Z.J. Sun, Y.Z. Luo, P. Di Lizia, F. Bernelli Zazzera
Nonlinear Orbital Uncertainty Propagation with Differential Algebra and Gaussian Mixture Model
Science China. Physics, Mechanics and Astronomy, Vol. 62, N. 3, 034511 (11 pages)
doi:10.1007/s11433-018-9267-6

SCIENCE CHINA
Physics, Mechanics & Astronomy

• Article •
SPECIAL TOPIC:

January 2017 Vol. 60 No. 1: 000000
doi: 10.1007/s11432-016-0037-0

Nonlinear Orbital Uncertainty Propagation with Differential Algebra and Gaussian Mixture Model

Zhen-Jiang Sun[1*], Ya-Zhong Luo[1*], Pierluigi di Lizia[2*], and Franco Bernelli Zazzera[2*]

[1] *College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China;*
[2] *Department of Aerospace Science and Technology, Politecnico di Milano, Milano 20156, Italy*

Received January 11, 2016; accepted April 6, 2016

Nonlinear uncertainty propagation is of critical importance in many application fields of astrodynamics. In this paper, a framework combining the differential algebra technique and the Gaussian mixture model method is presented to accurately propagate the state uncertainty of a nonlinear system. A high-order Taylor expansion of the final state with respect to the initial deviations is firstly computed with the differential algebra technique. Then the initial uncertainty is split to a Gaussian mixture model. With the high-order state transition polynomial, each Gaussian mixture element is propagated to the final time, forming the final Gaussian mixture model. Through this framework, the final Gaussian mixture model can include the effects of high-order terms during propagation and capture the non-Gaussianity of the uncertainty, which enables a precise propagation of probability density. Moreover, the manual derivation and integration of the high-order variational equations is avoided, which makes the method versatile. The method can handle both the application of nonlinear analytical maps on any domain of interest and the propagation of initial uncertainties through the numerical integration of ordinary differential equation. The performance of the resulting tool is assessed on some typical orbital dynamic models, including the analytical Keplerian motion, the numerical J_2 perturbed motion, and a nonlinear relative motion.

Nonlinear orbit, Uncertainty propagation, Differential Algebra, Gaussian Mixture Model, Taylor expansion

PACS number(s): 02.50.Ng, 05.10.Gg, 95.10.Ce, 91.10.Sp

Citation: Sun Z-J, Luo Y-Z, Di Lizia P, et al,
, Sci. China-Phys. Mech. Astron. **60**, 000000 (2017), doi: 10.1007/s11432-016-0037-0

1 Introduction

Uncertainty propagation plays a key role in many application fields of astrodynamics, such as vehicle navigation, orbit determination, target tracking, space situational awareness, as well as collision assessment [1-6]. Moreover, when a complete and accurate statistical description of the propagated statistics is of interest, uncertainty propagation poses the significant challenge of solving partial differential equations to map the probability density functions (PDF) [7-9], or carrying out particle-type, computationally demanding studies such as Monte Carlo simulations [3, 10-12]. Therefore, some approximations are usually introduced in practical applications.

In the last forty years of the 20th century, orbital uncertainty propagation was usually addressed by linearization methods [13-17] or nonlinear Monte Carlo simulations. The linear methods are highly efficient as they linearize the problem, but their accuracy declines in the cases with a highly nonlinear system or long propagation duration. On the other hand, the Monte Carlo simulation provides high-precision results, but is computationally expensive. To avoid these shortcomings, several nonlinear techniques for orbital uncertainty propagation have recently been developed, e.g. differential algebra (DA) [18-20], unscented transformation (UT) [4, 21], state transition tensors (STT) [22-25], polynomial chaos ex-

pansions (PCE) [26-28], as well as Gaussian mixture model (GMM) [29-31].

Among the above nonlinear propagation methods, the DA technique supplies the tools to compute the derivatives of functions within a computerized environment [32]. More specifically, by substituting the classical implementation of real algebra with the implementation of a new algebra of Taylor polynomials, any function f of n variables is expanded into its Taylor polynomial up to an arbitrary order k. In other words, the DA method can automatically give the Taylor series of any sufficiently regular function and can provide the Taylor expansion of the solution flow to a nonlinear system in any high-order without deriving the high-order derivatives manually, so that the method is self-adaptive to different system models. The availability of such order expansions can be exploited when nonlinear uncertainty propagation is performed [18, 19].

For a nonlinear system, the initial Gaussian uncertainty may become non-Gaussian after propagation. However in conventional methods, the uncertainty is usually considered Gaussian throughout the process, and only the mean value and the covariance are computed and provided. To capture the non-Gaussianity of the uncertainty, the GMM method employs a finite sum of weighted Gaussian PDFs to approximate an arbitrary PDF [29-31]. Consequently, the final statistics is described by a set of Gaussian distribution rather than by the overall mean and covariance matrix. Besides, as the elements of the GMM are usually much smaller than the whole uncertainty distribution, the truncation errors during the propagation process can be efficiently decreased.

On one hand, with the nonlinear methods such as DA, UT, STT and PCE, the mean and covariance of the uncertainty can be computed accurately, but the PDF cannot be completely described by the mean and covariance because of the non-Gaussianity. On the other hand, with the GMM method, the non-Gaussianity of uncertainty can be captured, but the propagation of each element still needs certain more accurate method to reduce the corresponding truncation error. Therefore, in order to decrease the truncation errors and capture the non-Gaussianity at the same time, there are some works combining nonlinear methods with the GMM method, such as the UT [33], STT [34], and PCE [35]. In this paper, a framework combining the DA technique with the GMM method is presented. With the help of DA technique, the framework is self-adaptive to any regular nonlinear system in any high-order, without deriving the high-order derivatives manually as in the STT method [34], nor simulating many samples as in the UT or PCE method [33, 35].

The paper is organized as follows. First, a summarized background is reviewed for the DA and GMM methods. Then, a framework combining the DA and GMM is presented, notated as DA_GMM in this work, to accurately propagate and capture the non-Gaussianity of the uncertainty. The performance of the resulting DA_GMM tool is assessed for different types of nonlinear functions and propagated orbits. Finally, conclusions are provided from these results.

2 Differential Algebra Technique

The DA allows the derivatives of any function f of n variables to be computed up to an arbitrary order k, along with the function evaluation [18]. Thus, the Taylor polynomial of the function f up to order k can be obtained. The DA used for the computations in this work was implemented in the software DACE [36].

Consider a generic n-dimensional, sufficiently regular, nonlinear function

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \tag{1}$$

and initialize $[\boldsymbol{x}]$ to be the DA counterpart of the independent variable $\boldsymbol{x}$ around its reference value $\bar{\boldsymbol{x}}$ by adding the perturbation $\delta\boldsymbol{x}$ to $\bar{\boldsymbol{x}}$, i.e.

$$[\boldsymbol{x}] = \bar{\boldsymbol{x}} + \delta\boldsymbol{x} \tag{2}$$

As can be seen from Eq. (2), $[\boldsymbol{x}]$ can already be interpreted as the Taylor expansion of $\boldsymbol{x}$ around the reference value $\bar{\boldsymbol{x}}$.

