

# State-of-the-Art Orbit Uncertainty Propagation Methods used in Space Situational Awareness (SDA)

Srihari S

## Introduction

In the contemporary landscape of Space Domain Awareness (SDA) and Space Situational Awareness (SSA), the precise characterization of orbital state uncertainty is not merely a mathematical exercise but a requirement for the preservation of the orbital environment. As the population of resident space objects (RSOs) grows exponentially, the risk of catastrophic collisions—such as the 2009 Iridium-Cosmos event—escalates, potentially precipitating the “Kessler Syndrome”, according to which, as the density of objects in orbit increases, the likelihood of collisions grows, leading to a cascading chain reaction where each collision generates new debris. This self-perpetuating process could eventually render certain orbits unusable for satellites and spacecraft, posing a severe threat to both current and future space operations.

Uncertainty Propagation (UP) serves as the mathematical engine for several mission-critical tasks:

- **Conjunction Assessment (CA):** Predicting the probability of collision ( $P_C$ ) between two objects by propagating their state uncertainties to the time of closest approach (TCA).

- **Object Custody:** Establishing and maintaining the identity and tracking of newly detected objects through Bayesian filtering.
- **Sensor Tasking:** Optimizing the use of limited ground-based radars and optical sensors by predicting where an object's probability mass will be at future epochs.
- **Maneuver Planning:** Assessing the necessity and efficacy of collision avoidance maneuvers (CAMs) under non-Gaussian uncertainty regimes.

## Problem Formulation of Uncertainty Propagation

The stochastic uncertainty propagation problem is defined on a probability space  $(\mathcal{S}, \Sigma, \mathcal{P})$  where  $\mathcal{S}$  is the sample space,  $\Sigma$  is the  $\sigma$ -algebra, and  $\mathcal{P}$  is the probability measure.

The random input vector  $\boldsymbol{\xi} \in \mathbb{R}^d$  is defined on  $(\mathcal{S}, \Sigma, \mathcal{P})$  and consists of  $d$  independent random inputs that characterize the stochastic system. These inputs are drawn from a probability distribution defined over the space  $\mathcal{S} \rightarrow \Gamma^d \subseteq \mathbb{R}^d$ , which is the support of the random variables. The choice of  $\Gamma^d$  depends on the problem; for example,  $\Gamma^d = [0, 1]^d$  for  $d$ -dimensional uniform random variables in the range  $[0, 1]$ .

The uncertainty propagation problem seeks to generate a solution to a stochastic ordinary differential equation:

$$\mathcal{A}(t, \cdot; \mathbf{x}) = 0, \quad (t, \cdot) \in [0, t_f] \times \Gamma^d, \quad \mathcal{P}\text{-a.s. in } \mathcal{S} \quad (1)$$

where  $t \in [0, t_f]$  is the time variable,  $\mathbf{x} \in \mathbb{R}^n$  is a vector of Quantities of Interest (QoIs), and  $\mathcal{A}$  is the stochastic ODE operator that defines the flow of  $\mathbf{x}$  to  $t$  as a function of  $\cdot$ . A solution to the uncertainty propagation problem quantifies the effects of random inputs on the random QoIs  $\mathbf{x}$  as a function of time  $t$  and the random inputs  $\boldsymbol{\xi}$ .

The random inputs  $\boldsymbol{\xi}$  may include initial state uncertainties, force-modeling errors, and environmental perturbations.

- **Sample Space ( $\mathcal{S}$ ):** The set of all possible outcomes of a random experiment.  
*Example:* For a coin toss,  $\mathcal{S} = \{\text{Heads}, \text{Tails}\}$ .
- **$\sigma$ -Algebra ( $\Sigma$ ):** A collection of subsets of the sample space that includes  $\mathcal{S}$  and is closed under complements and countable unions. These subsets are the “events” we can assign probabilities to.  
*Example:* For a die roll,  $\Sigma$  might include sets like  $\{1, 2, 3\}$ ,  $\{4, 5, 6\}$ ,  $\mathcal{S}$ , and  $\emptyset$ .
- **Probability Measure ( $\mathcal{P}$ ):** A function that assigns a number between 0 and 1 to each event in the  $\sigma$ -algebra, satisfying:
  - $\mathcal{P}(\mathcal{S}) = 1$
  - If events are disjoint,  $\mathcal{P}(\bigcup_i A_i) = \sum_i \mathcal{P}(A_i)$ .  
*Example:* For a fair die,  $\mathcal{P}(\{2\}) = \frac{1}{6}$ .

## Taxonomy of Orbital Uncertainties

The literature categorize the non-deterministic inputs into two distinct mathematical and physical frameworks:

- **Aleatory Uncertainty (Random):**

- **Definition:** Characterizes the inherent, irreducible randomness within the physical system.
- **Examples:** Sensor measurement noise from ground-based radars or optical telescopes, and fluctuations in atmospheric density that affect drag in a stochastic manner.
- **Modeling:** Almost universally represented using precise probability distributions (PDFs).

- **Epistemic Uncertainty (Systematic):**

- **Definition:** Arises from a lack of knowledge or limited information about the system’s physics or model parameters.
- **Examples:** Modeling inadequacies such as truncated gravity field harmonics (e.g., using a 2x2 vs. 70x70 model), unknown area-to-mass ratios for solar radiation pressure, and errors introduced by using fast surrogate models (metamodels) instead of high-fidelity propagators.
- **Modeling:** While often forced into a PDF-based representation, recent research suggests using Outer Probability Measures (OPMs) and credibilistic filtering to bound these errors without over-characterizing them (Jones et al. 2019).

## Foundational Assumptions in Orbital Uncertainty Propagation

To make the complex dynamics of orbital motion computationally tractable, researchers generally operate under a standardized set of mathematical and physical assumptions:

- **Point Mass Dynamics:** For most scenarios, space objects are approximated as point masses, although this assumption may be relaxed for large structures like the International Space Station (ISS).
- **State Representation:** In orbital mechanics, the QoIs are represented by the **state vector**  $\mathbf{x}$ , representing position and velocity in an inertial frame (e.g., ECI) or a specialized coordinate set like the Modified Equinoctial Orbital Elements (MEqOE).
- **Initial State Quantification:** It is assumed that an initial state  $\mathbf{x}_0$  is known at epoch  $t_0$ , accompanied by a multivariate Probability Density Function (PDF), often initially modeled as Gaussian  $N(\mu_0, \mathbf{P}_0)$ , where  $\mu_0 = \mathbf{x}_0$  is the mean state vector and  $\mathbf{P}_0$  is the initial covariance matrix.
- **Deterministic Flow:** The trajectory follows non-linear Ordinary Differential Equations (ODEs) influenced by both central-body gravity and various perturbative accelerations. Hence we do not consider epistemic uncertainty in the dynamics model itself, but rather focus on the propagation of the initial state uncertainty through these deterministic equations.

## The Fokker-Planck Equation

The **Fokker-Planck Equation** (FPE) is the fundamental partial differential equation (PDE) governing the time evolution of the PDF of a dynamical system subject to both deterministic forces and stochastic process noise.

- **Role in UP:** It describes how a probability cloud “flows” and “spreads” through state space. The equation consists of a **drift term** (representing deterministic motion like Keplerian flow) and a **diffusion term** (representing the broadening of the PDF due to stochastic forcing).

- **Mathematical Expression:** For a state vector  $\mathbf{x}$ , the FPE is given by:

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = -\nabla \cdot (\mathbf{f}p) + \frac{1}{2}\text{Tr}(\nabla^2 p) \quad (2)$$

where  $\mathbf{f}$  is the deterministic flow vector,  $\mathbf{D}$  is the diffusion tensor, and  $p(\mathbf{x}, t)$  is the PDF at time  $t$ .

- **The Computational Challenge in astrodynamics:** Direct numerical solutions to the FPE are generally intractable in astrodynamics due to the “**curse of dimensionality**”; a 6D orbital state requires spatial grids that grow exponentially, necessitating the use of the state-of-the-art approximation methods discussed in this report.

## Concept Map of Orbital Uncertainty Propagation

Section 1.1 of Fossa et al. (2024) provides a detailed concept map on the topic of **Orbit Uncertainty propagation**. This taxonomy is fundamental for understanding how different methodologies balance computational efficiency and statistical accuracy in space situational awareness (SSA) applications.

The map is divided into three primary pillars: **Linear Methods**, **Nonlinear Methods**, and **Coordinates Transformation**.

### 1. Linear Methods

These methods assume that a linearized dynamical model can capture the evolution of neighboring trajectories and that the uncertainty remains characterized by a multivariate Gaussian distribution.

- **Linear Covariance (LinCov):** Uses a first-order Taylor expansion to compute the time evolution of the covariance matrix.
- **CADET (Covariance Analysis Describing Function Technique):** A statistical linearization method.

- **Limitations:** These are computationally “light” but lose accuracy in highly nonlinear regimes, large initial uncertainties, or long propagation windows.

## 2. Nonlinear Methods

Nonlinear methods can be into three subcategories based on their fundamental approach to the UP problem.

### A. Sample-based Methods

These techniques treat the underlying orbital dynamics as a “**black-box**,” which simplifies implementation because they do not require internal modifications to existing code.

- **Monte Carlo (MC):** The non-parametric benchmark. While accurate and able to handle any nonlinearity, it is often too computationally expensive for real-time applications.
- **Unscented Transform (UT) and Conjugate Unscented Transform (CUT):** Deterministic sampling methods that approximate the state pdf rather than the transformation itself by using “sigma points”.
- **Non-intrusive Polynomial Chaos (PC):** Constructs a functional representation of the uncertainty by fitting spectral coefficients to transformed samples.
- **SC Multifidelity (Jones & Weisman):** A hybrid method included in the mind-map that uses stochastic collocation to correct low-fidelity samples with a small subset of high-fidelity runs.

### B. Dynamics-based Methods

Unlike sample-based methods, these require complete mathematical knowledge of the system’s governing equations.

- **Intrusive PC:** Transforms a stochastic differential equation into a higher-dimensional deterministic one using Galerkin projection.

- **State Transition Tensors (STT):** Captures nonlinear warpings by numerically integrating variational equations for higher-order partial derivatives.
- **Differential Algebra (DA):** Automates arbitrary-order Taylor expansions by replacing standard arithmetic with polynomial algebra.
- **DA-LOADS Multifidelity (Fossà):** The mind-map explicitly labels the author’s core contribution here—a multifidelity technique that leverages **Low-Order Automatic Domain Splitting (LOADS)** to adaptively manage nonlinearities.

### C. Pdf-based Methods

These methods aim to provide a functional or analytical representation of the evolved probability density function.

- **Fokker-Planck Equation (FPE):** A partial differential equation (PDE) that provides a complete statistical description but suffers from the “curse of dimensionality”.
- **Liouville’s Equation:** A quasi-linear PDE used when the diffusion term (process noise) can be neglected.
- **Gaussian Mixture Models (GMM):** Approximates a non-Gaussian pdf as a weighted sum of Gaussian kernels, transforming one complex problem into several smaller sub-problems.

