DIFFERENTIAL ALGEBRA SPACE TOOLBOX FOR NONLINEAR UNCERTAINTY PROPAGATION IN SPACE DYNAMICS

M. Rasotto[*], *A. Morselli*[†], *A. Wittig*[‡], *M. Massari*[§], *P. Di Lizia*[§], *R. Armellin*[¶], *C. Y. Valles*[‡] *and G. Ortega*[‡]

[*] Dinamica Srl, Via Morghen 13, 20158 Milano, Italy.
[†] European Space Agency (ESA/ESOC), Robert-Bosch-Str. 5, 64293 Darmstadt, Germany.
[‡] European Space Agency (ESA/ESTEC), Keplerlaan 1, 2201 AZ Noordwijk, Netherlands.
[§] Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy.
[¶] Universidad de La Rioja, Avda. de la Paz, 93, 26006 Logroño, La Rioja, Spain.

ABSTRACT

This paper is aimed at presenting a new tool developed by Dinamica, with the support of ESA, for the efficient nonlinear propagation of uncertainties in space dynamics. The newly implemented software is based on Differential Algebra, which provides a method to easily extend the existing linearization techniques and allows the implementation of efficient arbitrary order methods. These theoretical concepts represent the building blocks over which the Differential Algebra Space Toolbox is implemented. The application areas for the tool are plenty. To illustrate the power of the method in general and to give the user a better understanding of the various features, several different examples in the field of astrodynamics and space engineering are presented.

Index Terms— Nonlinear uncertainty propagation, Differential Algebra.

1. INTRODUCTION

The problem of uncertainty propagation represents a crucial issue in spaceflight dynamics since all practical systems - from vehicle navigation to orbit determination or target tracking - involve nonlinearities of one kind or another. In addition, within the space mission design process, uncertainty propagation is a fundamental tool to assess the fulfillment of mission requirements and constraints, to evaluate mission performances, to perform sensitivity analyses, and to verify the robustness of guidance and control laws. In light of the above, the scientific community has started focusing on the development of new tools, aiming at improving the approximation of standard linear methods available in the literature or to reduce the computational effort required by standard Monte-Carlo simulations. Dinamica Srl, with the support of ESA, has recently completed the implementation of the Differential Algebra Space Toolbox (DAST), a software tool based on Differential Algebra (DA) for the efficient, nonlinear propagation of uncertainties in space dynamics. Although the general mathematical foundation suggests potential applications of the same approach to various fields, this work focuses on the application of DAST to a set of test cases in the field of astrodynamics and space engineering, with the aim of illustrating the potentials with respect to classical approaches. Thanks to the use of DA techniques, indeed, DAST has been shown to be orders of magnitude more efficient than traditional methods. The manuscript is organized as follows. First, a brief introduction to differential algebra is given, focusing in particular to the high-order expansion of the flow of ODEs. Then, three different high-order methods for the nonlinear propagation of uncertainties are described in detail, highligthing for each the role played by differential algebra. The following section is mainly aimed at presenting the architecture of the DAST, including a detailed description of each module. Finally, the effectiveness of the tool is demonstrated through several numerical examples.

2. DIFFERENTIAL ALGEBRA

Historically, the treatment of functions in numerics has been based on the treatment of numbers, and the classical numerical algorithms are based on the mere evaluation of functions at specific points. DA technique relies on the observation that it is possible to extract more information on a function rather than its mere values. The basic idea is to bring the treatment of functions and the operations on them to the computer environment in a similar way as the treatment of real numbers. In fact, the real numbers cannot be treated, in a strict sense, in a computer environment, instead they are approximated by floating point (FP) numbers with a finite number of digits. With reference to Figure 1, let us consider two real numbers a and b, and their floating point counterpart $\bar{a}$ and $\bar{b}$ respectively; then, given any operation "$*$" in the set of real numbers, an adjoint operation "$\otimes$" is defined in the set of floating point numbers such that the diagram in figure commutes. Consequently, transforming the real numbers a and b in their FP representation and operating on them in the set of FP num-

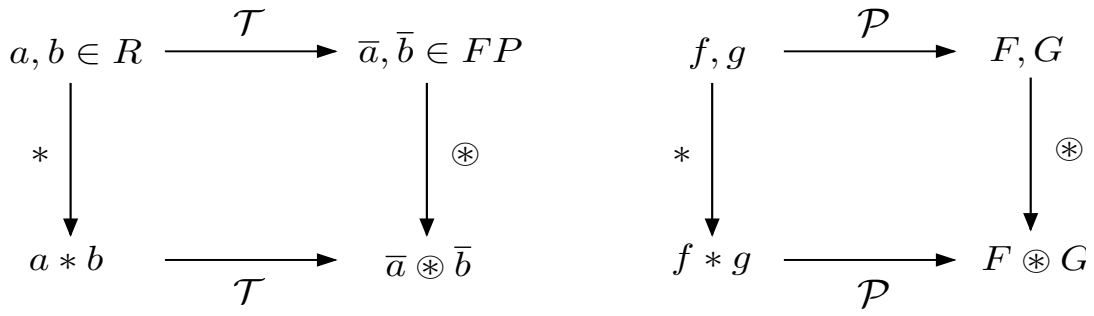

Fig. 1. Analogy between the FP representation of real numbers in computer environment (left) and the algebra of Taylor polynomials in DA framework (right)

bers returns the same result as carrying out the operation in the set of real numbers and then transforming the achieved result in its FP representation. In a similar way, suppose two sufficiently regular functions f and g are given. In the framework of differential algebra, the computer operates on them using their Taylor series expansions, F and G respectively. Therefore, the transformation of real numbers in their FP representation is now substituted by the extraction of the Taylor expansions of f and g. For each operation in the function space, an adjoint operation in the space of Taylor polynomials is defined such that the corresponding diagram commutes; i.e., extracting the Taylor expansions of f and g and operating on them in the function space returns the same result as operating on f and g in the original space and then extracting the Taylor expansion of the resulting function.