Substituting $[\boldsymbol{x}]$ into Eq. (1) and carrying out in the differential algebraic framework all the operations involved in the evaluation of $\boldsymbol{f}$ yields

$$\begin{aligned}[\boldsymbol{y}] &= \boldsymbol{f}([\boldsymbol{x}]) = \mathcal{T}_{\boldsymbol{y}}^{k}(\delta\boldsymbol{x}) \\ &= \sum_{p_1+\cdots+p_n\leqslant k} \boldsymbol{c}_{p_1\cdots p_n}\cdot\delta x_1^{p_1}\cdots\delta x_n^{p_n}\end{aligned} \tag{3}$$

where $\mathcal{T}_{\boldsymbol{y}}^{k}$ is the k-th order Taylor expansion of the dependent variable $\boldsymbol{y}$ with respect to $\delta\boldsymbol{x}$, and $\boldsymbol{c}_{p_1\cdots p_n}$ are the Taylor coefficients of $\mathcal{T}_{\boldsymbol{y}}^{k}$:

$$\boldsymbol{c}_{p_1\cdots p_n} = \frac{1}{p_1!\cdots p_n!}\cdot\frac{\partial^{p_1+\cdots+p_n}\boldsymbol{f}}{\partial x_1^{p_1}\cdots\partial x_n^{p_n}} \tag{4}$$

An important application of DA is the automatic computation of the high order Taylor expansion of the solution to an ordinary differential equation (ODE) with respect to either the initial conditions or any parameter of the dynamics [37]. This can be achieved by replacing the classical floating point operations of the numerical integration scheme, including the evaluation of the right hand side, by the corresponding DA-based operations. This way, starting from the DA representation of the initial condition $[\boldsymbol{x}_0]$ as in Eq. (2), the DA-based ODE integration supplies $[\boldsymbol{x}(t_i)]$, which is the Taylor expansion of the solution of the ODE with respect to $\boldsymbol{x}_0$ at all integration times t_i, from the initial epoch t_0 to the

final time t_f. Thus, $\boldsymbol{c}_{p_1\cdots p_n}$ become the terms that relate deviations in the initial conditions $\delta\boldsymbol{x}_0$ to the state at the final time $\boldsymbol{x}_f$. This operator of high-order partials of the state is also referred to as the state transition tensor (STT) in the literature [22, 25]. Additionally, any explicit ODE integration scheme can be adapted to work in the DA framework in a straightforward way.

In uncertainty propagation problems, the first two moments, the mean value and the covariance, are often employed to represent the uncertainty. The DA framework offers an efficient and elegant way for their computation [18]. By exploiting the polynomial expansion Eq. (3), the mean of the i-th component of $\boldsymbol{y}$ can be computed as

$$\mu_i = E\left\{[y_i]\right\} = \sum_{p_1+\cdots+p_n\leqslant k} c_{\mu_i,p_1\cdots p_n} E\left\{\delta x_1^{p_1}\cdots\delta x_n^{p_n}\right\} \tag{5}$$

where $E\{\cdot\}$ is the expectation operator and $c_{i,p_1\cdots p_n}$ are the coefficient of the i-th component of the polynomial map Eq. (3). Similarly, from its definition, each component of the covariance matrix $\boldsymbol{P}$ is computed as

$$\begin{aligned} P_{i,j} &= E\left\{\left([y_i]-\mu_i\right)\left([y_j]-\mu_j\right)\right\} \\ &= \sum_{q_1+\cdots+q_n\leqslant 2k} c_{P_{i,j},q_1\cdots q_n} E\left\{\delta x_1^{q_1}\cdots\delta x_n^{q_n}\right\} \end{aligned} \tag{6}$$

where $c_{P_{i,j},q_1\cdots q_n}$ are the coefficients of the Taylor polynomial resulting from the evaluation of $\left([y_i]-\mu_i\right)\left([y_j]-\mu_j\right)$ in the DA framework. Thus, if Eq. (3) represents the Taylor expansion of the solution of an ODE with respect to the initial conditions, as long as the statistics of the initial state $\boldsymbol{x}$ is determined, the mean value and covariance of the final state $\boldsymbol{y}$ can be obtained.

The advantage of Eqs. (5) and (6) is that they reduce the computation of the mean and covariance of $\boldsymbol{y}$ to the computation of the expectation of the monomials of its Taylor expansion, which can be easily computed using Isserlis' formula [38]. More specifically, the Gaussian hypothesis is widely used in astrodynamics applications to represent the initial distribution of the state variable. Within this assumption, $\boldsymbol{x}$ is a Gaussian random vector (GRV), $\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{P}_0)$, in which $\boldsymbol{\mu}_0$ is its mean vector and $\boldsymbol{P}_0$ is its covariance matrix. If $\boldsymbol{x}$ is initialized as $[\boldsymbol{x}]=\boldsymbol{\mu}_0+\delta\boldsymbol{x}$ in the DA framework to compute $[\boldsymbol{y}]$ in Eq. (3), then the deviation $\delta\boldsymbol{x}$ is a GRV with zero mean and covariance $\boldsymbol{P}_0$. Consequently, given the non-negative integers $s_1,\ldots,s_n$ with $s=s_1+s_2+\cdots+s_n$, the expactation of any monomial can be computed as

$$E\left\{x_1^{s_1}x_2^{s_2}\ldots x_n^{s_n}\right\} = \begin{cases} 0, & \text{if } s \text{ is odd} \\ Haf\left(\boldsymbol{P}\right) & \text{if } s \text{ is even,} \end{cases} \tag{7}$$

where $Haf\left(\boldsymbol{P}\right)$ is the hafnian of $\boldsymbol{P}$, which is defined as [39]

$$Haf\left(\boldsymbol{P}\right) = \sum_{p\in\Omega_s}\prod_{i=1}^{\frac{s}{2}}\sigma_{p_{2i-1},p_{2i}}, \tag{8}$$

and Ω_s is the set of all permutations p of $\{1,2,\ldots,s\}$, satisfying the property $p_1<p_3<p_5<\cdots<p_{s-1}$ and $p_1<p_2, p_3<p_4,\ldots,p_{s-1}<p_s$

3 Gaussian Mixture Model

A GMM is a weighted sum of Gaussian PDFs:

$$p\left(\boldsymbol{x}\right) = \sum_{i=1}^{N}\omega_i p_g\left(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{P}_i\right), \quad \sum_{i=1}^{N}\omega_i = 1, \quad 0<\omega_i\leqslant 1 \tag{9}$$

where N is the number of Gaussian mixture elements (GME), ω_i is the weight associated to the i-th GME

$$p_g\left(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{P}_i\right) = \det\left(2\pi\boldsymbol{P}_i\right)^{-\frac{1}{2}}\exp\left[-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_i\right)^{\mathrm{T}}\boldsymbol{P}_i^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_i\right)\right] \tag{10}$$

For a nonlinear uncertainty propagation problem, the initial Gaussian uncertainty is firstly split into a GMM, and each GME is then propagated through the nonlinear function. Although each element is Gaussian, the weighted sum forms a non-Gaussian approximation of the true distribution.