### 3. Coordinates Transformation

In his thesis, Fossà emphasizes that nonlinearity is not just an intrinsic system attribute but is **coupled with the mathematical representation** chosen. In the context of UP, this fact can be leveraged to reduce the complexity of the problem by choosing a set of coordinates that minimizes the nonlinearity of the system at hand. In this case, linearized or lower-order UP methods could still provide a sufficient accuracy in the transformed

state pdf while significantly improving the computational performance (Fossa et al. 2024). Uncertainty propagation methods were applied to the following coordinate sets in literature:

- **Cartesian:** Popular for simplicity but highly nonlinear due to inverse-square gravity.
- **Orbital Elements:**
  - **Classical (COEs):** Intuitive but contain singularities for circular or equatorial orbits.
  - **Equinoctial (EqOEs, MEEs):** Non-singular sets suitable for computation and perturbation analysis.
  - **Alternate Equinoctial (AEqOEs):** Elements that evolve linearly in time under unperturbed Keplerian motion, which is highly beneficial for UP.
  - **$J_2$ -perturbed Equinoctial Elements:** A set of elements that account for the  $J_2$  perturbation, which is significant for low Earth orbit (LEO) satellites.
  - **Generalized Equinoctial Elements (GEqOEs):** A more flexible set of elements that can be adapted to any perturbing force derived from a potential function, including  $J_2$  and higher-order zonal harmonics.
- **Regularized Coordinates:** Redundant variables like the **Dromo** formulation or **Kustaanheimo-Stiefel (KS)** variables used to regularize dynamics.

The following sections provide a detailed overview of the most widely used methods in the literature, highlighting their theoretical foundations, practical implementations, and performance characteristics.

---

## The Sampling Baseline: Monte Carlo (MC)

The Monte Carlo method serves as the “ground truth” and absolute benchmark for all other techniques. The process involves generating a large set of random samples ( $N_{\text{samples}}$ ) from an initial multivariate Gaussian distribution using the square-right matrix ( $S$ ) of the covariance matrix.

$$P = SS^T \quad (3)$$

The samples are created as follows:

$$\mathbf{x}_0(\xi) = \mu_0 + S\xi, \quad \xi \sim N(0, I_6) \quad (4)$$

Where  $\xi$  is a vector of standard normal random variable inputs, generated using a pseudo random number simulator. The random inputs may include initial state uncertainties, force-modeling errors, and environmental perturbations.

The covariance matrix  $P$  captures the uncertainty in the state estimate which we get from orbit determination. The square-right matrix  $S$  is obtained through Cholesky decomposition or Singular Value Decomposition (SVD) of  $P$ . This allows us to generate samples that are consistent with the initial uncertainty.

## Mechanics of the Sampling-Based Approach

MC simulation functions by drawing a large ensemble of random samples from an initial distribution, which can be Gaussian or any other known PDF.

- **Independent Propagation:** Each sample represents a potential realization of the system state and is propagated independently through numerical integration of the dynamics.
- **Non-Intrusive Nature:** Because each sample is treated separately, MC is intrinsically non-intrusive, meaning it treats the dynamics model as a “black box” and requires no modifications to existing dynamics code.
- **Statistical Reconstruction:** After propagation, the ensemble of final states provides a discrete representation of the final PDF. Statistical properties, such as the mean and covariance, are estimated directly by computing the average and sample-covariance of the propagated points.

## Algorithmic Flow of Monte Carlo Simulation

```
graph TD
    A[Initial PDF & Covariance] --> B[Generate  $\Omega$  Random Samples]
    B --> C[Propagate Each Sample Independently via RK8/High-Fidelity Dynamics]
    C --> D[Collect Final States]
    D --> E[Compute Mean & Covariance]
    D --> F[Construct Terminal PDF/Histogram]
    E --> G[Validation Baseline for Other Methods]
    F --> G
```

## Role as the “Ground Truth” Baseline

The sources [Bos (2025)](Soman 2025) identify MC simulation as the **definitive benchmark** for evaluating the accuracy of all other UP methods.

- **Reliability:** MC results approach the “true” probability distribution as the number of samples goes to infinity.
- **Accuracy Metric:** In comparative studies, researchers typically use a large-scale MC run (such as  $10^5$  samples) as the reference truth to calculate accuracy metrics like the **N L2 distance** for alternative methods.

## The Computational Trade-off

While highly accurate, the primary limitation of MC is its **extreme computational intensity**.

- **Convergence Rate:** The rate of statistical convergence is proportional to the inverse square root of the ensemble size (Monte Carlo Standard error  $\propto 1/\sqrt{N}$ , where  $N$  is the number of samples), meaning achieving high accuracy requires a massive number of samples.

### 💡 Monte-Carlo Standard Error

The statistical convergence rate describes how quickly the Monte Carlo estimate approaches the true value as the number of samples increases.

For this, we assume that the samples are independent and identically distributed (i.i.d.) random variables drawn from the underlying probability distribution. Let these samples be denoted as  $X_1, X_2, \dots, X_N$ , where  $N$  is the total number of samples.

Now we assume  $Y = f(\mathbf{X})$  to be a random variable with

$$E[Y] = \mu, \quad \text{Var}(Y) = \sigma^2.$$

Which gives us  $Y_1, Y_2, \dots, Y_N$  as i.i.d. random variables which form the output underlying uncertainty distribution. The Monte Carlo estimator for the expected value of  $Y$  is given by,

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N Y_i.$$

Since the samples are independent,

$$\begin{aligned} \text{Var}(\hat{\mu}) &= \text{Var}\left(\frac{1}{N} \sum_{i=1}^N Y_i\right) \\ &= \frac{1}{N^2} \sum_{i=1}^N \text{Var}(Y_i) \frac{\sigma^2}{N} \end{aligned}$$

Hence, the standard error of the Monte Carlo estimator is

$$\boxed{\text{SE}(\hat{\mu}) = \sqrt{\text{Var}(\hat{\mu})} \frac{\sigma}{\sqrt{N}}.}$$

Therefore, the statistical error of Monte Carlo decreases as

$$\boxed{\text{SE}(\hat{\mu}) = O(N^{-1/2}),}$$

implying that reducing the error by a factor of  $(k)$  requires approximately  $(k^2)$  times more samples.

- **Collision Assessment Barriers:** For rare-event quan-

tification, such as conjunction assessment where collision probabilities ( $P_C$ ) may be  $10^{-4}$  or lower, the number of required samples can reach  $10^7$  **or more** to achieve low statistical error.

- **Practical Infeasibility:** Due to these requirements, the sources note that standard MC simulations are often infeasible for practical, real-time application to the vast population of resident space objects (RSOs).

## Variations and Technological Extensions

To mitigate the extreme computational burden of the base Monte Carlo simulation, several variations have been developed that leverage modern hardware and statistical theory:

- **Parallel MC (GPGPU):** Since samples are independent, they are ideally suited for massively parallel architectures,. Implementation on Graphics Processing Units (GPGPUs) can reduce computational time linearly with the number of cores, though it introduces bottlenecks in memory transfer (PCI Express bus) and requires re-coding dynamics without standard libraries like SPICE.
- **Importance Sampling (IS) and Cross-Entropy (CE):** In conjunction analysis, where collision probability ( $P_C$ ) is a “rare event” (often  $10^{-4}$  to  $10^{-6}$ ), brute-force methods are computationally exhaustive,. IS biases the sampling toward the “failure set” (the collision region), and the Cross-Entropy method iteratively solves for the optimal proposal distribution.
- **Surrogate-Assisted MC:** Rather than numerical integration for every sample, surrogate models—such as high-order Taylor polynomials via Differential Algebra (DA) or Polynomial Chaos Expansions (PCE)—are used to rapidly evaluate samples.
- **MC from Epoch vs. MC from TCA:** Baseline “from Epoch” propagation uses high-fidelity models from the initial measurement time, while “from TCA” propagates uncertainty to the Time of Closest Approach (TCA) first

and then performs a localized MC simulation, significantly saving time when it comes to rare-event probability (like collision probability  $P_C$ ) estimation.

## Linearized Covariance (LinCov)

In the taxonomy of uncertainty propagation (UP) methods, **Linearised Covariance (LinCov)** represents the foundational entry in the category of **Linear Approximation Methods**. It is characterized by its reliance on first-order Taylor-series expansions to map statistical moments through nonlinear dynamical systems. While computationally superior in terms of execution time, its mathematical architecture introduces significant epistemic risks when the system's nonlinearity induces non-Gaussianity.

### Mathematical Framework and Mechanics

LinCov operates on the fundamental assumption that if a reference trajectory  $x^*(t)$  is sufficiently close to the true trajectory  $x(t)$ , the deviation  $\Delta x(t)$  can be modeled as a linear system.

- **First-Order Truncation:** The method defines the system dynamics  $\dot{x}(t) = F(x, t)$  and expands them into a Taylor series about the reference trajectory. By neglecting higher-order terms (second-order and above), the dynamics are simplified to  $\Delta \dot{x}(t) \approx J^*(t) \Delta x(t)$ , where  $J^*$  is the Jacobian matrix evaluated along the reference path.
- **The State Transition Matrix (STM):** The solution to this linear system is facilitated by the STM,  $\Phi(t, t_0)$ , which maps the initial deviation to a future epoch.

#### State Transition Matrix (STM)

The STM is formally defined as the solution to the matrix differential equation  $\dot{\Phi}(t, t_0) = \mathbf{J}^*(t) \Phi(t, t_0)$ , with the initial condition  $\Phi(t_0, t_0) = \mathbf{I}$  (the identity matrix).

- **Moment Propagation:** The mean state vector and covariance matrix are defined as the expected value of the state, and the expected squared deviation from the mean state, respectively:

$$\bar{\mathbf{x}} = \int_{-\infty}^{\infty} \xi p(\xi) d\xi$$

$$\mathbf{P} = \int_{-\infty}^{\infty} (\xi - \bar{x})(\xi - \bar{x})^T p(\xi, t) d\xi$$

Assuming that the differences between the true and reference trajectories are small, the mean and covariance can be propagated through the STM as follows:

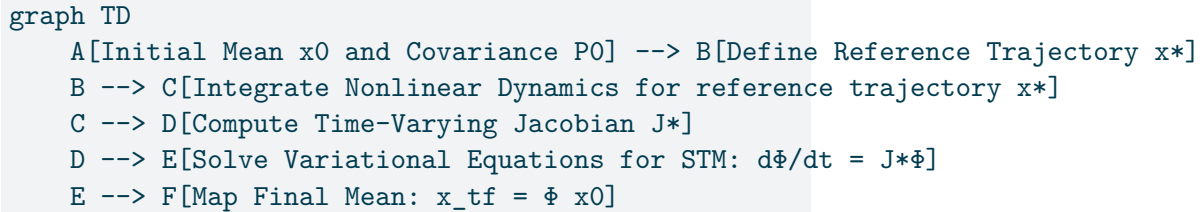
$$\begin{aligned} \bar{\mathbf{x}}(t) &= \Phi(t, t_0) \bar{\mathbf{x}}(t_0) \\ \mathbf{P}(t) &= \Phi(t, t_0) \mathbf{P}(t_0) \Phi^T(t, t_0) + G(t) Q(t) G^T(t) \end{aligned} \quad (5)$$

Where  $G(t)$  is the process noise input matrix (that characterize the diffusion of a Brownian motion process) and  $Q(t)$  is the process noise covariance, both of which are used to account for force-model mismodelling perturbation errors. Bos (2025) does not include process noise in his LinCov implementation.

- **Gaussian Assumption:** A defining characteristic of LinCov is the preservation of the distribution's functional form; it assumes that an initially Gaussian PDF remains Gaussian throughout the propagation horizon.