The straightforward implementation of differential algebra in a computer allows computing the Taylor coefficients of a function up to a specified order k, along with the function evaluation, with a fixed amount of effort. The Taylor coefficients of order k for sums and product of functions, as well as scalar products with real numbers, can be computed from those of summands and factors; this means that the set of equivalence classes of functions can be endowed with well-defined operations, leading to the so called truncated power series algebra (TPSA). In addition to sum and product, other algebraic operations can be performed, as composition/inversion of functions or nonlinear systems solution. Moreover, the analytic operations of differentiation and integration have been developed on these function spaces, defining a differential algebraic structure, [1].

2.1. High-order expansion of the flow

The high-order expansion of the flow can be obtained by considering the fact that any integration scheme, explicit or implicit, is characterized by a finite number of algebraic operations, involving the evaluation of the ODEs right-hand side at several integration points. Therefore, replacing the operations between real numbers with those on DA numbers, it yields to the kth-order Taylor expansion of the flow of the ODE, $\phi\left(t; \delta x_0; t_0\right) = \mathcal{M}_\phi\left(\delta x_0\right)$, at each integration time, assuming a perturbed initial condition $x_0 + \delta x_0$.

Without loss of generality, consider the scalar initial value problem:

$$\dot{x}\left(t\right) = f\left(t, x\right), \qquad x\left(t_0\right) = x_0 \tag{1}$$

and the associated flow $\phi\left(t; \delta x_0; t_0\right)$. For the sake of simplicity, consider uncertain initial conditions only. Starting from the kth-order DA representation of the initial condition, $[x_0] = x_0 + \delta x_0$, which is a $(k+1)$-tuple of Taylor coefficients, and performing all the operations in the DA framework, allows us to propagate the Taylor expansion of the flow in x_0 forward in time, up to the final time t_f.

For the sake of clarity, consider the forward Euler's scheme:

$$x_i = x_{i-1} + f\left(x_{i-1}\right)\Delta t \tag{2}$$

and substitute the initial value with the DA identity $[x_0] = x_0 + \delta x_0$. At the first time step one has

$$[x_1] = [x_0] + f\left([x_0]\right)\Delta t \tag{3}$$

If the function f is evaluated in the DA framework, the output of the first step, $[x_1]$, is the kth-order Taylor expansion of the flow $\phi\left(t; \delta x_0; t_0\right)$ in x_0 for $t = t_1$. Note that, as a result of the DA evaluation of $f\left([x_0]\right)$, the $(k+1)$-tuple $[x_1]$ may include several non zeros coefficients corresponding to high order terms in δx_0. The previous procedure can be inferred through the subsequent steps. The result of the final step is the kth-order Taylor expansion of $\phi\left(t; \delta x_0; t_0\right)$ in x_0 at the final time t_f. Thus, the flow of a dynamical system can be approximated, at each time step t_i, as a kth-order Taylor expansion in a fixed amount of effort.

3. NONLINEAR UNCERTAINTY PROPAGATION

The use of DA and the consequent availability of a high order expansion of the flow, paves the way to the nonlinear uncertainty propagation through the implementation of new advanced methods, referred to as high order methods. They represent a trade-off between the efficiency of linearized models

and the accuracy of classical Monte Carlo simulations using high order information:

1. DA-based Monte Carlo, Monte Carlo simulations based on the repeated evaluation of one single flow expansion rather than on multiple pointwise integrations [2, 3, 4];

2. Covariance propagation, analytically mapping PDF [5], or computing high order moments of the mapped PDF [6, 7, 8];

3. Bounding-box propagation, range estimation using polynomial bounder [9].

3.1. DA-based Monte Carlo

A DA-based Monte Carlo (DAMC) requires the following steps:

1. Perform a single DA integration with given expansion order;

2. Generate random samples based on the statistical distribution of the uncertainty to be propagated;

3. Evaluate the flow expansion map for all the samples;

4. Perform the statistical analysis of the results.

With respect to classical Monte Carlo approach, this method allows avoiding multiple numerical integration of the ODE system, replacing it with fast polynomial evaluations, thus reducing the computational burden typically associated to classical Monte Carlo simulation.

Moreover, the DA-based approach delivers a map expansion, whose accuracy can be controlled by changing the expansion order. Furthermore, as the flow expansion is analytical, an analytic framework is delivered, which can be used for additional analyses. For instance, in case new statistics (different number of samples and/or different initial distributions) need to be propagated, it is not necessary to perform an additional DA integration, as only steps 2-4 are required.

The ratio between the computational time of a DA-based Monte Carlo simulation and its pointwise counterpart is

$$\tau = \frac{t_n + n_s t_e}{n_s t_0} \tag{4}$$

where t_n, t_e, and t_0 are the computational times of a k-th order DA integration, a flow map evaluation, and a pointwise integration, respectively; and n_s is the number of samples of the Monte Carlo simulation. The computational cost of a Taylor map evaluation depends on the expansion order, but it is negligible with respect to that of a pointwise integration. For this reason, the expression above can be approximated with R_τ/n_s, in which R_τ is the ratio between the computational time of a k-th order DA integration and a pointwise integration. The value of R_τ strongly depends on the expansion order and the dynamical model, but in any case is orders of magnitude smaller than the number of samples required for a good representation of the statistics (e.g. $> 10^4$ for NEO case).

In the rest of this work, whenever a DA-based Monte Carlo approach is used, it is referred to as DAMC-k, where k indicates the Taylor series expansion order.