In this work, a sampling algorithm is employed to split the initial Gaussian uncertainty into a GMM. The detailed principles of the method is referred to [31], and the brief procedure is stated as follows.

(1) For an initial Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{P}_0)$, given a predefined upper bound covariance matrix $\boldsymbol{P}_{\max}$, the covariance matrix $\boldsymbol{P}_i$ of the GMEs is firstly determined.

(1.1) The square-root information matrix $\boldsymbol{R}$ is introduced here to take the place of covariance matrix $\boldsymbol{P}$

$$\boldsymbol{P} = \boldsymbol{R}^{-1}\boldsymbol{R}^{-\mathrm{T}} \tag{11}$$

where $\boldsymbol{R}$ is a $n\times n$ matrix, and can be determined by a Cholesky decomposition of the matrix $\boldsymbol{P}^{-1}$. Thus the upper bounds of the GME covariance, $\boldsymbol{P}_0$ and $\boldsymbol{P}_{\max}$, are transformed to the lower bounds of the square-root information matrix, $\boldsymbol{R}_0$ and $\boldsymbol{R}_{\min}$, i.e.

$$\begin{cases} \boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{R}_i \geqslant \boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{R}_0 \\ \boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{R}_i \geqslant \boldsymbol{R}_{\min}^{\mathrm{T}}\boldsymbol{R}_{\min} \end{cases} \tag{12}$$

(1.2) Compute the singular value decomposition of the matrix $\boldsymbol{R}_0\boldsymbol{R}_{\min}^{-1}$:

$$\boldsymbol{U}_i\boldsymbol{S}_i\boldsymbol{V}_i^{\mathrm{T}} = \boldsymbol{R}_0\boldsymbol{R}_{\min}^{-1} \tag{13}$$

where $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ are orthonormal matrices and $\boldsymbol{S}_i = \mathrm{diag}\left(\sigma_{\mathrm{i1}},\ldots,\sigma_{\mathrm{in}}\right)$ is a diagonal matrix with the n positive singular values $\sigma_{i1},\ldots,\sigma_{in}$ on its diagonal.

(1.3) If $\sigma_{ik} \geqslant 1$, for all $k = 1, \ldots, n$, then the choice for $\boldsymbol{R}_i$ is just $\boldsymbol{R}_i = \boldsymbol{R}_0$. Otherwise, one forms the $n \times n$ diagonal matrix

$$\delta \boldsymbol{S}_{i\_\text{full}} = \begin{bmatrix} \sqrt{\max\left(1-\sigma_{i1}^2, 0\right)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\max\left(1-\sigma_{in}^2, 0\right)} \end{bmatrix} \tag{14}$$

Next, one deletes all of the zero-valued rows of $\delta \boldsymbol{S}_{i\_\text{full}}$ to form the matrix $\delta \boldsymbol{S}_i$. That is, row k of $\delta \boldsymbol{S}_{i\_\text{full}}$ is deleted for every k such that $\sigma_{ik} \geqslant 1$. This latter matrix is then used to form the matrix

$$\delta \boldsymbol{R}_i = \delta \boldsymbol{S}_i \boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{R}_{\min} \tag{15}$$

(1.4) One uses QR decomposition to compute $\boldsymbol{R}_i$ as follows:

$$\boldsymbol{Q}_i \begin{bmatrix} \boldsymbol{R}_i \\ 0 \end{bmatrix} = \begin{bmatrix} \delta \boldsymbol{R}_i \\ \boldsymbol{R}_0 \end{bmatrix} \tag{16}$$

Thus, with Eqs. (11) to (16), the GME covariance $\boldsymbol{P}_i$ is determined.

$$\boldsymbol{P}_0, \boldsymbol{P}_{\max} \mapsto \boldsymbol{P}_i \tag{17}$$

(2) Then, the mean vector $\boldsymbol{\mu}_i$ of the i-th GME is computed.

(2.1) The covariance matrix decrement can be calculated as

$$\delta \boldsymbol{P}_i = \boldsymbol{P}_0 - \boldsymbol{P}_i \tag{18}$$

(2.2) Subsequently, the mean vector $\boldsymbol{\mu}_i$ of the i-th GME is randomly selected according to the Gaussian distribution

$$\boldsymbol{\mu}_i \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \delta \boldsymbol{P}_i\right) \tag{19}$$

(2.3) The step (2.2) is repeated for the overall number N of GMEs to be generated.

(3) In the resulting GMM, all GMEs are assigned the same weight

$$\omega_i = \frac{1}{N}. \tag{20}$$

It has been proved in theory that the resulting GMM converges to the original Gaussian distribution as the sampling number increases [31]. Moreover, as the mean vectors of the GMEs are selected randomly, the non-Gaussianity of the final uncertainty along any direction can be captured.

4 Framework Combining Differential Algebra and Gaussian Mixture Model

As there is no guarantee that the state uncertainty will remain Gaussian after a nonlinear propagation, the first two moments are not able to completely describe the statistics of the final state. In the DA method, the Taylor expansion order needed in computing higher moments is times higher than the order needed in computing mean value, e.g. if the Taylor expansion of the mean value is in kth order, that of the covariance should be in $2k$th order, that of the third moment should be in $3k$th order, and so on. Thus if we want a mean value of very high order, it is not efficient to compute higher moments as the number of the high-order terms in a Taylor expansion increases exponentially. Combining with the GMM is a good choice for the DA, because the GMM makes it possible to capture the non-Gaussianity of the uncertainty without computing higher moments.

In the GMM method, an uncertainty propagation method through the nonlinear function is naturally needed. The accuracy of the method used in propagating every GME is significant for the accuracy of the whole GMM method. Combining with the DA is also a good choice for the GMM, because the DA automatically supplies the Taylor expansion of the solution or flow up to any order k, without manually deriving the high-order derivatives of the nonlinear function or ODE. Thus the DA technique is generic and self-adaptive to the nonlinear function or the right side of the ODE.

In this work, a framework combining the DA technique with the GMM method is presented for uncertainty propagation. The high order Taylor expansion of the final state with respect to initial state deviation is computed with the DA technique. And the initial uncertainty is split into a GMM with the sampling algorithm. By using the high order Taylor expansion, every initial GME is respectively propagated, forming a final GMM that approximates to the final uncertainty. The presented framework works as the following flow in detail.

(1) Given the initial reference point $\bar{\boldsymbol{x}}(t_0)$, define the initial state as a DA variable, as in Eq. (2)

$$[\boldsymbol{x}(t_0)] = \bar{\boldsymbol{x}}(t_0) + \delta \boldsymbol{x}(t_0) \tag{21}$$

(2) Substituting Eq. (21) into the nonlinear function or ODE with DA technique, the result is the kth-order Taylor expansion of the final state $\boldsymbol{x}\left(t_f\right)$ with respect to the initial deviation, as in Eq. (3)

$$\left[\boldsymbol{x}\left(t_f\right)\right] = \mathcal{T}^k_{\boldsymbol{x}(t_f)}\left(\delta \boldsymbol{x}(t_0)\right) \tag{22}$$

(3) Given the initial state distribution $\boldsymbol{\mu}_0$ and $\boldsymbol{P}_0$, split it into a GMM of N elements, wherein the weight, mean value and covariance of each GME are ω_i, $\boldsymbol{\mu}_i(t_0)$ and $\boldsymbol{P}_i(t_0)$.