## Algorithmic Implementation of LinCov

The following flowchart delineates the procedural logic for propagating orbital uncertainty using the LinCov framework:



```
F --> G[Map Final Covariance: P_tf = Φ P0 ΦT]
G --> H[Output Analytic Gaussian PDF]
```

## Performance Analysis: The Efficiency-Accuracy Trade-off

LinCov’s role in Space Domain Awareness (SDA) is defined by its extreme computational efficiency, often being several orders of magnitude faster than sampling-based methods.

- **Computational Primacy:** LinCov is the fastest implemented method in modern comparative studies because it requires the integration of only one nominal state vector to solve the variational equations.
- **Failure in Nonlinear Regimes:** The accuracy of LinCov inversely correlates with propagation time and dynamical nonlinearity (e.g., highly elliptical orbits). As the true PDF warps into “banana” or “kidney” shapes, the linear mapping fails to capture the curvature, leading to a significant increase in the  $NL_2$  distance metric relative to the Monte Carlo truth.
- **Structural Limitations:** While LinCov captures the location (mean) and the general spread (scale) of the uncertainty, it systematically ignores the higher-order moments (skewness and kurtosis) that define the “tails” of a non-Gaussian distribution.

## Contextual Nuance: Surprising Utility in Conjunction Assessment

A critical insight provided by Bos (2025) is that LinCov’s failure to approximate the *shape* of a PDF does not necessarily preclude its utility in specific **Conjunction Assessment (CA)** tasks.

- **Preservation of Decision Metrics:** In simulations involving Low Earth Orbit (LEO) conjunctions with long propagation times (e.g., 96 hours), LinCov produced collision probabilities ( $P_C$ ) nearly identical to baseline Monte

Carlo runs, despite the underlying distributions being highly non-Gaussian.

- **Geometric Explanation:** This “surprising” performance suggests that for certain conjunction geometries, capturing the “length” and “width” of the uncertainty spread is more critical for  $P_C$  calculation than accurately modeling the time of closest approach (TCA) uncertainty distribution.
- **Caveats:** This robust performance may be an artifact of specific scenario settings, such as identical orbital regimes for both objects or symmetrical initial uncertainties.

## Taxonomy and Comparison

LinCov sits alongside several related techniques that utilize varying forms of linearization:

- **Statistical Linearization (CADET):** Unlike LinCov, the **Covariance Analysis Describing function Technique (CADET)** approximates system functions in an “average” way to minimize mean square error and does not strictly rely on the differentiability of the dynamics.
- **State Transition Tensors (STT):** This framework extends LinCov into higher-order statistics; a first-order STT solution corresponds exactly to the STM used in standard LinCov, while higher-order tensors allow for the propagation of uncertainties through stronger nonlinearities without random sampling. See Section for a detailed discussion.
- **Coordinate Frame Synergy:** The accuracy of linear methods is significantly enhanced when operating in **Generalized Equinoctial Orbital Elements (GEqOE)** or **Modified Equinoctial Elements (MEqOE)**, which render the dynamics more nearly linear than Cartesian coordinates.

## Moment Method

The **Moment Method** (or Method of Moments) is a statistical technique utilized in uncertainty propagation to estimate the evolution of a probability density function (PDF) by focusing on its statistical moments—primarily the mean and covariance. Within the hierarchy of uncertainty propagation methods, it serves as a computationally efficient bridge between low-fidelity linear approximations and high-fidelity, but expensive, sampling-based techniques like Monte Carlo (MC).

### Mathematical Framework and Methodology

The method relies on the assumption that if an input variable (or vector) is characterized by a Gaussian distribution, its first two moments are necessary and sufficient to describe its state (Soman 2025).

### Taylor Series Expansion

The core mechanism involves expanding the nonlinear system function into a Taylor series about the mean value of the inputs.

#### Explicit vs. Implicit Formulations:

- **Explicit Moment Method:** Applied when a direct functional relationship  $y = f(x)$  exists between input and output.
- **Implicit Moment Method:** Utilized in astrodynamics when only the governing differential equations  $\dot{x} = f(x, t)$  are known. It requires a numerical integrator (e.g., RKF78) to define the solution flow  $\phi$ , which is then expanded.

## Moment Propagation

In order to explain the moment propagation, consider an explicit moment propagation problem with a nonlinear function  $y = f(x)$  where  $x$  is a random variable with mean  $\mu_x$  and covariance  $\sigma_x^2$ . The first two moments of the output  $y$  can be approximated depending on the order of the Taylor approximation:

### First-Order Equations:

1. **Mean** ( $\mu_y$ ):  $\mu_y = f(\mu_x)$
2. **Variance** ( $\sigma_y^2$ ):  $\sigma_y^2 = [f'(\mu_x)]^2 \sigma_x^2$

While fast, moment propagation on the first-order Taylor approximation fails to capture the “mean shift” caused by nonlinearities and performs poorly when the PDF becomes highly non-Gaussian.

### Second-Order Equations:

1. **Mean**:  $\mu_y = f(\mu_x) + \frac{1}{2} f''(\mu_x) \sigma_x^2$
2. **Variance**:  $\sigma_y^2 = [f'(\mu_x)]^2 \sigma_x^2 + f'(\mu_x) f''(\mu_x) \mu_3^{(x)} + \frac{1}{4} [f''(\mu_x)]^2 (\mu_4^{(x)} - \sigma_x^4)$

### Terms Defined:

- $f(\mu_x)$ : Function evaluation at the mean.
- $f'(\mu_x)$ : First derivative (sensitivity) at the mean.
- $f''(\mu_x)$ : Second derivative (curvature) at the mean.
- $\mu_3^{(x)}$ : The third central moment of the input distribution.
- $\mu_4^{(x)}$ : The fourth central moment of the input distribution.
- $\sigma_x^4$ : The square of the input variance.

### Implicit Formulation:

In high-fidelity astrodynamics, an explicit formula  $y = f(x)$  is rarely available. Instead, the system is defined by a set of nonlinear Ordinary Differential Equations (ODEs):

$$\dot{\mathbf{x}} = f(\mathbf{x}, t)$$

The **Implicit Moment Method** uses the **solution flow**  $\mathbf{x}(t) = \phi(t; \mathbf{x}_0)$  obtained via numerical integration (e.g., Runge-Kutta schemes). The methodology treats the numerical integrator as the mapping function:

1. **Trajectory Integration:** Propagate the nominal mean  $\mu_{x0}$  to find the final state  $\phi(\mu_{x0})$ .
2. **Sensitivity Extraction:** Compute the Jacobian and Hessian of the flow map, often utilizing **Automatic Differentiation (AD)** to ensure machine-precision accuracy.
3. **Moment Transformation:** Apply the multivariate moment equations using the flow derivatives to advance the mean and covariance to the target epoch  $t_f$ .

In a 6-dimensional orbital state, the method propagates the mean vector  $\mu$  and the covariance matrix  $\Sigma$ . The following sections detail the multivariate extension of the moment method.

### Multivariate Extension ( $R^n$ )

When the input is an  $n$ -dimensional random vector  $\mathbf{x} \sim \mathcal{N}(\mu_x, \Sigma_x)$ , the propagation involves the Jacobian matrix and Hessian tensors.

#### First-Order (Linearized) Moments:

$$\begin{aligned}\mu_y &= f(\mu_x) \\ \Sigma_y &= \mathbf{J}_x \mathbf{J}_x^T\end{aligned}$$

#### Second-Order Moments (Component-wise):

$$\begin{aligned}\mu_{y,i} &= f_i(\mu_x) + \frac{1}{2} \text{tr}(\mathbf{H}_{fi}(\mu_x) \Sigma_x) \\ \Sigma_y(i, p) &= (\mathbf{J}_x \mathbf{J}_x^T)_{ip} + \frac{1}{2} \text{tr}(\mathbf{H}_{fix} \mathbf{H}_{fpx} \Sigma_x)\end{aligned}$$

#### Terms Defined:

- **J**: The  $n \times n$  Jacobian matrix,  $\mathbf{J} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mu_x}$ .

- $\mathbf{H}_{fi}$ : The Hessian matrix of the  $i$ -th component of the function  $f$ .
- $\text{tr}(\cdot)$ : The trace operator, summing the diagonal elements of the resulting matrix product.

## Comparative Orders of Approximation

The accuracy of the Moment Method is strictly tied to the order of the Taylor expansion used:

- **First-Order (M1):**
  - Assumes the system is locally linear.
  - Propagates the mean as the function of the mean:  $\mu_y = f(\mu_x)$ .
  - While fast, it fails to capture the “mean shift” caused by nonlinearities and performs poorly when the PDF becomes highly non-Gaussian.
- **Second-Order (M2):**
  - Incorporates the **Hessian matrix** ( $H$ ) of the system dynamics.
  - The mean is corrected by a term involving the trace of the Hessian and the covariance:  $\mu_{y,i} \approx f_i(\mu_x) + \frac{1}{2}\text{tr}(H_i \Sigma_x)$ .
  - The covariance propagation includes additional terms to account for the “spreading” effect of curvature:  $\Sigma_y \approx J \Sigma_x J^T + \frac{1}{2}\text{tr}(H_i \Sigma_x H_p \Sigma_x)$ .
  - Empirical results show that M2 consistently maintains lower error norms ( $L_2$  and  $L_\infty$ ) than M1, especially in nonlinear regimes like Highly Elliptical Orbits (HEO).

## Integration with Gaussian Mixture Models (GMM)

In practice, the implicit moment method is often combined with **Gaussian Mixture Models (GMM)** to handle the “loss of Gaussianity” that occurs in long-term orbital propagation. Each Gaussian component is propagated independently using the moment method, and the final non-Gaussian PDF is reconstructed by recombining the weighted sum of these components.

1. **Decomposition:** The initial Gaussian uncertainty is split into  $N$  smaller Gaussian components (kernels) using algorithms like the Horwood suboptimal algorithm.
2. **Local Propagation:** Each individual component has a sufficiently small covariance to remain “locally Gaussian,” allowing the Moment Method to propagate its first two moments accurately over longer durations.
3. **Reconstruction:** The final non-Gaussian PDF is reconstructed by recombining the weighted sum of these propagated components.

## Algorithm for Multivariate Implicit GMM-Moment Propagation

The following flowchart outlines the computational algorithm for a multivariate system (e.g., orbital mechanics in  $R^6$ ) using the implicit second-order moment method:

```
graph TD
    A["Initial State: Mean  $\mu$ , Covariance  $\Sigma$ "] --> B["Horwood Splitting Algorithm"]
    B --> C["For each component  $i=1$  to  $N$ "]
    C --> D["Numerical Integration: RK8/RKF78"]
    D --> E["Automatic Differentiation: TAPENADE"]
    E --> F["Extract Jacobian  $J$  and Hessian  $H$ "]
    F --> G["Implicit Moment Method Update"]
    G --> H["Propagated Component:  $\mu_i$ ,  $\Sigma_i$ "]
    H --> I["Recombine Components with Weights  $w_i$ "]
    I --> J["Final Reconstructed PDF"]
    J --> K["Compute Overall Mean  $\mu_f$  and Covariance  $\Sigma_f$ "]
```

## Critical Strengths and Limitations

- **Computational Efficiency:** The GMM-Moment method is significantly faster than Monte Carlo simulations. For a 3-day LEO propagation, it completed in seconds/minutes compared to the hours required for a  $10^7$ -sample MC run.
- **Derivative Sensitivity:** Unlike the Unscented Transform (UT), which is sample-based, the Moment Method requires explicit derivative information (Jacobians and Hessians).
- **Automatic Differentiation (AD):** To avoid the errors of finite differences, modern implementations use AD (e.g., TAPENADE) to obtain exact derivatives by applying the chain rule to the source code of the propagator.
- **Differentiability Requirement:** The method assumes that the governing equations are continuous and differentiable, which may pose challenges in the presence of discrete events like shadow entries (SRP discontinuities).