3.2. Covariance propagation

A Gaussian pdf remains Guassian under a transformation when the transformation is linear. Thus, for small value of uncertainties, weakly nonlinear dynamical systems or short-term propagation, the approach based on the linearization of the dynamics can deliver enough accurate result in a very efficient way. An initial covariance matrix $P(t_0)$ can be mapped forward in time by

$$P(t) = \Phi(x_0, t) P(t_0) \Phi(x_0, t)^T \tag{5}$$

where $\Phi(x_0, t)$ is the state transition matrix (STM). Through the use of DA, the computation of STM is straightforward and allows avoiding the drawbacks related with the use of variational equations or finite differences. In the first case, the equations are problem dependent, thus need to be analytically derived and numerically integrated for each set of ODEs. Moreover, when coordinate transformations are needed, a linearized version of this transformations must also be derived. Finite differences on the other side, are more suitable for automation, but a careful selection of the size of the perturbation required to approximate the derivatives is necessary. The steps required for the DA-based approach are:

1. Perform a single DA integration (and coordinate transformation when necessary) with expansion order 1. The propagated mean is the constant part of the obtained polynomial map;

2. Extract the STM from the Taylor polynomial map (linear part of the polynomial);

Since the use of relation (5), throughout this work we will refer to this method as Covariance Propagation or Linearized Dynamics (LD).

3.3. Bounding-box propagation

In some applications (e.g. to verify the satisfaction of constraints) it is not necessary to map a statistical distribution, but it is sufficient to estimate the range of the propagated uncertainties. Once the high order expansion of the flow is obtained via DA integration, the range can be estimated by bounding the range of the resulting Taylor polynomials. The implementation of this method can be summarized as follows:

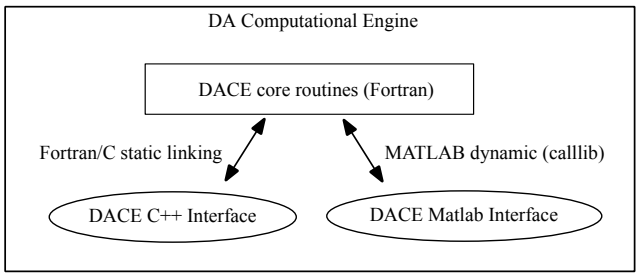

Fig. 2. Overview of the DACE architecture.

1. Scale the uncertain variables by the initial range provided by the user such that the domain of each variable is [-1,1];
2. Perform a single DA integration with expansion order selected by the user.
3. Compute the range of the polynomial representing the Taylor expansion of the flow, by assessing the bounds of each monomial.

In the following sections, we will use both Bounding-box propagation or Polynomial Bounder (PB) to refer to this class of high-order methods.

4. SOFTWARE ARCHITECTURE

The theoretical concepts presented in sections 2-3 have been implemented by Dinamica Srl in the Differential Algebra Space Toolbox (DAST), a DA-based software tool for the efficient, nonlinear propagation of uncertainties in space dynamics (ESA contract: *"ITT AO/1-7570/13/NL/MH Nonlinear Propagation of Uncertainties in Space Dynamics based on Taylor Differential Algebra"*). DAST is divided in three main layers: the Differential Algebra Computational Engine (DACE), the Software Framework (SF), and the Uncertainty Propagation Tool (UPT).

4.1. Differential Algebra Computational Engine (DACE)

This module provides a set of basic DA routines with the aim to approximate the result of an operation by its Taylor expansion around zero. After each operation an approximation is obtained, yielding eventually to the Taylor expansion of arbitrarily complex expressions.

As shown in Figure 2, DACE module is composed by three different sub-modules:

- the DACE Fortran core;
- the C++ interface;
- the Matlab interface;

4.1.1. DACE Fortran core

This sub-module contains the most basic and time-critical operations since they form the foundation of all the higher-level algorithms. To maximize the performance, it is implemented in Fortran 95, an efficient procedural language, with some modern features for what concerns memory management and exception handling. These routines are not intended to be used directly by the user or developer, and their exact Fortran interface is considered private. A convenient access to them is provided through two interface modules.

4.1.2. DACE C++ interface

The C++ interface, based on a mature and wide spread object oriented language, blends in with some useful features (e.g. strict typing, operator overloading for intuitive coding, namespaces to avoid naming collisions, templates (STL), I/O streams for seamless input and output, etc.). Furthermore, it is characterized by a very little overhead, which perfectly suits with high performance requirements.

The design of C++ interface makes use of advanced object oriented techniques and the programming APIs are mainly intended for the inclusion of the DACE in the development of custom code. To this aim, a new DA data type is defined, and all the associated mathematical functions are designed to mimic the calling conventions of the Standard C++ math functions. More in detail, the DA arithmetic data type encapsulates DA routines, handles all aspects of the DACE core routines (calling to the DACE Fortran library, DA allocation) and integrates a modern error management based on C++ exception handling.

Thanks to the powerful template features of C++, the transition from traditional to DA enabled code is simple and straight forward. However, it is worth clarifying that for non-trivial code some changes due to conceptual differences between DA and other arithmetic data types might still be required.

The C++ interface and the DACE Fortran core are integrated in a single DACE Shared Library.

4.1.3. DACE Matlab interface

The second interface is the DACE Matlab interface, based on a procedural language, whose main advantages are the large user base, the interactivity, the automatic memory management, and the availability of built-in functions.

The design is almost similar to the one used in C++ interface, but with the aim to define a new DA type that can be used to perform DA operations directly and interactively from within Matlab. Thus, a Matlab class is implemented to mimic the calling conventions of a subset of the Matlab intrinsic functions. This class encapsulates the calling to the DACE Fortran library, through the loading of the DACE shared library.

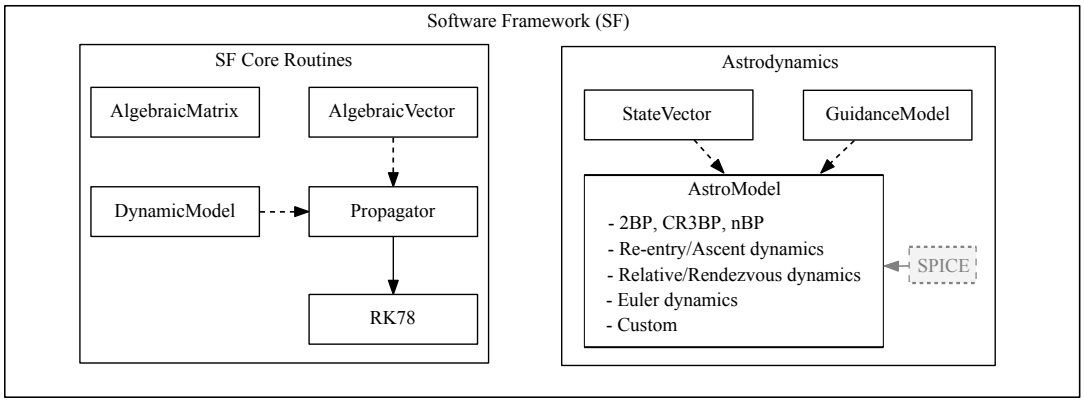

Fig. 3. Overview of the SF structure.

