}.

As said before this approach is easily extended to N-body problems, by adding to the majorant of the vector field all the contributions. Anyway, the estimates are relatively pessimistic because we are considering the worst case in a complex ball around the initial point.

Similar results can be applied to general analytic equations. If we assume $\dot{x} = f(x)$, $x \in \mathbb{R}^n$, it is possible to produce estimates of the radius of convergence and of the tail of the Taylor series from the Cauchy estimates of f in a complex ball around the given point. Rough values of these estimates can be obtained from bounds of f in a ball and the residue theorem.

Anyway, this approach has to be compared to the use of variational equations of moderate order to obtain a good and accurate enclosure. This should allow better time steps.