## Unscented Transform (UT)

In the rigorous taxonomy of orbital uncertainty quantification, the **Unscented Transform (UT)** represents a pivotal **deterministic sampling method** that bridges the gap between computationally cheap linear approximations and prohibitively expensive brute-force stochastic simulations. Unlike the Monte Carlo (MC) method, which relies on random realizations, the UT operates on the foundational intuition that it is mathematically easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation.

## Theoretical Context: Deterministic vs. Random Sampling

Within the landscape of sampling-based methodologies, the UT occupies a unique niche defined by “moderate fidelity” and high efficiency.

- **Sigma Point Paradigm:** The UT is predicated on the intuition that “it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation”. Unlike the EKF, which linearizes the system dynamics via a first-order Taylor series expansion (State Transition Matrix), the UT parameterizes the state distribution using a set of “sigma points”. While MC requires  $10^5$  to  $10^7$  random samples to achieve statistical convergence, the UT utilizes a minimal, carefully selected set of weighted **sigma points** that precisely capture the initial mean and covariance of a distribution.
- **Black-Box Dynamics:** Like MC, the UT is non-intrusive; it treats the system dynamics as a “black box,” passing the discrete sigma points through the actual nonlinear equations of motion without requiring the derivation of Jacobians or Hessians.
- **Gaussian Constraint:** A critical limitation noted in the sources is that the standard UT is primarily constrained by **Gaussian assumptions**. It nonlinearly propagates the first and second moments but may fail to naturally represent the “banana-shaped” non-Gaussian warpings that emerge during long-term orbital propagation.

## Mathematical Architecture of the UT

For an  $n$ -dimensional state  $\mathbf{x}$  with mean  $\bar{\mathbf{x}}$  and covariance  $\mathbf{P}_{xx}$ , a set of  $2n + 1$  sigma points  $\chi_i$  is generated using a matrix square root (typically Cholesky decomposition) of the scaled covariance. The mathematical selection and weight assignment are governed by the following formulations:

## Sigma Point Selection

$$\chi_0 = \bar{\mathbf{x}}$$

$$\chi_i = \bar{\mathbf{x}} + \left( \sqrt{(n + \kappa)\mathbf{P}} \right)_i, \quad i = 1, \dots, n \quad (6)$$

$$\chi_{i+n} = \bar{\mathbf{x}} - \left( \sqrt{(n + \kappa)\mathbf{P}} \right)_i, \quad i = 1, \dots, n \quad (7)$$

## Weight Allocation

$$W_0 = \frac{\kappa}{n + \kappa}$$

$$W_i = \frac{1}{2(n + \kappa)}, \quad i = 1, \dots, 2n$$

## Terms Defined:

- $\bar{\mathbf{x}}$ : The mean state vector.
- $\mathbf{P}$ : The state covariance matrix.
- $(\sqrt{\cdot})_i$ : The  $i$ -th column of the matrix square root (typically calculated via **Cholesky decomposition**).

### Cholesky Decomposition

The Cholesky decomposition of a positive definite matrix  $\mathbf{P}$  is a lower triangular matrix  $\mathbf{L}$  such that  $\mathbf{P} = \mathbf{L}\mathbf{L}^T$ .

- $\kappa$ : A scaling/tuning parameter, often set such that  $n + \kappa = 3$  for Gaussian distributions to minimize higher-order errors.

## Statistical Reconstruction

After passing each  $\chi_i$  through the nonlinear dynamics  $\mathbf{f}(\cdot)$ ,

$$\mathbf{y}_i = \mathbf{f}(\chi_i)$$

the terminal mean ( $\hat{\mathbf{y}}$ ) and covariance ( $\hat{\mathbf{P}}$ ) are reconstructed:

**Transformed Mean:**

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n} W_i \mathbf{y}_i$$

**Transformed Covariance:**

$$\bar{\mathbf{P}} = \sum_{i=0}^{2n} W_i (\mathbf{y}_i - \hat{\mathbf{y}})(\mathbf{y}_i - \hat{\mathbf{y}})^T$$

### Algorithm: The Unscented Filter (UKF) Logic

The following flowchart delineates the procedural execution of the UT within a predictive filtering framework, such as the Unscented Kalman Filter:

```
graph TD
    A[Initial Mean and Covariance P] --> B[Compute Matrix Square Root of P]
    B --> C[Generate 2n+1 Sigma Points _i]
    C --> D[Propagate points through Nonlinear Dynamics: _new = f _i]
    D --> E[Reconstruct Predicted Mean: _pred = Σ W_i _new]
    E --> F[Reconstruct Predicted Covariance: P_pred = Σ W_i diff^2]
    F --> G{Sensor Measurement Available?}
    G -- Yes --> H[Transform Points through Observation Model: Z_i = h _new]
    H --> I[Compute Innovation Covariance and Cross-Correlation]
    I --> J[Update and P via Kalman Gain]
    G -- No --> K[Output Propagated Analytic Gaussian PDF]
```

### Variations and Hybrid Architectures

The sources identify several advanced iterations of the UT designed to overcome specific numerical or statistical limitations:

- **Square-Root Unscented Kalman Filter (SR-UKF)**  
[Van der Merwe and Wan (2001)]:
  - **Description:** Instead of propagating the full matrix  $\mathbf{P}$ , the SR-UKF propagates the Cholesky factor  $\mathbf{S}$  (where  $\mathbf{P} = \mathbf{S}\mathbf{S}^T$ ) directly.

- **Advantage:** It guarantees positive semi-definiteness of the covariance and offers superior numerical stability and computational efficiency ( $O(L^3)$  for state estimation).
- **Higher-Order Unscented Filter (HOUF)** [Tenne and Singh (2003)]:
  - **Description:** Expands the sigma-set to include additional layers of points to match input moments beyond the 4th order (e.g., up to the 12th order).
  - **Context:** This is essential for highly nonlinear regimes where standard UT truncation errors (which start at the 4th order) become significant.
- **Conjugate Unscented Transform (CUT)** [Adurthi et al. (2012)]:
  - **Description:** A non-product cubature rule that uses “Conjugate Axes” (combinations of principal axes) to satisfy higher-order moment equations (up to 8th order) with fewer points than Gauss-Hermite rules.
  - **Advantage:** It captures cross-moments (e.g.,  $E[X_i^2 X_j^2]$ ) that standard UT cannot, while avoiding the “curse of dimensionality”.
- **Gaussian Mixture Model-Unscented Transform (GMM-UT):**
  - **Description:** To overcome the Gaussian limitation, the UT is frequently used to propagate the individual kernels of a **GMM (GMM\_UT)**. The GMM-UT method operates on the intuition that while the global distribution may be highly non-Gaussian (e.g., a “banana” shape), the local regions of the probability mass remain nearly Gaussian for a longer duration under nonlinear flow.
  - **Methodology**

- \* **Sigma Point Propagation:** For each of the  $N_g$  components, a set of  $2n + 1$  deterministic sigma points is generated using the component’s mean  $\mathbf{x}_i$  and covariance  $\mathbf{P}_i$ .
  - \* **Independent Nonlinear Mapping:** These sigma points are passed through the high-fidelity nonlinear dynamics  $\mathbf{f}(\cdot)$ .
  - \* **Moment Reconstruction:** The transformed mean and covariance for *each* kernel are reconstructed using standard UT weighting rules, preserving second-order accuracy.
  - \* **Re-Synthesis:** The final non-Gaussian PDF is the re-weighted sum of these nonlinearly propagated Gaussian kernels with their corresponding weights  $\alpha_i$ .
- **AEGIS Framework:** In the **Adaptive Entropy-based Gaussian-mixture Information Synthesis (AEGIS)** method, the UT is employed to calculate “nonlinear entropy”. By comparing this to the linear entropy from LinCov, the system can detect when the dynamics have become too nonlinear and trigger a component split.
  - **Multi-Fidelity (MF) Advantage:** While the UT is an efficient “middle-ground,” the sources characterize Multi-Fidelity (MF) as superior for high-consequence non-Gaussian cases, as MF uses large-scale low-fidelity sampling corrected by high-fidelity samples to capture complex shapes that the UT’s single-Gaussian assumption cannot.

## Comparative Performance

The sources highlight the UT’s performance relative to other “State-of-the-Art” methods:

- **UT vs. Linearised Covariance (LinCov):** The UT is consistently more accurate than LinCov because it captures more of the distribution’s tails and accounts for higher-order nonlinear effects that first-order Taylor expansions ignore.

- **UT vs. Monte Carlo:** The UT is significantly faster than MC, requiring only 13 points for a standard 6D orbital state compared to the millions needed for MC to reach similar statistical convergence.

## Polynomial Chaos Expansions (PCE)

In the rigorous domain of orbital uncertainty propagation (UP), **Polynomial Chaos Expansion (PCE)** represents a powerful spectral decomposition technique that characterizes the evolution of state uncertainties through highly nonlinear dynamical systems. Unlike traditional linearization methods that assume Gaussianity, PCE provides a non-parametric framework capable of capturing the increasingly non-Gaussian “banana-shaped” probability density functions (PDFs) characteristic of long-term orbital motion.

### Theoretical Classification and Role

Within the methodological landscape, PCE is distinguished by its ability to map input uncertainties to output responses using a “black-box” approach that does not require alterations to existing high-fidelity solvers.

- **Beyond Linearization:** Unlike Linearized Covariance (LinCov), which relies on first-order Taylor expansions and assumes Gaussian distributions, PCE represents inputs and outputs using a series of approximations that capture **higher moments** of the probability density function (PDF), allowing it to represent non-Gaussian shapes.
- **Surrogate Modeling Paradigm:** PCE functions as a data-driven mathematical emulator, replacing expensive physical orbital propagators with a computationally cheap polynomial surrogate.
- **Spectral Accuracy:** For systems where the output depends smoothly on the input parameters, PCE can achieve **exponential convergence** rates, far exceeding the  $\mathcal{O}(1/\sqrt{N})$  convergence of standard Monte Carlo (MC).

## Mathematical Foundations of PCE

The PCE models a square-measurable stochastic output vector  $\mathbf{X}$  as a linear combination of multivariate polynomials that are orthogonal with respect to the joint PDF of the input uncertainties  $\xi$ .

### The PCE Surrogate Formula

The stochastic output at a future time  $t$  is approximated by the finite series expansion:

$$\hat{\mathbf{X}}(t, \xi) \approx \sum_{j=0}^P c_j(t) \Psi_j(\xi)$$

#### Terms Defined:

- $\hat{\mathbf{X}}(t, \xi)$ : The approximated Quantity of Interest (QoI) vector (e.g., the terminal orbital state).
- $\xi$ : A vector of independent standard random variables representing input uncertainties.
- $\Psi_j(\xi)$ : Multivariate orthonormal basis functions, typically formed as the product of univariate polynomials, selected according to the **Wiener-Askey scheme**.
- $c_j(t)$ : Deterministic spectral coefficients (weights) determined at the epoch of interest.
- $P$ : The number of terms in the expansion, determined by the polynomial order  $p$  and input dimension  $d$ .