Anyway, due to a limited support for C and Fortran routines, an incomplete object-oriented framework, and a huge overhead for object and memory handling, the current Matlab interface has shown to be much slower, thus is not recommended for serious applications.

4.2. Software Framework (SF)

The second main module includes the implementation of more advanced features, like algorithms to properly handle vectors and matrices of DA, propagation schemes (RK78), etc. These functions, which are grouped in the SF Core Routine sub-module (see Figure 3), have a general purpose and can be used for the application of DA to any dynamical system.

Moreover, SF provides all the required routines to perform DA-based propagations in astrodynamics, including coordinate and time transformations, ephemeris evaluation, as well as the implementation of various dynamical systems and specific astrodynamics routines (DA Lambert problem solver, DA Kepler solver etc.). All these routines form the Astrodynamics block in Figure 3. In order to standardize the evaluation of astrodynamics routines, the SF is linked to the SPICE library. This freely available library already provides most of the astrodynamics routines implemented in the SF for double precision numbers ([10]). Its use allows the SF to only focus on the DA evaluation of these routines.

4.3. Uncertainty Propagation Toolbox (UPT)

The purpose of the UPT is to allow users to perform uncertainty propagations and statistical analyses from Matlab, without the need for writing or compiling their own code. Differently from the DACE Matlab interface described in Section 4.1.3, the UPT is a small component that provides access to the SF routines by using Matlab Mex-files. It allows the user to provide inputs for uncertainty propagation in Matlab, and then interfaces those inputs with the SF, where the actual uncertainty propagation is performed. The results are finally returned back to Matlab, where they can be analyzed and/or used for statistical assessments. Figure 4 shows the logical components structure of the UPT. As can be seen, it is composed by:

- Interface routines, for the declaration of dynamical model and propagation method;
- Mex core routines, for the interaction with the SF;
- Analysis routines, for the evaluation of final DA map and computation of statistical quantities;
- Output routines, for the export and plot of the results;

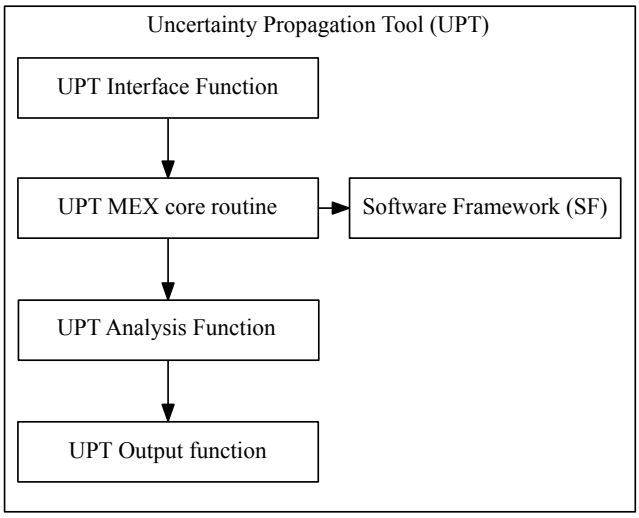

Fig. 4. Overview of the UPT structure.

5. ASTRODYNAMICS APPLICATIONS

To properly illustrate the power of the method and give a better understanding of the various features, two main test cases

in the field of astrodynamics and space engineering are presented: the simple two-body problem and a more advanced re-entry problem.

5.1. Two-body problem

In the two-body problem, the dynamics of an uncontrolled artificial satellite orbiting the Earth is described by the second-order differential equations

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{\boldsymbol{r}^3}\boldsymbol{r}, \tag{6}$$

where $\boldsymbol{r}$ is the position vector of the spacecraft, and μ is the Earth's gravitational parameter. The initial states assumed for the analysis is reported in Table 1 both in terms of cartesian coordinates and orbital elements.

Table 1. Initial state in the 2-body problem.

Element	Value	Units	Coord.	Value	Units
a	1.5e4	km	r_x	7.5e3	km
e	0.5	-	r_y	0.0	km
i	0.0	rad	r_z	0.0	km
Ω	0.0	rad	v_x	0.0	km/s
ω	0.0	rad	v_y	8.9286	km/s
θ	0.0	rad	v_z	0.0	km/s

For the test, only the r_x and r_y components of the position vector are affected by uncertainty δx and δy, with a variance of 5e-02 km^2. The propagation is performed for 30.8 orbital periods, thus computing the statistics in the proximity of the pericenter of the orbit (the initial condition). The analysis is split into two separate phases with the goal of assessing:

1. The consistency of the results coming from different propagation methods;
2. The performance and accuracy with respect to the expansion order.

5.1.1. Consistency with respect to different propagation methods

The first part of the test includes the use of the propagation methods presented in Section 3, followed by a comparison of the associated results.