(4) Propagate every GME with the Taylor expansion in Eq. (22). The detailed steps of this procedure are as follows.

(4.1) Initialize the initial state deviation $\delta\boldsymbol{x}(t_0)$ as a deviation from the mean value of a GME $\boldsymbol{\mu}_i(t_0)$.

$$[\delta\boldsymbol{x}(t_0)] = \boldsymbol{\mu}_i(t_0) - \bar{\boldsymbol{x}}(t_0) + \delta\boldsymbol{x}^i(t_0) \tag{23}$$

(4.2) Substituting Eq. (23) into Eq. (22), the result is the kth-order Taylor expansion of the final state $\boldsymbol{x}\left(t_f\right)$ with respect to the initial deviation from the mean value of the i-th GME

$$\left[\boldsymbol{x}\left(t_f\right)\right] = \mathcal{T}^k_{\boldsymbol{x}(t_f)}\left(\delta\boldsymbol{x}^i(t_0)\right) \tag{24}$$

(4.3) Compute the final mean value and covariance of the ith GME by using Eqs. (5) and (6).

$$\begin{aligned}\boldsymbol{\mu}_i\left(t_f\right) &= E\left\{\left[\boldsymbol{x}\left(t_f\right)\right]\right\}\\ &= \sum_{p_1+\cdots+p_n\leqslant k} \boldsymbol{c}_{\boldsymbol{\mu}_i(t_f),p_1\ldots p_n} E\left\{\left(\delta x_1^i(t_0)\right)^{p_1}\cdots\left(\delta x_n^i(t_0)\right)^{p_n}\right\}\end{aligned} \tag{25}$$

$$\begin{aligned}\boldsymbol{P}_i\left(t_f\right) &= E\left\{\left(\left[\boldsymbol{x}\left(t_f\right)\right]-\boldsymbol{\mu}_i\left(t_f\right)\right)\left(\left[\boldsymbol{x}\left(t_f\right)\right]-\boldsymbol{\mu}_i\left(t_f\right)\right)^{\mathrm{T}}\right\}\\ &= \sum_{q_1+\cdots+q_n\leqslant 2k} \boldsymbol{c}_{\boldsymbol{P}_i(t_f),q_1\ldots q_n} E\left\{\left(\delta x_1^i(t_0)\right)^{q_1}\cdots\left(\delta x_n^i(t_0)\right)^{q_n}\right\}\end{aligned} \tag{26}$$

wherein the higher moments of the deviation $\delta\boldsymbol{x}^i(t_0)$ can be calculated by substituting its covariance $\boldsymbol{P}_i(t_0)$ into Eq. (7).

(5) Finally, after the final mean value $\boldsymbol{\mu}_i\left(t_f\right)$ and covariance $\boldsymbol{P}_i\left(t_f\right)$ of every GME are computed, the GMM for the final state distribution is obtained.

$$\begin{cases} p\left(\boldsymbol{x}\left(t_f\right)\right) = \sum\limits_{i=1}^{N} \omega_i p_g\left(\boldsymbol{x}\left(t_f\right);\boldsymbol{\mu}_i\left(t_f\right),\boldsymbol{P}_i\left(t_f\right)\right)\\ \sum\limits_{i=1}^{N}\omega_i = 1 \quad 0<\omega_i\leqslant 1\end{cases} \tag{27}$$

Additionally, it is noticed that in the presented framework, the Taylor expansion of the final state with respect to the initial deviation, i.e. Eq. (22), is computed only once. All the following computations use the same Taylor polynomials. In other words, in the case of integrating an ODE, which is computationally expensive, increasing the number of GME will only add some efficient polynomial computation, without affecting the computation time so much.

5 Simulation Results and Discussion

The framework developed thus far is tested and applied to the uncertainty propagation problems with some typical orbital dynamic models. Three test cases are presented in this section: the first case is an analytical Keplerian orbit, i.e. analytical solution to two-body dynamics; the second case is subject to a numerical ODE model of a perturbed two-body dynamics; and the last case is subject to a nonlinear ODE model of relative motion with a circular chief orbit. As previously stated, the Taylor polynomials in the DA technique can be automatically extended to any given order k without deriving the high-order derivatives manually. In the following cases the presented framework is tested for orders 1 to 4, i.e. k is changed from 1 to 4.

5.1 Analytical Keplerian Orbit

The Keplerian motion can be analytically solved by iteration [40]. Given the initial Cartesian state $\boldsymbol{x}(t_0)$ of a satellite, the initial classical orbital elements are firstly determined and the Kepler's Equation is solved iteratively. Then, the final Cartesian state $\boldsymbol{x}\left(t_f\right)$ of the satellite is computed with the F and G solutions. Thus the final state $\boldsymbol{x}\left(t_f\right)$ of a satellite can be formulated as a nonlinear function of the propagation duration Δt and the initial state $\boldsymbol{x}(t_0)$ in the Earth-centered inertial (ECI) reference frame.

$$\boldsymbol{x}\left(t_f\right) = \boldsymbol{f}\left(\Delta t, \boldsymbol{x}(t_0)\right) \tag{28}$$

To determine the initial Cartesian state of the satellite, the initial classical orbital elements are listed in Table 1, wherein the inclination is selected approximately to 0, such that the x-y plane is nearly the orbital plane. Besides, the initial state uncertainty expressed in the ECI frame is also found in Table 1, including the standard deviations of the position uncertainty σ_p and the velocity uncertainty σ_v. In this example, the initial uncertainty is assumed isotropic, i.e. $\sigma_x = \sigma_y = \sigma_z = \sigma_p$ and $\sigma_{v_x} = \sigma_{v_y} = \sigma_{v_z} = \sigma_v$, and maybe quite larger than practical orbit determination error, only for testing the performance of the framework. The uncertainty propagation time is selected as $\Delta t = 10T$, where T means the orbital period of the satellite.

Table 1 Initial conditions for a Keplerian orbit

a (km)	e (rad)	i (rad)	Ω (rad)	ω (rad)	f (rad)	σ_p (m)	σ_v (m/s)
6778.138	0.2	0.001	0	0	0	100	10

Without Gaussian mixture model, the mean vector and covariance matrix of the orbital state uncertainty are propagated with the DA technique, as in [18]. Fig. 1 shows the final mean value points and 3σ covariance ellipses in the x-y plane. Since the initial argument of latitude is 0, herein the x-axis represents the radial direction and the y-axis represents the tangential direction. The result of a Monte Carlo simulation is also plotted, represented by "MC" in Fig. 1, wherein 10^5 samples are employed. The 1-order result is equivalent to the result of the conventional linearization method. By increasing the order up to 4, the mean vector and covariance matrix finally approximate to the result of the Monte Carlo simulation. It indicates that the DA technique can handle the overall mean vector and covariance matrix very well. However, it is also noted that the Monte Carlo samples distribute along the Keplerian orbit and the overall distribution is non-Gaussian [14]. If the probability density of the final uncertainty is of interest, it cannot be accurately obtained only with the overall mean vector and covariance matrix.