### Expansion Order and the Curse of Dimensionality

The total number of coefficients  $P$  required for a given order  $p$  and dimension  $d$  is calculated as:

$$P + 1 = \frac{(p + d)!}{p!d!}$$

Here, a random variable or a stochastic solution is called **square-measurable** if its second moment is finite:

$$\mathbb{E}[\mathbf{X}^2] < \infty$$

This condition ensures that the PCE framework can represent the random solution as a series of orthogonal polynomials and that statistical moments (like mean and variance) can be computed analytically from the expansion coefficients.

This factorial growth highlights the primary limitation of classical PCE: the **curse of dimensionality**, where high-dimensional problems (e.g.,  $d > 10$ ) become computationally demanding due to the rapid growth of expansion terms.

## Orthogonality and Basis Selection

The basis functions are chosen according to the **Wiener-Askey scheme**, ensuring they are orthogonal relative to the input PDF,  $\rho(\xi)$ :

$$\langle \Psi_\alpha, \Psi_\beta \rangle = \int_{\Gamma} \Psi_\alpha(\xi) \Psi_\beta(\xi) \rho(\xi) d\xi = \delta_{\alpha\beta}$$

where  $\delta_{\alpha\beta}$  is the Kronecker delta. This property allows for the **analytical evaluation of statistical moments** directly from the spectral coefficients without additional sampling:

- **Mean:**  $\mu = c_0$ .
- **Variance:**  $\sigma^2 = \sum_{j=1}^P c_j^2 \|\Psi_j\|^2$ .

### Polynomial Families and Corresponding Input Distributions

Usually, the choice of polynomial family is dictated by the distribution of the input uncertainties:

- **Hermite Polynomials:** For Gaussian inputs.
- **Legendre Polynomials:** For Uniform inputs.

## Non-Intrusive PCE Methodology

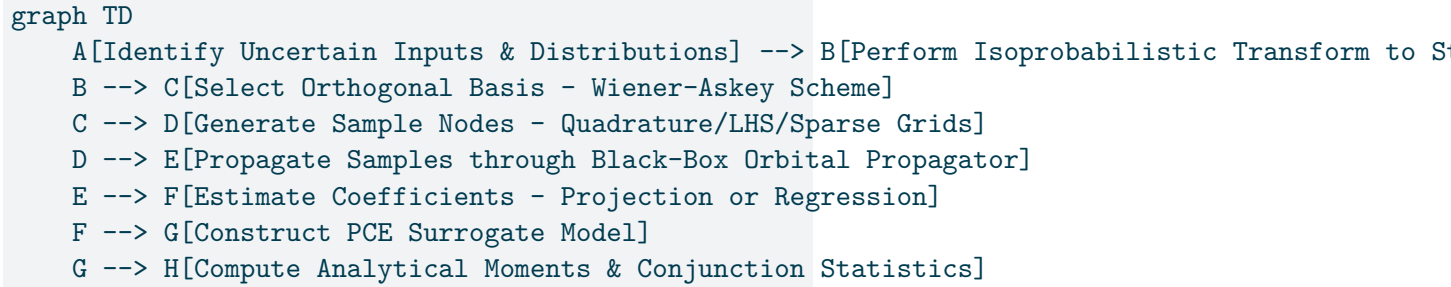
Non-intrusive methods are favored in astrodynamics because they treat high-fidelity propagators as “black boxes,” requiring no modification to existing legacy code.

## Methodology for Coefficient Estimation

NIPC relies on a design of numerical experiments to estimate coefficients. Two primary approaches are utilized:

1. **Pseudo-Spectral Projection:** Exploits basis orthogonality by projecting model output onto each basis function using numerical integration (e.g., Gaussian quadrature or Smolyak sparse grids).
2. **Least-Squares (LS) Regression:** Also known as point collocation, this method minimizes the residual between the expansion and model evaluations at a set of sampled points (e.g., Latin Hypercube Sampling).

## Flowchart: NIPC Algorithmic Workflow



## Intrusive PCE Methodology

Intrusive methods, or **Intrusive Spectral Projection (ISP)**, reformulate the original deterministic model equations into a larger system of coupled deterministic equations for the PC coefficients.

## Mathematical Framework: The Galerkin Projection

The core of the intrusive method is the **Galerkin projection**, which ensures the approximation error is orthogonal to the

functional space spanned by the basis. For an ODE system  $\dot{x} = f(x, t, \xi)$ , the PCE is substituted directly:

$$\sum_{j=0}^P \dot{c}_j \Psi_j = f \left( \sum_{j=0}^P c_j \Psi_j, t, \xi \right)$$

Multiplying by  $\Psi_k$  and taking the expectation  $\langle \cdot \rangle$  yields a system of  $(P + 1) \times n$  coupled deterministic ODEs.

## Procedural Logic for Intrusive PCE

Unlike NIPC, IPCE requires a complete “code rewrite” to implement polynomial algebra operators.

### Flowchart: IPCE Algorithmic Workflow

graph TD

```

A[Define Stochastic Governing Equations] --> B[Initialize Uncertain Variables as 1st Degree]
B --> C[Apply Galerkin Projection to Equations of Motion]
C --> D[Reformulate Force Models via Polynomial Algebra]
D --> E[Solve Large Coupled System of Deterministic ODEs]
E --> F[Reconstruct Terminal PDF from Coefficients]
```

## Variations of Expansion and Surrogate Methods

To address specific limitations—such as non-standard distributions, discontinuities, or high dimensionality—the sources identify several key variations of the PCE method.

- **Generalized Polynomial Chaos (gPC):** Built upon the **Wiener-Askey scheme**, gPC matches specific polynomial families to standard input distributions to ensure optimal convergence. For example, **Hermite polynomials** are optimal for Gaussian inputs, while **Legendre polynomials** are used for uniform inputs.
- **Arbitrary Polynomial Chaos (APC):** A **data-driven approach** that circumvents the need for a presumed parametric distribution. APC constructs an orthogonal basis using only a finite number of moments calculated directly

from raw sampling data (e.g., from an admissible region in orbit determination).

- **Sparse PCE and Compressive Sensing:** To combat the curse of dimensionality, modern frameworks exploit the **sparsity-of-effects principle**—the idea that systems are dominated by low-order interactions. Solvers such as **Orthogonal Matching Pursuit (OMP)**, **Least Angle Regression (LARS)**, or  $l_1$ -regularized **Compressive Sensing** identify only the most significant coefficients, allowing for accurate surrogates using significantly fewer function evaluations than terms in the full basis.
- **Multi-Element PCE (ME-PCE):** For systems with sharp gradients, discontinuities, or long-term integration errors, global polynomials often fail. ME-PCE utilizes **domain decomposition**, splitting the random input space into smaller sub-regions (elements) and fitting local PCE surrogates to each.
- **Hybrid Meta-Modeling (PC-Kriging):** PCK is a hybrid metamodeling technique that combines the global trend-capturing capabilities of PCE with the local interpolation accuracy of **Kriging** (Gaussian Process Regression).
  - **Formula:**  $M_{PCK}(X) = \sum_{i=1}^P \alpha_i \Psi_i(X) + \sigma^2 Z(X)$ .
  - **Methodology:** The PCE expansion serves as the trend function for Universal Kriging, and the Kriging part handles the local residuals. PCK often requires an order of magnitude fewer training points than standard Kriging or PCE to reach the same accuracy.
- **GMM-PCE Hybrid methods:** For cases with extreme nonlinearity or long propagation times, **Gaussian Mixture Models (GMMs)** are combined with PCE.
  - **Methodology:** The initial large Gaussian distribution is split into  $N$  smaller Gaussian sub-components (h-refinement). A lower-order PCE is then applied to each sub-component (p-refinement). This “hp-refinement” strategy reduces the domain of approximation for each element, allowing for rapid convergence where global PCE might fail.

- **Dimension Reduction (AS-NIPC & POD-PCE):** To handle high-dimensional state spaces (e.g., constellation sets or high-fidelity fields), dimension reduction techniques are integrated:
  - **Active Subspace (AS):** Identifies lower-dimensional directions in the input space that most influence the output, performing NIPC only in this active subspace.
  - **Proper Orthogonal Decomposition (POD):** Used for “field” outputs, POD identifies a latent space to reduce output dimensionality before mapping inputs to latent coordinates via PCE or PCK.
- **Multi-Fidelity (MF) Extensions:** Multi-Fidelity PCE incorporates information from a hierarchy of models. It uses a low-fidelity model (e.g., 2-body dynamics) as a foundation and a lower-order PCE to resolve the **model discrepancy** (additive or multiplicative) from a limited set of high-fidelity evaluations.

## Summary of Strengths and Limitations

| Feature                         | PCE Strength                                                                                              | Methodological Limitation                                                     |
|---------------------------------|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| <b>Statistical Analysis</b>     | Moments and <b>Sobol Indices</b> (sensitivity) are computed analytically.                                 | Requires square-measurable/finite variance stochastic solutions.              |
| <b>Computational Efficiency</b> | Massive speed-up relative to MC; can resolve collision risk with $<300$ evaluations vs. $10^5$ MC trials. | Significantly slower than linear approximations or the Multi-Fidelity method. |

| Feature                     | PCE Strength                                           | Methodological Limitation                                                         |
|-----------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------------|
| <b>Modeling Flexibility</b> | Non-intrusive; treats complex dynamics as a black box. | Standard global versions struggle with discontinuities or steep local variations. |

## Differential Algebra (DA)

In the larger context of **Polynomial and Expansion Methods**, the sources define **Differential Algebra (DA)** as a computational technique that facilitates the nonlinear propagation of uncertainties by substituting classical real algebra with **Taylor series expansions**. While related to other expansion techniques like Polynomial Chaos Expansions (PCE) and State Transition Tensors (STT), DA is distinguished by its ability to calculate higher-order derivatives automatically.

### Mathematical Foundations of Differential Algebra methods

The core methodology of DA involves substituting the classical real algebra of floating-point numbers with a truncated power series algebra, typically denoted as  ${}^kD_n$ , representing polynomials of order  $k$  in  $n$  variables.

- **The Algebra  ${}^kD_n$ :** This algebra consists of equivalence classes of functions that share the same Taylor polynomial up to order  $k$  in  $n$  variables centered at  $x_0$ .
- **State Initialization:** Instead of a single initial state  $\mathbf{x}_0$ , the state is initialized as a DA variable  $[\mathbf{x}] = \mathbf{x}_0 + \delta\mathbf{x}$ , where  $\delta\mathbf{x}$  represents a deviation or uncertainty around the nominal reference.
- **Automatic Derivation:** By overloading standard mathematical operators, any numerical integration scheme (e.g.,

Runge-Kutta) can be implemented to propagate the Taylor expansion of the flow automatically.

- **Coefficient Cardinality:** The maximum number of coefficients  $N_{coeff}$  required to represent a polynomial of order  $k$  in  $n$  variables is given by:

$$N_{coeff} = \frac{(n+k)!}{n!k!}$$

For instance, a 10th-order expansion in 6 dimensions (10D6) requires 8,008 coefficients.

- **The Taylor Map:** The primary output is a high-order Taylor polynomial map relating initial deviations to the final state.