Firstly, the consistency between DAMC and LD methods is analyzed. The initial conditions, reported in Table 1, are initialized as first order DA variables and propagated forward to get the Taylor expansion of the final state $[\widetilde{\boldsymbol{x}}_f] = \mathcal{M}^1_{x_f}(\delta\boldsymbol{x}_0)$. The obtained map is then used to perform two different analyses. The first one consists in the evaluation of the polynomial with an initial set of samples, generated from a Guassian distribution. This approach is generally referred to as DA-based Monte Carlo analysis (see Section 3.1). The resulting distribution can be used to compute statistical information, such as the final covariance Cov_f^{DAMC}. In the second analysis, the LD method (see Section 3.2) is exploited. The state transition matrix is extracted from the linear part of the map $\mathcal{M}^1_{x_f}$ and used to evaluate the final covariance Cov_f^{LD}, following relation (5). Finally, the compatibility between the two covariance estimations is checked by computing the relative difference component-wise

$$\epsilon_{f,LD} = \frac{|Cov_f^{DAMC} - Cov_f^{LD}|}{|Cov_f^{LD}|}, \tag{7}$$

and the ratio

$$\tau_{f,LD} = \frac{|Cov_f^{DAMC}|}{|Cov_f^{LD}|}. \tag{8}$$

Table 2 shows the maximum relative error and the maximum ratio found among all the covariance components. As can be seen, the distance between the two estimations tends to decrease as the number of considered samples increases. Roughly speaking, the LD method provides the values towards which the DAMC-1 would converge if an infinite number of samples is used.

Table 2. Maximum relative error $\epsilon_{f,LD}$ and maximum ratio $\tau_{f,LD}$ for the 2-body example with respect to the number of variables.

# samples	$\max(\epsilon_{f,LD})$	$\max(\tau_{f,LD})$
10^1	1.148	2.148
10^2	0.2967	1.297
10^3	0.02243	1.022
10^4	0.009701	1.01
10^5	0.006454	0.9935
10^6	0.002108	1.002

In the second part of the test, a DAMC method is exploited again to obtain an estimation of the distribution range. This time the obtained results are compared against the PB method (see Section 3.3). More specifically, the first step involves the propagation of the given initial conditions in the DA framework. Even in this case, a first order approximation is considered. The computed map $\mathcal{M}^1_{x_f}$ is firstly used to perform a DAMC analysis, using an initial uniform distribution. The range of the provided final distribution is evaluated as

$$[\boldsymbol{x}_{f,min}^{DAMC}, \boldsymbol{x}_{f,max}^{DAMC}] = [min_i(\widetilde{\boldsymbol{x}}_f^i), max_i(\widetilde{\boldsymbol{x}}_f^i)]. \tag{9}$$

The range of the propagated uncertainties can also be estimated by using the PB algorithm, which provides upper and lower bounds, $\boldsymbol{x}_{f,UB}^{PB}$ and $\boldsymbol{x}_{f,LB}^{PB}$, without the need to generate samples and evaluate polynomials, but simply using interval

algebra [11]. The consistency of the results is checked by computing the relative error

$$\epsilon_{f,LB} = \frac{|\boldsymbol{x}_{f,min}^{DAMC} - \boldsymbol{x}_{f,LB}^{PB}|}{\boldsymbol{x}_{f,LB}^{PB}} \quad (10)$$

$$\epsilon_{f,UB} = \frac{|\boldsymbol{x}_{f,max}^{DAMC} - \boldsymbol{x}_{f,UB}^{PB}|}{\boldsymbol{x}_{f,UB}^{PB}}, \quad (11)$$

and the ratio

$$\tau_{LB} = \frac{|\boldsymbol{x}_{f,min}^{DAMC}|}{\boldsymbol{x}_{f,LB}^{PB}} \quad (12)$$

$$\tau_{UB} = \frac{|\boldsymbol{x}_{f,max}^{DAMC}|}{\boldsymbol{x}_{f,UB}^{PB}}. \quad (13)$$

Table 3 shows the distance between bounds obtained with DAMC and those given by PB. In particular note how the difference becomes smaller as the number of samples increases.

Table 3. Maximum relative error on bounds with respect to the number of variables for the 2-body example.

# samples	$\max(\epsilon_{f,LB})$	$\max(\epsilon_{f,UB})$
10^1	0.003894	0.09002
10^2	0.001408	0.006865
10^3	2.961e-4	0.004459
10^4	3.433e-5	6.987e-4
10^5	2.37e-6	3.81e-4
10^6	1.749e-6	1.293e-4

5.1.2. Performance and accuracy with respect to the expansion order

The second part of the test is mainly aimed at assessing the performance and the accuracy of the numerical technique with respect to the expansion order. To obtain more appreciable results, the variance on the x and y position component is increased up to 5e-01 km^2. Figure 5 illustrates the maximum error on position and velocity between the DAMC and the pointwise propagation of the same initial set of 10^4 samples. As can be seen, the influence of the order on the accuracy of the map is quite clear, since the error tends to decrease when higher orders are selected. This point can be easily explained by considering that the additional terms of the Taylor expansion allow a better approximation of nonlinear effects, thus producing better results.

The effects of the order can also be appreciated in Figure 6, where the final distributions and the 3-σ covariance ellipsoids obtained using a DAMC propagation method for various orders are depicted.

Although the benefits in the approximation accuracy, the use of higher orders affects also the computational burden required to compute the Taylor expansions. Figure 5 shows the ratio R_τ between the computational time required for a kth-order DA propagation and for its pointwise counterpart. It is evident that higher expansion orders entail higher computational times. Nonetheless, for the present test case, a second order approximation is only one and half times slower than a pointwise propagation. Known this, it is worth clarifying that the computational time, and consequently also the ratio shown in Figure 5, is mainly related with the number of operations required. Thus, not only the order, but also the number of variables considered, the model complexity or, whenever the integration of an ODE is involved, the different numerical integration scheme and demanded accuracy, may have large effects.