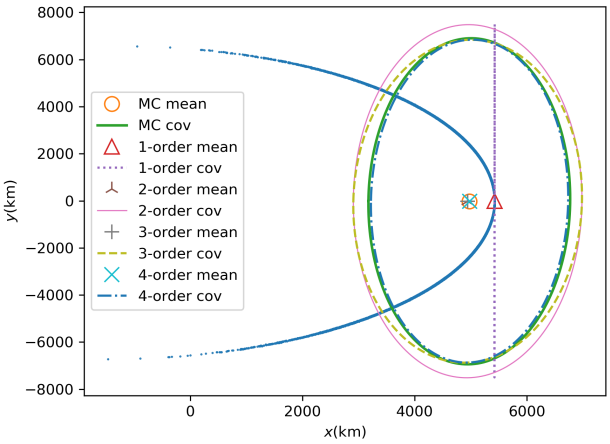


Figure 1 Final uncertainty of the Keplerian orbit by DA

With the presented DA_GMM framework, a GMM is propagated to describe PDF of the final orbital state uncertainty. Herein the initial uncertainty is split into a GMM with $N = 5000$ GMEs in total, and the upper-bound of the GME covariance matrix $\boldsymbol{P}_{\max}$ is defined as

$$\boldsymbol{P}_{\max} = \frac{\boldsymbol{P}_0}{s^2} \tag{29}$$

In this example, the splitting scale is selected as $s = 20$, i.e. the standard deviation of each initial GME is no larger than 0.05 time the size of the initial uncertainty. Fig. 2 shows the final Monte Carlo samples and 3σ covariance ellipses of the GMEs.

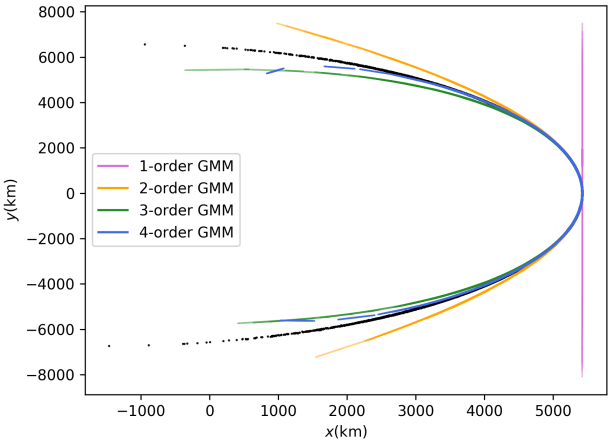
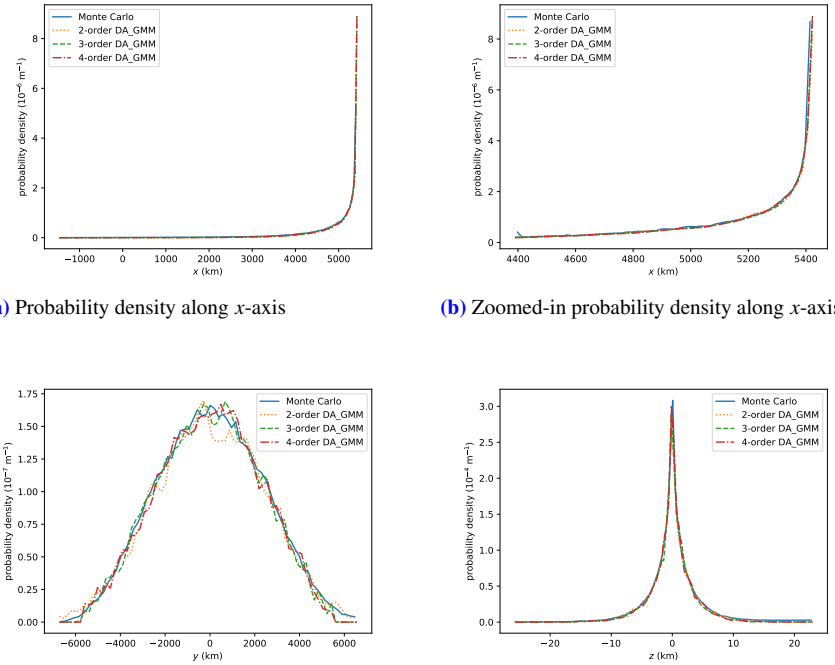


Figure 2 Final uncertainty of the Keplerian orbit by DA_GMM

As all the GMEs are respectively propagated with the same Talor expansion with repect to the initial reference point, the distribution of the final 1-order DA_GMM in Fig. 2 is equivalent to the linearized covariance ellipse in Fig. 1. By increasing the order from 2 to 4, the distribution of final GMMs approximates to the Monte Carlo samples. The non-Gaussianity of final uncertainty is generally captured by the high-order DA_GMM. Though there still remains a little incompatibility at the tail of the high-order DA_GMM, it affects little to the accuracy of the final PDF since the probability density at the tail is very low, as illustrated in the following Fig. 3. Of course, the remaining incompatibility can be further eliminated with a higher-order Taylor expansion. However, considering the computation cost and the expected PDF accuracy, it is not recommended to propagate with a too high order, though it is automatic in the DA framework. Besides, it may be noticed that there is a little discontinuous jump of the GMM at the tail. Since the initial GMM is produced by a sampling algorithm, there are only a few GMEs at the tail which has very low PDF, maybe making the PDF uneven in this region. Of course, this can be overcome by adding the total number of GMEs in the GMM according to the accuracy requirement.

The PDFs along axes are plotted and compared in Fig. 3. For a GMM, the PDF value at a point can be calculated by a weighted sum of the PDF values of all the GMEs. For a Monte Carlo result, the PDF value at a point is obtained by counting the number of samples in a small interval. It is found that even the probability density with 2-order DA_GMM framework agrees with the Monte Carlo simulation result well, though the incompatibility at the tail of the GMM distribution is clear as in Fig. 2. The non-Gaussianity is mainly displayed on x- and z- axes, i.e. the radial and nor-

mal directions in this case. Along the radial direction, i.e. the x-axis, the reference state has the maximum radial distance, hence the probability density mainly focus at the right end. Along the normal direction, i.e. the z-axis, the reference state is inside the orbital plane, and the PDF has an apex for $z = 0$.