Mathematically, the transformation is expressed as:

$$[\mathbf{y}] = f([\mathbf{x}]) = \mathcal{T}_y^k(\delta\mathbf{x}) = \sum_{p_1+\dots+p_n \leq k} c_{p_1\dots p_n} \cdot \delta x_1^{p_1} \dots \delta x_n^{p_n}$$

#### Terms Definition:

- $\mathcal{T}_y^k$ : The  $k$ -th order Taylor expansion of the dependent variable  $\mathbf{y}$ .
- $c_{p_1\dots p_n}$ : Taylor coefficients representing high-order partial derivatives of the state with respect to the initial conditions.
- $\delta x_i$ : The  $i$ -th component of the initial state deviation vector.
- $k$ : The expansion order, which determines the accuracy of the nonlinear capture.
- $\mathbf{y}$ : The dependent variable whose uncertainty is being propagated.

## Operational Methodology: High-Order Flow Expansion

The DA methodology facilitates nonlinear UP by propagating a “neighborhood” of states rather than a single point.

1. **DA Initialization:** The initial state  $x_0$  is initialized as a DA variable  $[x_0] = \bar{x}_0 + \delta x_0$ , where  $\bar{x}_0$  is the nominal mean and  $\delta x_0$  represents first-order variations.
2. **Polymorphic Integration:** The governing ordinary differential equations (ODEs),  $\dot{x} = f(x, t)$ , are integrated using standard schemes (e.g., Runge-Kutta) where every real-number operation is replaced by its adjoint operation in the space of Taylor polynomials.
3. **Map Extraction:** The result is the  $k$ -th order Taylor expansion of the flow  $\phi(t; x_0, \alpha)$ . This map is analytic and can be used for near-instantaneous evaluations of displaced initial conditions.
4. **Statistical Mapping:** Terminal moments are computed by applying the expectation operator  $\mathbb{E}[\cdot]$  to the resulting Taylor map.
  - **Terminal Mean:**  $\mu_i(t) = \sum_{|\alpha| \leq k} c_\alpha(t) \mathbb{E}[\delta x_0^\alpha]$ .
  - **Terminal Covariance:**  $P_{ij}(t) = \mathbb{E}[x_i(t)x_j(t)] - \mu_i(t)\mu_j(t)$ , which involves algebraic operations on the coefficients and initial moments.

## Algorithmic Implementation Flowcharts

### Standard DA Uncertainty Propagation

```
graph TD
    A[Initial PDF: Mean 0, Covariance P0] --> B[Initialize DA State: x_0 = 0 + x0]
    B --> C[Select Integration Scheme: e.g., RK45]
    C --> D[Propagate via DA Arithmetic: Substitute real ops with polynomial ops]
    D --> E[Obtain Terminal Taylor Map: T_tf]
    E --> F[Analysis Type?]
    F -- Analytic Moments --> G[Evaluate E[T_tf] and Covariance via Moment Mapping]
```

```

F -- Monte Carlo --> H[Sample  $x_0$  and Evaluate  $T_{tf}$  point-wise]
G --> I[Output Post-Propagation Statistics]
H --> I

```

## DA with Automatic Domain Splitting (ADS)

```

graph TD
    A[Uncertainty Volume:  $\Omega$ ] --> B[Propagate Taylor Map]
    B --> C[Estimate Truncation Error: Order  $k+1$  terms]
    C --> D{Error > Threshold?}
    D -- Yes --> E[Split Domain: Divide  $\Omega$  into  $\Omega_{left}$ ,  $\Omega_{right}$ ]
    E --> F[Propagate Patch of Local Maps]
    D -- No --> G[Continue Integration]
    F --> H[Reconstruct Patched Global Manifold]
    G --> H

```

## Variations and Algorithmic Enhancements

To overcome the limitations of local expansions and computational growth, several advanced variants of the DA framework exist.

- **Automatic Domain Splitting (ADS/LOADS):** In highly nonlinear regimes, a single Taylor expansion may fail to maintain accuracy over the entire uncertainty volume. ADS monitors the truncation error (e.g., the magnitude of terms of order  $k+1$ ) and adaptively subdivides the uncertainty domain into smaller sub-regions when error thresholds are exceeded.
- **DA-based Monte Carlo (DAMC):** Instead of thousands of expensive numerical integrations, DAMC samples from the initial PDF and evaluates the samples using the DA-derived Taylor map. This typically achieves a 10-fold reduction in computation time for order  $k=3$ .
- **Hybrid GMM-DA:** The initial uncertainty is split into a **Gaussian Mixture Model (GMM)**, and each kernel is nonlinearly propagated via a high-order DA map. This

approach is self-adaptive and avoids manual derivation of high-order derivatives.

- **DA-Based Higher-Order (DAHO):** This method analytically computes the moments (mean, covariance, skewness, kurtosis) of the transformed probability density function (PDF). This is achieved by computing the expectation of the monomials in the Taylor expansion.

For a Gaussian initial distribution, the expectation of monomials is computed using **Isserlis’ (Wick’s) Formula**:

$$E\{x_1^{s_1} \dots x_n^{s_n}\} = \begin{cases} 0 & \text{if } \sum s_i \text{ is odd} \\ Haf(\mathbf{P}) & \text{if } \sum s_i \text{ is even} \end{cases}$$

**Terms Definition:**

- $E\{\cdot\}$ : The expectation operator.
- $Haf(\mathbf{P})$ : The Hafnian of the covariance matrix  $\mathbf{P}$ , which is a sum over all permutations of product terms of the covariance components.
- **DA-LOADS Pruning:** Utilizes polynomial bounding techniques to discard regions of the state manifold that do not intersect with observations during initial orbit determination (IOD).
- **Multifidelity DA (MF-DA):**
  - **Methodology:** This approach combines low-fidelity (LF) analytical models (like SGP4) propagated in DA with high-fidelity (HF) point-wise numerical integrations. The LF DA propagation provides the nonlinear “shape” of the uncertainty, while HF integrations of the polynomial centers correct the absolute position accuracy.
  - **Efficiency:** This method can achieve speed-ups of 15–20x compared to full HF numerical DA integrations while maintaining high accuracy.

## Differential Algebra in the Expansion Taxonomy

Within the broader context of Expansion and Surrogate methods, DA is distinguished by its **intrusive nature** and **problem independence**.

- **DA vs. State Transition Tensors (STT):** STTs require the manual derivation and integration of complex high-order variational equations, which becomes prohibitively complex for high-fidelity models. DA automates this process by overloading the integrator’s basic operations, making it essentially ODE-independent.
- **DA vs. Polynomial Chaos (PCE):** PCE is a **non-intrusive** “black-box” spectral technique that characterizes uncertainty using a sum of orthogonal polynomials of random variables. In contrast, DA focuses on the Taylor expansion of the state and requires the system dynamics to be continuous and differentiable ( $C^{k+1}$ ).
- **Coordinate Frame Dependency:** Unlike methods restricted to specific coordinates to minimize nonlinearity, DA-based methods are largely unaffected by coordinate selection (e.g., Cartesian vs. Element space) because they can adjust expansion orders to maintain accuracy.
- **Handling Discontinuities:** Traditional DA requires continuous and differentiable dynamics. However, advanced implementations (like Thales’ PACE [Bignon et al. (2015)]) approximate discontinuities—such as shadow entry/exit in solar radiation pressure—using infinitely differentiable functions like the arctangent to remain within the DA framework.
- **The Curse of Dimensionality:** Like all expansion methods, DA suffers from factorial growth in terms ( $N_{coeff}$ ) as order  $k$  increases. Modern extensions like **Distribution Transport (DT)** attempt to mitigate this by organizing coefficients into vectors for parallel computation.

## Applications and Integration

DA is often used to enhance or complement other uncertainty quantification frameworks:

- **Enhancing Monte Carlo (MC):** DA can significantly speed up Monte Carlo simulations by replacing expensive numerical integrations with simple **evaluations of the Taylor-expanded dynamics**, maintaining accuracy while reducing computation time.
- **Multi-Fidelity (MF) Integration:** Researchers have developed methods combining **MF with DA and Gaussian Mixture Models (GMMs)**; in these frameworks, the nonlinearity index for the low-fidelity step is determined using Taylor expansions computed via DA.

## Requirements and Limitations

Despite its efficiency, DA has strict mathematical requirements that limit its general applicability:

- **Differentiability Requirement:** The governing physical equations must be both **continuous and differentiable**.
- **Discontinuity Barriers:** This requirement makes DA unsuitable for scenarios involving discontinuities, such as the step changes in solar radiation pressure that occur when a satellite enters or exits the Earth’s shadow.

## State Transition Tensors (STT)

In the rigorous taxonomy of orbital uncertainty quantification (UQ), **State Transition Tensors (STT)** represent a high-fidelity **semi-analytic expansion method**. Situated between first-order linear approximations like LinCov and non-intrusive surrogate models like PCE, STTs facilitate the nonlinear propagation of uncertainties by applying higher-order Taylor series

expansions to the deviation of a system's state from a nominal trajectory.

## Core Mechanics and Relationship to Linear Approximation

STT based methods function by analytically mapping initial uncertainties using expansions that describe nonlinear behavior within a limited region of the state space.

- **Extension of the STM:** The sources note that a first-order Taylor expansion in this framework corresponds exactly to solving for the **State Transition Matrix (STM)** used in the **Linearised Covariance (LinCov)** method (See Section ).
- **Handling Nonlinearity:** By solving for higher-order tensors, the method can provide significantly better results than standard linearisation for cases involving **strong nonlinearity**.
- **Algebraic Propagation:** Once the tensors are obtained by numerically integrating along a nominal trajectory, they provide a means to compute the nonlinear propagation of the first two statistical moments (mean and covariance) through relatively simple **algebraic operations**.

## Mathematical Architecture of STTs

The fundamental premise of the STT method is that the state deviation (that arises from the initial uncertainty of state)  $\delta\mathbf{x}(t)$  and its derivative  $\delta\dot{\mathbf{x}}(t)$  at a future time can be expressed as functions of the initial deviation  $\delta\mathbf{x}_0$  using a  $P$ -order Taylor series expansion about a nominal trajectory  $\mathbf{x}^*(t)$ .

## Nonlinear State Deviation

Equation 8 defines the deviation of a current state from a nominal trajectory as the difference between two solution flows of the system's equations of motion.

$$\delta \mathbf{x}(t) = (t; \mathbf{x}_0 + \delta \mathbf{x}_0, t_0) - (t; \mathbf{x}_0, t_0) \quad (8)$$

**Term Definitions:**

- $\delta \mathbf{x}(t)$ : The vector of state deviation at time  $t$ .
- $(t; \mathbf{x}_0, t_0)$ : The **solution flow** of the system's equations of motion, mapping the initial state  $\mathbf{x}_0$  at time  $t_0$  to the state at time  $t$ . It represents the explicit solution of the system's ordinary differential equations.

**! Solution Flow in Orbital dynamics**

The solution flow  $\phi(t; \mathbf{x}_0, t_0)$  is not found in closed form for most orbital dynamics problems due to the complexity and nonlinearity of the governing equations. This is especially true for high-fidelity models that include perturbations such as atmospheric drag, solar radiation pressure, and third-body effects. Instead, the state at time  $t$  is typically obtained through numerical integration of the equations of motion starting from the initial state  $\mathbf{x}_0$  at time  $t_0$ . Hence, in orbital dynamics the solution flow is treated as a representation of the numerically propagated state. STT leverages the existence of such a solution flow to perform Taylor expansions around the nominal trajectory, allowing for the analytical propagation of uncertainties without requiring a closed-form solution.