5.2. Re-entry problem

For this second example, the Hayabusa re-entry scenario is considered. The set of ODEs used to describe the motion of the vehicle's center of mass is:

$$\begin{cases} \dot{r} &= v\sin\gamma, \\ \dot{\alpha} &= \frac{v\cos\gamma\cos\psi}{r\cos\delta}, \\ \dot{\delta} &= \frac{v\cos\gamma\sin\psi}{r}, \\ \dot{v} &= -\frac{D}{m} - g\sin\gamma, \\ v\dot{\gamma} &= \frac{L\cos\sigma}{m} - g\cos\gamma + \frac{v^2\cos\gamma}{r}, \\ v\dot{\psi} &= \frac{L\sin\sigma}{m\cos\gamma} - \frac{v^2\tan\delta\cos\gamma\cos\psi}{r}, \end{cases} \quad (14)$$

where r and v are vehicle's speed and radius, σ is the bank angle, defined as the angle between the lift vector and the plane described by the local vertical and the velocity vector, and g is the planetary gravitational acceleration. Finally, L and D are the lift and the drag force respectively computed as:

$$L = 0.5\rho S C_L v^2, \quad (15)$$

$$D = 0.5\rho S C_D v^2, \quad (16)$$

in which C_L and C_D are the lift and drag coefficients, ρ, is the atmosphere density, and S is the base area of the capsule. This set of ODEs is written in a local non-inertial reference frame, attached to the vehicle with x axis aligned with the local vertical, y pointing towards east and z to complete the right handed triad. An inertial planetocentric ecliptic reference frame X, Y, Z is assumed; in particular, in the fundamental plane the X axis is collinear to the vernal equinox direction. The relative orientation between the two frames is defined by the right ascension α and the declination δ, whereas the flight path angle γ and the heading angle ψ are introduced to identify the vehicle velocity vector in the local reference frame.

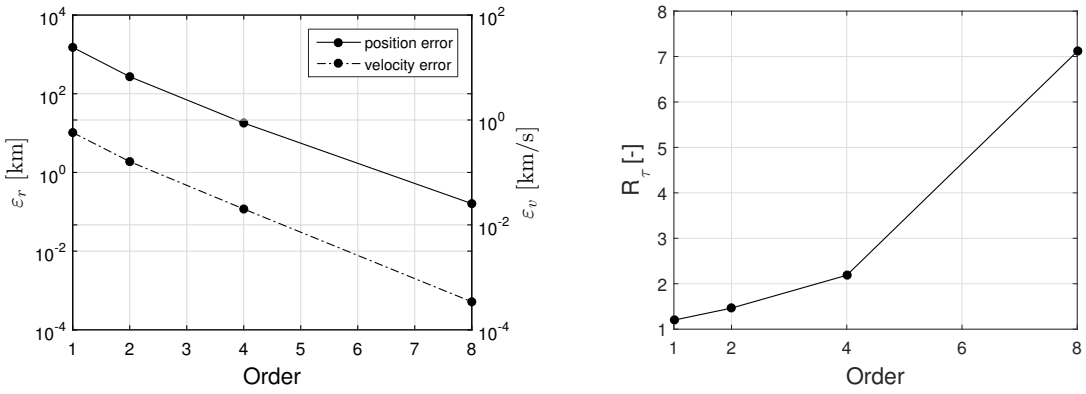

Fig. 5. Position and velocity error (left) and ratio R_τ (right) obtained using DAMC-k (k = 1,2,4,8) for the two-body dynamics.

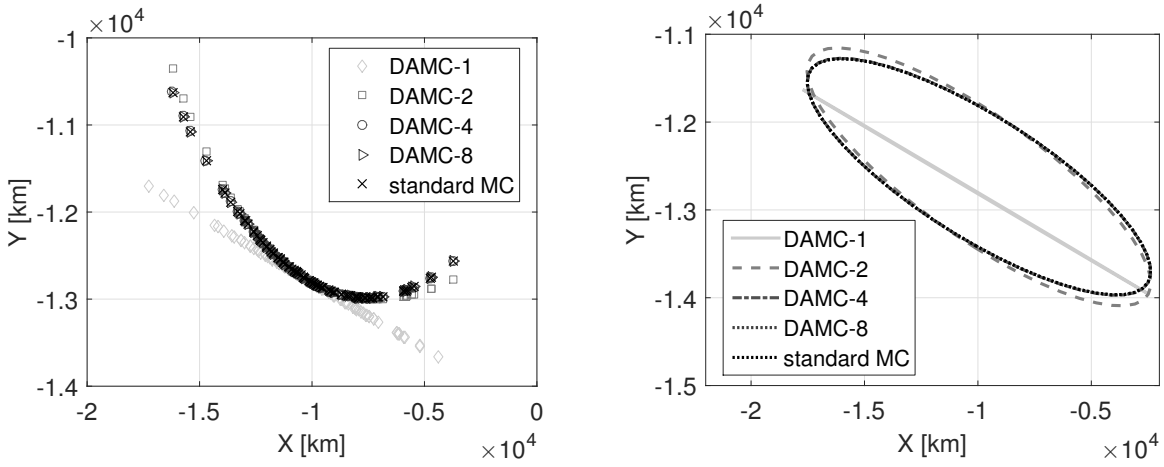

Fig. 6. Final distribution of samples (left) and covariance ellipsoids (right) obtained using DAMC-k (k = 1,2,4,8) for the two-body dynamics.

For the sake of simplicity, an exponential atmospheric model is assumed:

$$\rho = \rho_0 \exp(-\frac{h}{\beta}), \tag{17}$$

where ρ_0 is the atmosphere's density at sea level, h is the object's altitude and β is the scaling altitude.

The initial states, parameters and corresponding uncertainties assumed for the analysis are reported in Tables 4-5.

Table 4. Initial state and uncertainties.

Element	Value	Std. Dev.	Units
h	201.992	0.1085	km
α	-124.28	0.0074	deg
δ	-27.33	0.008	deg
v	12.035	0.002	km/s
γ	-12.35	0.0044	deg/s
ψ	-22.06	0.0119	deg/s

Table 5. Re-entry parameters and uncertainties.

Element	Value	Std. Dev.	Units
C_D	1.30	3.3%	-
C_L	0	-	-
m	18	-	kg
S	0.126	-	m^2
ρ_0	1.217	6.6%	kg/m^3
β	8.5	-	km

Firstly, the Hayabusa nominal trajectory is computed (see Figure 7). The latitude and longitude of the parachute deployment (at 25 km altitude) found with DAST, and listed in Table 6, are close to the values available in literature (see [12]-[13]), where the parachute deployment latitude is included between -30.75 deg and -30.0 deg and the longitude between 135.5 deg E and 136.5 deg E. The small difference in longitude is mainly due to the atmospheric model used in DAST simulations.