(a) Probability density along x-axis

(b) Zoomed-in probability density along x-axis

(c) Probability density along y-axis

(d) Probability density along z-axis

Figure 3 Final probability density of the Keplerian orbit

5.2 Numerical Perturbed Two-Body Orbit

The second test case is a numerical two-body orbit considering the classical J_2 perturbation. The motion is subject to the following classical ODE model [40].

$$\begin{cases} \dot{x} = v_x, \dot{y} = v_y, \dot{z} = v_z \\ \dot{v}_x = -\frac{\mu}{r^3}x - \frac{3\mu J_2 R_e^2}{2r^5}\left(1 - \frac{5z^2}{r^2}\right)x \\ \dot{v}_y = -\frac{\mu}{r^3}y - \frac{3\mu J_2 R_e^2}{2r^5}\left(1 - \frac{5z^2}{r^2}\right)y \\ \dot{v}_z = -\frac{\mu}{r^3}z - \frac{3\mu J_2 R_e^2}{2r^5}\left(3 - \frac{5z^2}{r^2}\right)z \end{cases} \tag{30}$$

wherein $\boldsymbol{r} = [x, y, z]^{\mathrm{T}}$ and $\boldsymbol{v} = \left[v_x, v_y, v_z\right]^{\mathrm{T}}$ represent the position and velocity vectors in the ECI frame, μ and R_e represent the gravity constant and equator radius of the central body. The initial conditions remain the same with that in the first test case, as in Table 1. Other conditions are also unchanged unless otherwise specified.

Fig. 4 shows the propagated results in x-y plane after 5 orbital periods of the 4-order DA_GMM framework and Monte Carlo simulation, with respectively 5000 GMEs and 10^5 samples. Compared with the results on Keplerian orbit, the final uncertainty ceases to be symmetric about the radial direction because of the J_2 perturbation. The non-Gaussianity of the final uncertainty is also well captured, indicating that a nonlinear system subject to either a nonlinear function or a nonlinear ODE can be handled by the presented DA_GMM framework.

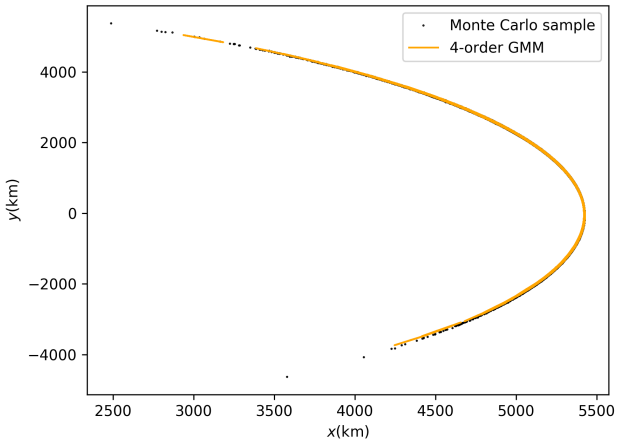
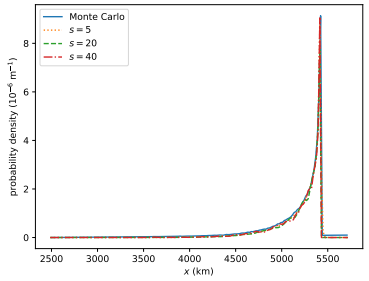
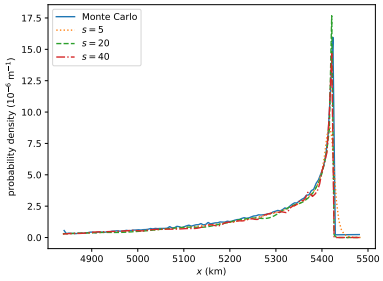


Figure 4 Final uncertainty of the J_2 perturbed two-body orbit

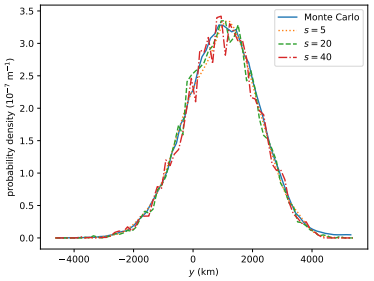
With different initial uncertainty splitting scales in Eq. (29), $s = 5, 20, 40$ respectively, the PDFs of the DA_GMM framework and Monte Carlo simulation are plotted and compared in Fig. 5. Generally, the PDFs with different splitting scales agree with the PDF of Monte Carlo simulation well. For the PDF with the splitting scale $s = 5$, the initial GMEs are largest among the three cases. On one hand, the truncation errors of GMEs during propagation is the largest, leading to the PDF not so accurate at some areas, e.g. at the right end of x-axis in Fig. 5b. On the other hand, as the initial uncertainty splitting method in this work is a random sampling algorithm, the overall PDF of the GMM formed by these GMEs is smoother than the other two cases, which is especially obvious in Fig. 5c. In contrast, for the PDF with the splitting scale $s = 40$, the truncation error of each GME is smaller but the overall PDF is more uneven than the other two cases. In a word, the value of splitting scale s should be determined according to the non-Gaussianity of the system. In a case with high non-Gaussianity, a larger splitting scale s is preferred to get a more accurate PDF, while in a case with not so high non-Gaussianity, a relatively smaller splitting scale s is preferred to make the PDF more even.



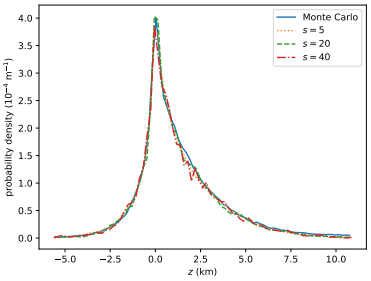
(a) Probability density along x-axis



(b) Zoomed-in probability density along x-axis



(c) Probability density along y-axis



(d) Probability density along z-axis

Figure 5 Final probability density of the J_2 perturbed orbit

For a nonlinear system subject to an ODE dynamic model, numerical integration is required to propagate the system state, thus the computation efficiency is an important aspect of an uncertainty propagation method. Table 2 compares the computation time cost of the presented DA_GMM framework with different orders and the Monte Carlo simulation, running on a quad-core and eight-thread Intel Core i7-6700HQ CPU. Specifically, the Monte Carlo simulation is

parallelized, while the DA_GMM framework is serial. For the DA_GMM framework, increasing the order improves the propagation precision and costs much more computation, because the number of terms in a Taylor expansion increases exponentially as the order increases. On the whole, the computation efficiency of the presented DA_GMM framework is much higher than the efficiency of Monte Carlo simulation, while the final PDF is a good approximation to the Monte Carlo result with high accuracy.

Table 2 Computation cost for a J_2 perturbed two-body orbit

Method	Time cost (sec)
DA_GMM (1-order, 5000 GMEs)	19
DA_GMM (2-order, 5000 GMEs)	26
DA_GMM (3-order, 5000 GMEs)	37
DA_GMM (4-order, 5000 GMEs)	62
Parallel Monte Carlo (10^5 samples)	270×8

5.3 Nonlinear Relative Motion with Circular Chief Orbit

To demonstrate the versatility of the presented DA_GMM framework, the uncertainty propation of nonlinear relative motion with a circular chief orbit is tested in this section. Considering two spacecraft, one is termed *chief* and the other one is referred to as *deputy*. The motion of the deputy can be expressed in the local-vertical local-horizontal (LVLH) reference frame of the chief, wherein the *Origin* is located at the center of mass of the chief; the x-axis is along the radial direction; the z-axis is along the angular momentum direction; and the y-axis completes the right-handed system. If the chief follows a circular orbit, usually the x, y, z-axes are respectively called R-bar, V-bar and H-bar, and the unperturbed relative motion is subject to the following ODE model [41].

$$\begin{cases} \ddot{x} = 2n\dot{y} + n^2 x - \dfrac{\mu(a+x)}{\left[(a+x)^2+y^2+z^2\right]^{\frac{3}{2}}} + \dfrac{\mu}{a^2} \\ \ddot{y} = -2n\dot{x} + n^2 y - \dfrac{\mu y}{\left[(a+x)^2+y^2+z^2\right]^{\frac{3}{2}}} \\ \ddot{z} = -\dfrac{\mu z}{\left[(a+x)^2+y^2+z^2\right]^{\frac{3}{2}}} \end{cases} \tag{31}$$

wherein a represents the radius of chief orbit, n represents the orbital angular velocity of the target. The initial conditions are found in Table 3. There is only an initial relative velocity along V-bar, hence the deputy drifts along the $-$V-bar and the phase difference between the two spacecraft increases.