- $\mathbf{x}_0$ : The nominal trajectory initial state.
- $\delta \mathbf{x}_0$ : The initial deviation or perturbation from the nominal state.

The deviation vector also follows the dynamics of the system (of the form  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ ), which is expressed as the following differential equation:

$$\delta \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}^* + \delta \mathbf{x}, t) - \mathbf{f}(\mathbf{x}^*, t) \quad (9)$$

The superscript  $\star$  indicates that the nominal trajectory  $\mathbf{x}^*(t)$  is used as the reference point for evaluating the dynamics.

## Taylor Expansion of the Solution Flow

To arrive at Equation 10, we perform a multivariate Taylor series expansion of the perturbed solution flow  $\phi(t; \mathbf{x}_0 + \delta\mathbf{x}_0, t_0)$  about the nominal initial conditions  $\mathbf{x}_0$ .

- **Step 1: Scalar Component Expansion** The  $i$ -th component of the perturbed state at time  $t$  can be expressed as a function of the initial state components:

$$x_i(t) = \phi_i(t; x_{0,1} + \delta x_{0,1}, \dots, x_{0,n} + \delta x_{0,n}, t_0)$$

- **Step 2: Multivariate Taylor Series Application** Expanding this function about the nominal initial state  $x_{0,j}$  yields:

$$x_i(t) = \phi_i(t; \mathbf{x}_0, t_0) + \left. \frac{\partial \phi_i}{\partial x_{0,k_1}} \right|_{\mathbf{x}_0} \delta x_{0,k_1} + \frac{1}{2!} \left. \frac{\partial^2 \phi_i}{\partial x_{0,k_1} \partial x_{0,k_2}} \right|_{\mathbf{x}_0} \delta x_{0,k_1} \delta x_{0,k_2} + \dots$$

The derivative terms here can be compactly represented using the **State Transition Tensor** notation, where the  $p$ -th order tensor  $\Phi_{i,k_1 \dots k_p}$  encapsulates the  $p$ -th partial derivatives of the solution flow with respect to the initial conditions.

### **i** State Transition Tensor Definition

$\Phi_{i,k_1 \dots k_p}$  is the  $p$ -th order **State Transition Tensor**, defined as the  $p$ -th partial derivative of the solution flow with respect to initial conditions:  $\frac{\partial^p x_i}{\partial x_{0,k_1} \dots \partial x_{0,k_p}}$ . The set  $\{k_1, k_2, \dots, k_p\}$  is a re-labelling of the original state components  $\{1, 2, \dots, n\}$  with one-to-one mapping. This notation allows the flexibility to choose  $k_j$  such that it reflects a specific ordering or a subset of state components of interest for the expansion, while still maintaining the generality of the tensor representation.

- **Step 3: Subtraction of the Nominal Flow** Substituting this expanded form back into Equation 8:

$$\delta x_i(t) = \left[ \phi_i(t; \mathbf{x}_0, t_0) + \sum_{p=1}^P \frac{1}{p!} \Phi_{i,k_1 \dots k_p} \delta x_{0,k_1} \dots \delta x_{0,k_p} \right] - \phi_i(t; \mathbf{x}_0, t_0)$$

The zeroth-order term  $\phi_i(t; \mathbf{x}_0, t_0)$  (the nominal trajectory) cancels out, leaving only the sum of the higher-order partial derivatives.

### The High-Order Taylor Map

The final expression relates initial deviations to future deviations through a sum of tensors:

$$\delta x_i(t) \approx \sum_{p=1}^P \frac{1}{p!} \Phi_{i,k_1 \dots k_p} \delta x_{0,k_1} \dots \delta x_{0,k_p} \quad (10)$$

#### Term Definitions:

- $\delta x_i(t)$ : The  $i$ -th component of the state deviation at time  $t$ .
- $\Phi_{i,k_1 \dots k_p}$ : The  $p$ -th order **State Transition Tensor**, see the note above.
- $\delta x_{0,k_j}$ : The  $k_j$ -th component of the initial state deviation.
- $P$ : The truncation order of the expansion (typically  $P = 2$  or  $3$ ).

Similar to state deviation, we can also derive the Taylor expansion for dynamics deviation caused by the initial uncertainty from Equation 9 via a similar pathway. This yields the following expression for the time derivative of the deviation:

$$\dot{\delta x}_i(t) = \sum_{p=1}^P \frac{1}{p!} \mathbf{f}_{i;l_1 \dots l_p}^* \delta x_{l_1} \dots \delta x_{l_p} \quad (11)$$

$$\mathbf{f}_{i;l_1 \dots l_p}^* = \left. \frac{\partial^p f_i}{\partial x_{l_1} \dots \partial x_{l_p}} \right|_{\mathbf{x}^*(t)}$$

We can also find  $\dot{x}_i(t)$  by differentiating Equation 10 with respect to time  $t$ , which yields:

$$\dot{\delta x}_i(t) \approx \sum_{p=1}^P \frac{1}{p!} \dot{\Phi}_{i;k_1 \dots k_p} \delta x_{0,k_1} \dots \delta x_{0,k_p} \quad (12)$$

Since the initial deviations  $\delta x_{0,k_j}$  are arbitrary (chosen from the initial uncertainty probability distribution), the time derivative does not affect them. Therefore, the time derivative of the deviation is expressed in terms of the time derivatives of the STTs.

### Differential Equations for Tensor Propagation

These STT's are not static; they evolve over time according to the system's dynamics. The evolution of the STTs is governed by a set of coupled ordinary differential equations (ODEs) derived from Equation 10, Equation 11 and Equation 12. The following steps outline the derivation of these ODEs:

1. Substituting Equation 10 into each  $\delta x_{l_j}$  in Equation 11, we obtain a new expression for  $\dot{\delta x}_i(t)$  in terms of the initial deviations  $\delta x_{0,k_j}$  and the STTs.
2. We then equate this new expression with Equation 12, which also expresses  $\dot{\delta x}_i(t)$  in terms of the initial deviations and the time derivatives of the STTs. The coefficients of the same monomials  $\delta x_{0,k_1} \dots \delta x_{0,k_p}$  on both sides of the equation must be equal for every set  $\{k_1, \dots, k_p\}$ , leading to a set of ODEs for the STTs.

An example derivation for the first and second-order STT is shown in Section . The resulting ODEs for the STTs are expressed as follows:

$$\begin{aligned} \dot{\Phi}_{i,a} &= f_{i,\alpha}^* \Phi_{\alpha,a} \\ \dot{\Phi}_{i,ab} &= f_{i,\alpha}^* \Phi_{\alpha,ab} + f_{i,\alpha\beta}^* \Phi_{\alpha,a} \Phi_{\beta,b} \end{aligned}$$

### Term Definitions:

- $f_{i,\alpha}^*, f_{i,\alpha\beta}^*$ : The first and second partial derivatives (Jacobian and Hessian) of the dynamics  $\mathbf{f}$  evaluated along the nominal trajectory.
- $\alpha, \beta$ : Dummy indices using Einstein summation notation.
- $\Phi_{\alpha,a}, \Phi_{\beta,b}$ : First-order STT components that couple into the second-order evolution.

## Methodology: Moment Propagation via Algebraic Mapping

Unlike sampling methods that propagate individual points, STT-based uncertainty propagation maps the initial statistical moments directly to the terminal state through simple algebraic operations.

1. **Reference Integration:** Solve the ODEs for the nominal state  $\mathbf{x}(t)$  and the tensors  $\phi$  up to the truncation order  $P$ . The value of  $P$  is chosen based on the nonlinearity index taken from the propagation of the deviation  $\delta\mathbf{x}(t)$ .

### 💡 Tip 1: Local Nonlinearity Index for STT Order Selection

Formula:

$$\eta_m(t, t_0) = \sup_{i=1, \dots, n} \sup_{k=1, \dots, N} \frac{|\Delta x_i^{(m)} - \Delta \bar{x}_i|}{|\Delta \bar{x}_i|}$$

where

- $\Delta x_i^{(m)}$  =  $i$ th component predicted by an  $m$ th-order STT,
- $\Delta \bar{x}_i$  =  $i$ th component obtained by direct nonlinear numerical integration,
- $N$  = sampled initial conditions,
- $n$  = state dimension.

In practice, the nonlinearity index is computed by propagating a set of initial deviations  $\delta\mathbf{x}_0$  through both the  $m$ -th order STT and a direct numerical integration of the full nonlinear dynamics. The relative error between these two

predictions is then evaluated to determine if the current order  $m$  is sufficient to capture the system's nonlinearity. If  $\eta_m$  exceeds a predefined threshold, a higher-order STT is computed to improve accuracy.

**2. Initial Moment Characterization:** The initial uncertainty is characterized by its statistical moments, which are computed from the Moment generating function (MGF) of the initial probability distribution, which as we said earlier, is typically assumed to be Gaussian. The MGF provides a convenient way to compute the moments of a known distribution, which are then used in the algebraic mapping.

**i Note 2: Moment Generating Function (MGF)**

For a multivariate random vector such that  $\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{P})$ , the MGF is defined as:

$$M_{\mathbf{x}}(\mathbf{t}) = E[e^{\mathbf{t}^T \mathbf{x}}] = \exp\left(\mathbf{t}^T \mu + \frac{1}{2} \mathbf{t}^T \mathbf{P} \mathbf{t}\right)$$

The moments of the distribution can be obtained by partially differentiating the MGF with respect to components of  $\mathbf{t}$  and evaluating at  $\mathbf{t} = 0$ . That is, the  $k$ -th order moment is given by:

$$E[x_1^{k_1} x_2^{k_2} \dots x_n^{k_n}] = \left. \frac{\partial^{k_1+k_2+\dots+k_n} M_{\mathbf{x}}(\mathbf{t})}{\partial t_1^{k_1} \partial t_2^{k_2} \dots \partial t_n^{k_n}} \right|_{\mathbf{t}=0}$$

For example, the first four moments of a Gaussian distribution are:

$$\begin{aligned} E[x_i] &= \mu_i \\ E[x_i x_j] &= \mu_i \mu_j + P_{ij} \\ E[x_i x_j x_k] &= \mu_i \mu_j \mu_k + \mu_i P_{jk} + \mu_j P_{ik} + \mu_k P_{ij} \\ E[x_i x_j x_k x_l] &= \mu_i \mu_j \mu_k \mu_l + \mu_i \mu_j P_{kl} + \mu_i \mu_k P_{jl} + \mu_i \mu_l P_{jk} + \mu_j \mu_k P_{il} + \mu_j \mu_l P_{ik} + \mu_k \mu_l P_{ij} + P_{ij} P_{kl} + P_{ik} P_{jl} + P_{il} P_{jk} \end{aligned}$$

**3. Statistical Mapping:** Use Equation 10 and the linearity of the expectation operator to compute the terminal mean

( $\hat{\mu}$ ) and covariance ( $\hat{\mathbf{P}}$ ):

$$\delta\mu_i(t) = E[\delta x_i(t)] = \sum_{p=1}^P \frac{1}{p!} \phi_{i,k_1 \dots k_p} \mathbb{E}[\delta x_{0,k_1} \dots \delta x_{0,k_p}]$$

$$P_{ij}(t) = E[\delta x_i \delta x_j] - \delta\mu_i \delta\mu_j = \sum_{p=1}^m \sum_{q=1}^m \frac{1}{p!q!} \Phi_{i;k_1 \dots k_p} \Phi_{j;l_1 \dots l_q} E[\delta x_{k_1} \dots \delta x_{k_p} \delta x_{l_1} \dots \delta x_{l_q}] - \delta\mu_i \delta\mu_j$$

where  $\delta\mu_i$  is the  $i$ -th component of the terminal mean vector and  $\mathbf{P}_{ij}$  is the  $(i, j)$ -th component of the terminal covariance matrix.