Table 6. Re-entry coordinates for Hayabusa at 25 km altitude.

Epoch	2010-Jun-13, 13:53:56.108	UTC
Altitude	25.00000865	km
Longitude	134.3776058	deg
Latitude	-30.12555199	deg

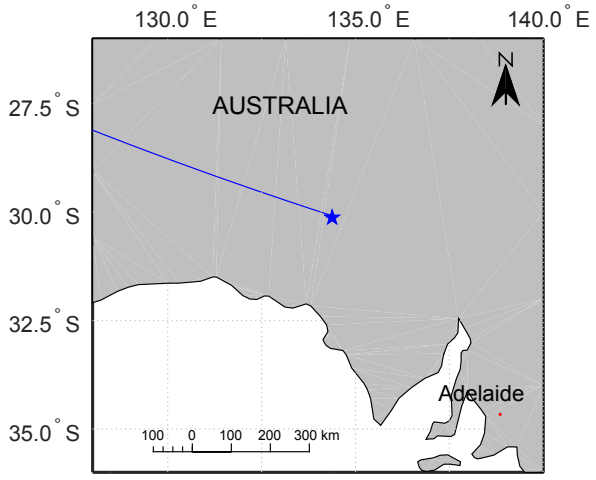

Fig. 7. Reentry site for Hayabusa at 25 km altitude.

Similarly to the two-body example, Figure 9 illustrates the maximum error on position and velocity between the DAMC and the pointwise propagation of the same initial set of samples. As expected both errors decrease as the order increases. However, higher orders mean also higher computational costs, as clearly shown on the right side of the same figure. Ratio τ defined in Eq. (4) is here reported for different orders and different number of samples. Note how the DAMC approach quickly becomes more convenient when the number of samples increases, whereas the computational gain decreases with the expansion order. Furthermore, fixing the number of samples, e.g 10^3, the computational time required by DAMC is orders of magnitude smaller with respect to a classical MC approach.

The DAMC results can be used as a reference to check other high-order propagation methods, i.e. PB and LD. For instance, Figure 8 compares the parachute deployment dispersions obtained with DAMC and PB approaches. In particular, note how the bounds estimation is quite accurate and provides a good approximation of the range for the propagated uncertainties. Table 7 summarizes relative errors and ratios on lower and upper bounds (see Eq. (10)) for different number of samples.

Finally, DAMC and LD methods are compared. The maximum relative errors obtained using Eq. (7) are reported in Table 8:

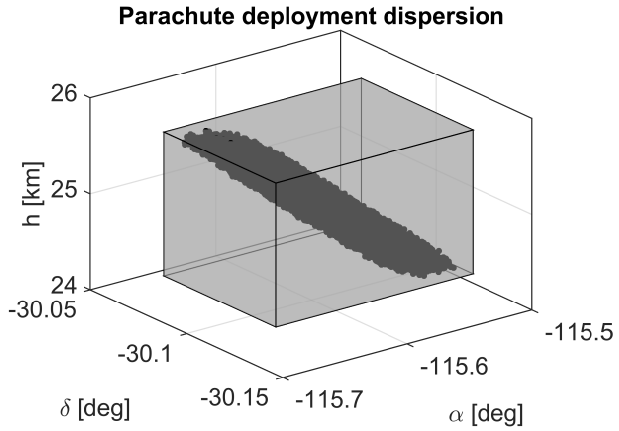

Fig. 8. Comparison between DAMC and PB range estimations at parachute deployment.

Table 7. Maximum relative error on bounds with respect to the number of variables for the re-entry example.

# samples	$\max(\epsilon_{f,LB})$	$\max(\epsilon_{f,UB})$
10^1	8.457e-03	4.104e-03
10^2	3.215e-03	1.979e-03
10^3	1.338e-03	8.025e-04
10^4	7.859e-04	4.434e-04
10^5	3.355e-04	4.657e-04

Table 8. Maximum relative error $\epsilon_{f,LD}$ for the re-entry example with respect to the number of variables.

# samples	$\max(\epsilon_{f,LD})$
10^1	6.613e-01
10^2	3.904e-02
10^3	3.756e-02
10^4	2.025e-02
10^5	5.297e-03

In particular note that the error among the two methods becomes smaller for increasing number of samples, which means a progressive convergence of the DAMC-1 approach to the LD results. The goodness of the results can also be appreciated by looking at Figure 10, where graphical representations of covariance ellipsoids (obtained through LD method) and final distributions (provided by DAMC-1 simulations) are depicted. At the same time it is to be noted that Figure 10 is also aimed to highlight the effects of parameters uncertainty. At this purpose, the final distributions and the covariance ellipsoids of two different simulations are illustrated. In the first

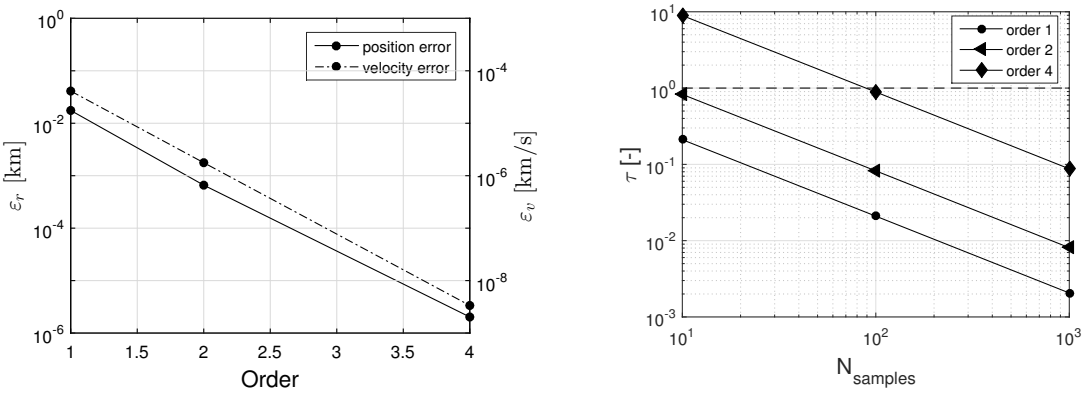

Fig. 9. Position and velocity error (left) and ratio τ (right) obtained using DAMC-k (k = 1,2,4) for the reentry dynamics.