Table 3 Initial conditions for a nonlinear relation motion

a (km)	x (m)	y (m)	z (m)	v_x (m/s)	v_y (m/s)	v_z (m/s)	σ_p (m)	σ_v (m/s)
6778.138	0	0	0	0	15	0	5	0.5

Fig. 6 shows the propagated results in orbital plane after 40 orbital periods of the 4-order DA_GMM framework and Monte Carlo simulation. It can be seen that the reference relative state has drifted to about 6700 km behind the chief on $-$V-bar and more than 6000 km below the chief on $-$R-bar, meaning that the phase difference between the two spacecraft reaches nearly $\frac{\pi}{2}$. The non-Gaussianity of final uncertainty is also well captured.

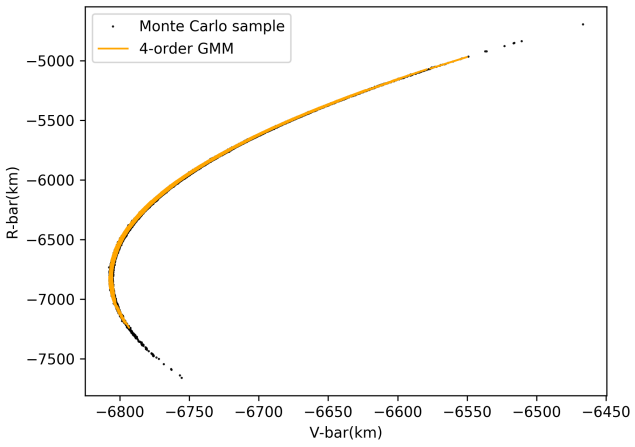
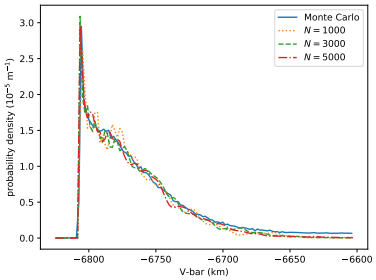


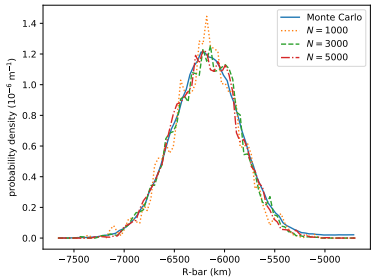
Figure 6 Final uncertainty of the nonlinear relative motion

With different total GME numbers in the GMMs, N =

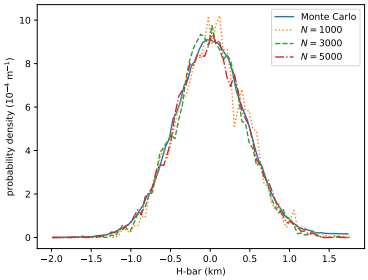
1000, 3000, 5000 respectively, the PDFs of the DA_GMM framework and Monte Carlo simulation are plotted and compared. Since the initial uncertainty splitting algorithm has some random, there is some vibration on the PDF of final GMM. Obviously in Fig. 7, though all the three PDFs of DA_GMM framework are approximate to the Monte Carlo result, the PDF with more total GMEs has a smaller amplitude of vibration.



(a) Probability density along V-bar



(b) Probability density along R-bar



(c) Probability density along H-bar

Figure 7 Final probability density of the nonlinear relative motion

The computation cost of DA_GMM framework with different orders and GME numbers are compared to the Monte Carlo simulation, as in Fig. 8. The computation speed of DA_GMM framework is much higher than the speed of Monte Carlo simulation. In the DA_GMM framework, all the GMEs are propagated with the same high-order Taylor expansion with respect to the initial reference state. Most of the time is spent on calculating the high-order state transition Tayloy polynomial, and then only some efficient polynomial computation is performed to obtain the final GMEs. Therefore, increasing the order would significantly cost more time, while only a little additional time is required for increasing the GME number. For example in Fig. 8, by increasing the order from 2 to 4, the computation time increases from less than 30 sec to more than 50 sec; while by increasing the GME number from 1000 to 5000, the computation time only increases by several seconds.

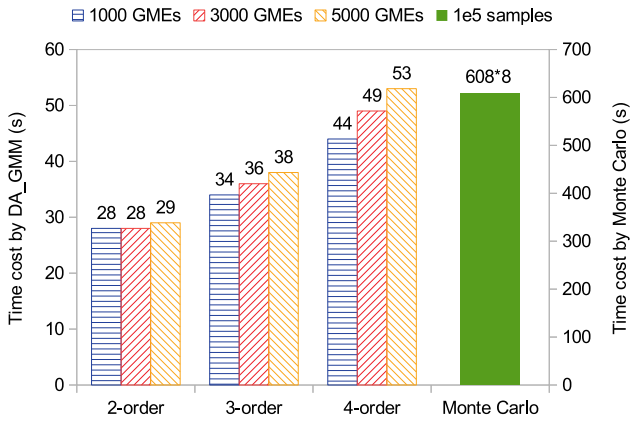


Figure 8 Computation cost for a nonlinear relative motion

5.4 Discussion

According to the above three examples, for a nonlinear system subject to either a nonlinear function or a nonlinear ODE, the non-Gaussianity of final state uncertainty is well cap-

tured and the PDF of final GMM is a good approximation to the Monte Carlo simulation, meanwhile the computation efficiency of the DA_GMM framework is much higher than the Monte Carlo method. In this framework, the high-order state transition Taylor polynomial with respect to the initial reference state is computed only once with the DA technique, and then all the final GMEs are calculated based on the same high-order polynomial. Thus, the computation efficiency is much higher than the Monte Carlo simulation, and most of the time is spent on computing the high-order state transition polynomial. However, as the truncation error of a Taylor polynomial becomes large at a point far from the reference point, there may be some incompatibility between the DA_GMM framework and Monte Carlo simulation at the tail of final GMM distribution. Of course, this incompatibility can be further eliminated with a even higher-order Taylor expansion. But in many cases, the PDF value in the tail region is very low, and there is no need to propagate with a too high order, considering the computation cost and the expected PDF accuracy.