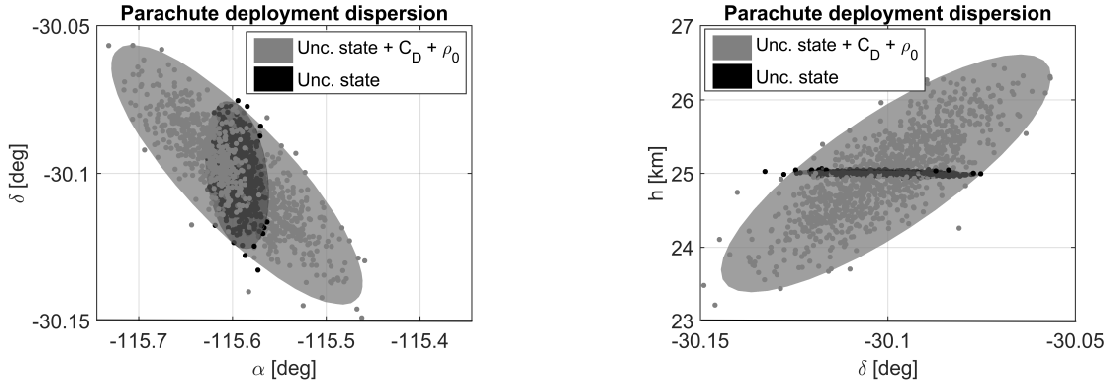

Fig. 10. Uncertainties effects on parachute deployment dispersion in α-δ plane (left) and in h-δ plane (right).

one uncertainty is applied only on the initial state, whereas in the second one also drag coefficient, C_D, and reference density, ρ_0, are perturbed. As shown, when uncertainties on the atmospheric model are considered, the dispersion over right ascension α and declination δ is much larger. Analogous effects are found also in altitude, with a parachute deployment range that goes from 23 km to 27 km, significantly wider than the one obtained in the first simulation.

6. CONCLUSIONS

The main objective of this work concerns the description of the new tool DAST for the nonlinear propagation of uncertainties in space dynamics. Differently from existing linearized models or Monte Carlo techniques, DAST is based on Taylor Differential Algebra, giving access to a wide new class of high-order methods. The architecture of the software is brefly described, with a particular attention to the logical decomposition in three modules: DACE, SF and UPT. The tool is here tested in two different astrodynamics applications. The obtained results show that final accuracy may depend on the size of the considered uncertainty set and on the order of the polynomials. Computational time typically increases for higher orders and larger numbers of uncertain variables. Nonetheless, for typical number of samples, the proposed tool is order of magnitudes more efficient than classical approaches. In addition, it provides an analytical representation of the expansion of the flow, that can be used in ways that are not possible with classical methods. Finally, the modular design used for the implementation of the software greatly simplifies its extension, allowing not only the introduction of new specific space dynamical models, but also the definition of a completely new class of ODEs, thus allowing the application of DAST and relative DA-based uncertainty propagation methods to any field (e.g., biology, finance, etc.).

7. REFERENCES

[1] Martin Berz, Kyoko Makino, Khodr Shamseddine, and Weishi Wan, *Modern map methods in particle beam physics*, vol. 108, Academic Press, 1999.

[2] Roberto Armellin, Pierluigi Di Lizia, Franco Bernelli-Zazzera, and M Berz, "Asteroid close encounters characterization using differential algebra: the case of

apophis," *Celestial Mechanics and Dynamical Astronomy*, vol. 107, no. 4, pp. 451–470, 2010.

[3] Ryan S Park and Daniel J Scheeres, "Nonlinear mapping of gaussian statistics: theory and applications to spacecraft trajectory design," *Journal of guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1367–1375, 2006.

[4] Ryan S Park and Daniel J Scheeres, "Nonlinear semi-analytic methods for trajectory estimation," *Journal of guidance, control, and dynamics*, vol. 30, no. 6, pp. 1668–1676, 2007.

[5] Ahmad Bani Younes, James Turner, Manoranjan Majji, and John Junkins, "High-order uncertainty propagation enabled by computational differentiation," in *Recent Advances in Algorithmic Differentiation*, pp. 251–260. Springer, 2012.

[6] Daniel Giza, Puneet Singla, and Moriba Jah, "An approach for nonlinear uncertainty propagation: Application to orbital mechanics," in *AIAA Guidance, Navigation, and Control Conference, Chicago IL*, 2009, pp. 1–19.

[7] M Valli, R Armellin, P Di Lizia, and MR Lavagna, "Nonlinear mapping of uncertainties in celestial mechanics," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 1, pp. 48–63, 2012.

[8] Manoranjan Majji, John L Junkins, and James D Turner, "A high order method for estimation of dynamic systems," *The Journal of the Astronautical Sciences*, vol. 56, no. 3, pp. 401–440, 2008.

[9] Roberto Armellin, Pierluigi Di Lizia, Martin Berz, and Kyoko Makino, "Computing the critical points of the distance function between two keplerian orbits via rigorous global optimization," *Celestial Mechanics and Dynamical Astronomy*, vol. 107, no. 3, pp. 377–395, 2010.

[10] "SPICE: an observation geometry system for planetary science missions.," `http://naif.jpl.nasa.gov/naif/`, Accessed: 2016-02-10.

[11] Ramon E Moore, R Baker Kearfott, and Michael J Cloud, *Introduction to interval analysis*, Siam, 2009.

[12] Manny E Antimisiaris, James Albers, and Peter Jenniskens, "Hayabusa re-entry: trajectory analysis and observation mission design," 2011.

[13] Nobuaki Ishii, Tetsuya Yamada, Koju Hiraki, and Yoshifumi Inatani, "Reentry motion and aerodynamics of the muses-c sample return capsule," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 51, no. 172, pp. 65–70, 2008.