Moreover, since a random sampling algorithm is performed when splitting the initial uncertainty, there is some vibration on the PDF of final GMM. With smaller initial GMEs, though the trancation error of each final GME is smaller, the overall PDF would be more uneven. This can be overcome by increasing the total number of GMEs, so that the amplitude of vibration of the PDF would decrease. Besides, since most of the computation time is spent on the high-order state transition polynomial, increasing the total GME number does not require accordingly so much additional computation cost.

6 Conclusion

In this paper, the problem of nonlinear uncertainty propagation is addressed. More specifically, a framework combining the differential algebra technique and Gaussian mixture model method is presented to propagate the initial uncertainty of a nonlinear system, which is notated as DA_GMM in this work. It is demonstrated that the DA_GMM framework enables a general approach to uncertainty propagation, which can include the effects of high-order nonlinear terms during propagation and capture the non-Gaussianity of final uncertainty. The obtained probability density function is a good approximation to that of Monte Carlo simulation, while the computing efficiency is significantly higher. Another advantage of the presented DA_GMM framework is that the propagation order can be automatically extended by the difference algebra technique, and the framework does not require manually writing and integrating the high-order variational equations. This makes the framework versatile. A nonlinear system that subject to either a state transition function or an ordinary differential equation can be handled. The performance of the presented framework is assessed on some typical orbital dynamic models, including the analytical two-body Keplerian motion, the numerical J_2 perturbed two-body orbit, and a nonlinear relative motion dynamic model.

This work was supported by the National Natural Science Foundation of China (Grant No. 11572345).

Conflict of interest The authors declare that there are no conflicts of interest to this work.

1 Y. Z. Luo, and Z. Yang, Prog. Aerosp. Sci. **89**, 23 (2017).
2 J. R. Chen, J. F. Li, X. J. Wang, J. Zhu, and D. N. Wang, Sci. China-Phys. Mech. Astron. **61**, 024511 (2018).
3 S. G. Hesar, D. J. Scheeres, and J. W. McMahon, J. Guid. Control Dynam. **40**, 81 (2017).
4 C. Sabol, K. Hill, K. Alfriend, and T. Sukut, Acta Astronaut. **84**, 69 (2013).
5 Y. Z. Luo, Z. Yang, and H. N. Li, Sci. China-Phys. Mech. Astron. **57**, 731 (2014).
6 T. H. Xu, K. F. He, and G. C. Xu, Sci. China-Phys. Mech. Astron. **55**, 738 (2012).
7 A. T. Fuller, Int. J. Control, **9**, 603 (1969).
8 R. M. Weisman, M. Majji, and K. T. Alfriend, Celest. Mech. Dyn. Astr. **118**, 165 (2014).
9 I. Park, K. Fujimoto, and D. J. Scheeres, J. Guid. Control Dynam. **38**, 2287 (2015).
10 R. Ghrist and D. Plakalovic, "Impact of non-Gaussian error volumes on conjunction assessment risk analysis", AIAA Paper No. 2012–4965, 2012.
11 N. Arora, V. Vittaldev, and R. P. Russell, J. Guid. Control Dynam. **38**, 1345 (2015).
12 K. Liu, B. Jia, G. Chen, K. Pham, and Erik Blasch, "A real-time orbit satellites uncertainty propagation and visualization system using graphics computing unit and multi-threading processing", in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, IEEE No. 8A2–1, 2015.
13 S. Lee, H. Lyu, and I. Hwang, J. Guid. Control Dynam. **39**, 1593 (2016).
14 J. L. Junkins, M. R. Akella, and K. T. Alfriend, J. Astronaut. Sci. **44**, 541 (1996).
15 D. K. Geller, J. Guid. Control Dynam. **29**, 1404 (2006).
16 F. L. Markley, and J. R. Carpenter, J. Astronaut. Sci. **57**, 233 (2009).
17 D. A. Vallado, "Covariance transformations for satellite flight dynamics operations", in *AIAA/AAS Astrodynamics Specialist Conference*, Big Sky, Montana, AAS-03-526, 2003.
18 M. Valli, R. Armellin, P. Di Lizia, and M. R. Lavagna, J. Guid. Control Dynam. **36**, 48 (2013).
19 A. Wittig, P. Di Lizia, R. Armellin, K. Makino, F. Bernelli, and M. Berz, Celest. Mech. Dyn. Astr. **122**, 239 (2015).
20 R. Armellin, and P. Di Lizia, J. Guid. Control Dynam. **41**, 101 (2018).
21 S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, IEEE T. Automat. Contr. **45**, 477 (2000).
22 K. Fujimoto, D. J. Scheeres, and K. T. Alfriend, J. Guid. Control Dynam. **35**, 497 (2012).
23 E. Pellegrini, and R. P. Russell, J. Guid. Control Dynam. **39**, 2485 (2016).
24 Z. Yang, Y. Z. Luo, J. Zhang, and G. J. Tang, J. Guid. Control Dynam. **39**, 2170 (2016).
25 I. Park, and D. J. Scheeres, J. Guid. Control Dynam. **41**, 240 (2018).
26 B. A. Jones, A. Doostan, and G. H. Born, J. Guid. Control Dynam. **36**, 430 (2013).

27 S. Oladyshkin, and W. Nowak, Reliab. Eng. Syst. Safe. **106**, 179 (2012).
28 D. M. Luchtenburg, S. L. Brunton, and C. W. Rowley, J. Comput. Phys. **274**, 783 (2014).
29 J. T. Horwood, and A. B. Poore, IEEE T. Automat. Contr. **56**, 1777 (2011).
30 K. J. DeMars, R. H. Bishop, and M. K. Jah, J. Guid. Control Dynam. **36**, 1047 (2013).
31 M. L. Psiaki, J. R. Schoenberg, and I. T. Miller, J. Guid. Control Dynam. **38**, 292 (2015).
32 M. Berz, *Modern Map Methods in Particle Beam Physics* (Academic Press, London, 1999).
33 M. Gunay, U. Orguner, and M. Demirekler, IEEE T. Aero. Elec. Sys. **52**, 2732 (2016).
34 K. Fujimoto, and D. J. Scheeres, J. Guid. Control Dynam. **38**, 1146 (2015).
35 V. Vittaldev, R. P. Russell, and R. Linares, J. Guid. Control Dynam. **39**, 2615 (2016).
36 M. Massari, P. Di Lizia, F. Cavenago, and A. Wittig, "Differential Algebra software library with automatic code generation for space embedded applications", AIAA Paper No. 2018–0398, 2018.
37 P. Di Lizia, R. Armellin, and M. Lavagna, Celest. Mech. Dyn. Astr. **102**, 355 (2008).
38 L. Isserlis, "On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables", Biometrika, **12** (1918).
39 R. Kan, J. Multivariate Anal. **99**, 542 (2008).
40 P. Gurfil, and P. K. Seidelmann, *Celestial Mechanics and Astrodynamics: Theory and Practice, volume 436 of Astrophysics and Space Science Library* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2016).
41 K. T. Alfriend, *Spacecraft Formation Flying: Dynamics, Control, and Navigation* (Elsevier astrodynamics series, Butterworth-Heinemann/Elsevier, Oxford, 2010).