### Algorithm: The STT Uncertainty Pipeline

The following flowchart delineates the procedural execution of an STT-based uncertainty propagation campaign.

graph TD

```

A[Initial PDF: 0, P0] --> B[Generate Initial Moments: E[x(t) | x(0) = 0]]
B --> C[Set Expansion Order: P]
C --> D[Numerically Integrate: Nominal State + STM + Higher-Order STTs]
D --> E[Obtain Terminal Tensors at tf]
E --> F[Reconstruct Terminal Mean: _tf = Σ Tensors * Initial Moments]
F --> G[Reconstruct Terminal Covariance: P_tf = Σ f Tensors, P0]
G --> H[Check for Non-Gaussianity from Nonlinearity Index]
H -- Yes --> I[Reconstruct Higher Moments: Skewness & Kurtosis]
H -- No --> J[Increase Expansion Order: P] --> D
I --> K[Pc Calculation for Conjunction Assessment]

```

- Expectation of a random variable  $x(t)$  is defined as:

$$\mu = E[x(t)]$$

- Covariance of two random variables  $x(t)$  and  $y(t)$  is defined as:

$$P_{xy} = E[(x(t) - \mu_x)(y(t) - \mu_y)] = E[xy] - \mu_x \mu_y$$

### Variations and Algorithmic Enhancements

To overcome the “curse of dimensionality” and the high computational cost of calculating factorial-order partial derivatives, several advanced STT variants have been developed.

- **Directional State Transition Tensors (DSTT):** Proposed to reduce complexity by only capturing nonlinear effects along “sensitive” directions (e.g., the maximum stretch direction of the Cauchy-Green Tensor). This reduces the number of integrated variables from  $O(n^{P+1})$  to  $O(n^2 + nm^{P-1})$ , where  $m$  is the number of sensitive directions.
- **Time-varying Directional STT (TDSTT):** Improves upon the DSTT by integrating the eigenvalue-eigenvector pairs over time, allowing uncertainty analysis at any historical point rather than just a predefined final epoch.
- **Distribution Transport (DT):** Organizes polynomial coefficients into a vector form to facilitate parallel calculation using linear algebra libraries and integral correction methods, achieving speed-ups of up to 60 times over conventional STT implementations.
- **Reduced STT (RSTT):** Assumes two-body dynamics dominate and retains only the secular terms to decrease the integration burden for high-order approximations.
- **GMM-STT Hybrid Method:** Combines Gaussian Mixture Models with STTs. The initial uncertainty is split into small Gaussian components, each of which is propagated linearly or with low-order STTs to maintain validity over long durations.
- **Non-Gaussian STT Propagation:** Extends the method to handle arbitrary initial PDFs (e.g., uniform distributions for asteroid gravity fields) by leveraging Moment Generating Functions (MGFs) to calculate initial expectations.

## Contextual Limitations of STTs in Uncertainty Propagation

Despite their accuracy, STTs are bounded by specific constraints within the larger UP landscape:

- **Local Validity:** STTs are localized approximations valid only within the **Radius of Convergence** of the Taylor series; if uncertainties are too large, the expansion fails to reconstruct the dynamics.
- **Differentiability:** The method requires the underlying dynamics to be continuous and differentiable to compute Local Dynamics Tensors (LDTs).
- **Complexity:** Although the algebraic evaluation of the final state is fast, the method is described as **computationally intensive** during the initial phase because of the difficulty involved in deriving and integrating the high-order tensors for high-fidelity models (including  $J_2$ , SRP, and drag).

The primary takeaway is that STTs provide a **local higher-order approximation** of the flow manifold. While they accurately capture the “banana-shaped” warping caused by  $1/r^2$  gravitational nonlinearities, they are strictly limited to the **radius of convergence** of the Taylor expansion. If the initial uncertainty is too large, the expansion fails, necessitating hybrid methods such as **GMM-STT**, which partition the state space into smaller, locally linear Gaussian kernels.

Complexity is further compounded by the need to compute high-order derivatives of the dynamics, which can be prohibitive for high-fidelity models. Variations like **Directional STTs** and **Reduced STTs** have been developed to mitigate these challenges, but they still require careful consideration of the trade-offs between accuracy and computational efficiency.

## Directional State Transition Tensors (DSTT)

Within the larger context of orbital uncertainty propagation, **Directional State Transition Tensors (DSTT)** represent a specialized dimensionality-reduction framework derived from the high-order **State Transition Tensor (STT)** method, developed primarily by Boone and McMahon (2023). While STTs provide superior precision over linearized models by capturing localized nonlinear motion through Taylor series expansions

of the solution flow, they are fundamentally constrained by the “curse of dimensionality,” where the number of variables to be integrated grows exponentially with the expansion order. DSTTs mitigate this by leveraging the physical observation that system nonlinearity is often concentrated along a few “sensitive” or unstable directions.

## Theoretical Framework and Methodology

The DSTT method reformulates the standard STT expansion by performing an orthogonal transformation of the initial state space, allowing derivatives to be taken with respect to a rotated basis that isolates directions of maximum instability. These STTs can be rotated such that the derivatives are taken with respect to an alternative basis  $y \in R^n$  with no loss in information as long as  $y$  constitutes an orthogonal basis that spans  $R^n$ . These bases are related through the transformation  $y = Rx$ , where  $R$  corresponds to a linear transformation matrix.

### The Polynomial Approximation

The state deviation at a future time  $t_f$  is approximated using the DSTT mapping:

$$\delta x_i(t_f) \approx \phi_{i,\kappa_1} \delta x_{\kappa_1 0} + \sum_{p=2}^P \frac{1}{p!} \varphi_{i,\gamma_1 \dots \gamma_p}(t_f, t_0) \delta y_{\gamma_1 0} \dots \delta y_{\gamma_p 0}$$

### Term Definitions:

- $\delta x_i$ : The  $i$ -th component of the final state deviation.
- $\phi_{i,\kappa_1}$ : The first-order State Transition Matrix (STM).
- $\varphi_{i,\gamma_1 \dots \gamma_p}$ : The  $p$ -th order DSTT elements.
- $\delta y_{\gamma_j 0}$ : The initial state deviation projected onto the rotated basis  $y = Rx$ .
- $P$ : The maximum truncation order of the Taylor series.

## Methodology of the Transformation

The construction of the DSTT relies on a linear transformation matrix  $R \in \mathbb{R}^{m \times n}$ , where  $m$  is the number of sensitive directions (typically  $m \ll n$ ). Each row of  $R$  corresponds to a sensitive direction, identified using the eigenvectors of the **Cauchy-Green Tensor (CGT)**.

### **i** Cauchy-Green Tensor (CGT)

The Cauchy-Green Tensor is a symmetric, positive-definite matrix that characterizes the local deformation of a flow. It is computed as the product of the transpose of the State Transition Matrix (STM) and the STM itself:

$$C(t_f, t_0) = \Phi(t_f, t_0)^T \Phi(t_f, t_0)$$

where  $t_f$  is the final time and  $t_0$  is the initial time. The eigenvalues of the CGT indicate the amount of stretching along each eigenvector direction, with larger eigenvalues corresponding to more sensitive directions. From Equation 5, we understand the STM ( $\Phi$ ) is a linear mapping of the final perturbation vector  $\delta x(t_f)$  to the initial perturbation vector  $\delta x(t_0)$ . The CGT is a squaring operation that returns the relationship between the initial and final distance from the nominal motion.

$$||\delta x(t_f)||^2 = \delta x(t_0)^T C(t_f, t_0) \delta x(t_0)$$

The distance  $||\delta x(t_f)||^2$  reveals the growth or the decay of the initial vector perturbation. Furthermore, the eigendecomposition of the CGT allows identification of the principal directions of expansion for the dynamical flow, including those that are locally associated with the largest and smallest stretching.

The  $p$ -th order DSTT elements are obtained by rotating the full STT ( $\phi$ ) onto this basis:

$$\varphi_{i, \gamma_1 \dots \gamma_p} = \phi_{i, k_1 \dots k_p} R_{\gamma_1, k_1} \dots R_{\gamma_p, k_p}$$

By choosing the eigenvector  $\xi_1$  associated with the largest eigen-

value  $\lambda_1$  of the CGT, the DSTT captures the “direction of maximum stretch”. This ensures that the terms which contribute most significantly to the final state deviation are preserved, while stable directions are ignored.

---

## Advantages over the Standard STT Method

- **Polynomial Complexity Growth:** A full  $p$ -th order STT contains  $n^{p+1}$  elements, resulting in a total integration burden of  $\sum_{p=1}^P n^{p+1}$  variables. In contrast, the DSTT reduces this to  $n + n^2 + \sum_{p=2}^P nm^p$  variables.
  - **Numerical Efficiency:** For a standard 6-dimensional orbital state ( $n = 6$ ) with one sensitive direction ( $m = 1$ ) at second order ( $P = 2$ ), the STT requires integrating 258 variables, whereas the DSTT requires only 48.
  - **Computational Speedup:** Numerical studies indicate that DSTTs can achieve up to **1000× speed improvements** in nonlinear uncertainty propagation compared to full STTs.
  - **Reduced Storage:** Because only the elements along unstable directions are stored, the method is significantly more suitable for real-time applications on on-board spacecraft computers with limited resources.
  - **Analytical Tractability:** DSTTs allow for the derivation of simplified, closed-form equations for propagated mean and covariance, bypassing the hundreds of terms required by Isserlis’ theorem in full higher-order covariance propagation.
-

## Variations and Enhancements

### Time-Varying Directional STT (TDSTT)

A major limitation of the standard DSTT is that it assumes sensitive directions are constant and is only valid at a predefined final epoch  $t_f$ . The **TDSTT** [Zhou et al. (2024)] improves this by simultaneously integrating the nominal orbit, the Taylor expansion, and the time-varying eigenvalue-eigenvector pairs. This allows for uncertainty analysis at any point along the trajectory (on a dense time grid), which is critical for **conjunction assessment** where the time of closest approach (TCA) is not known *a priori*.

### Methodology

The TDSTT’s primary advantage is its ability to perform uncertainty analysis on a **dense time grid** without re-integration.

- **Warm Start Strategy:** At the initial epoch  $t_0$ , the CGT is an identity matrix with repeated eigenvalues, causing singularity in the derivative equations. The methodology employs a “warm start” by integrating the full STT for a very small step (e.g., 1/100,000 of the period) to obtain a non-singular CGT and its associated sensitive basis.
- **Augmented ODE System:** The TDSTT methodology integrates the nominal state, the STM, the eigenvalues/eigenvectors, and the high-order TDSTT terms simultaneously.
- **Complexity Reduction:** By utilizing the orthogonality of the sensitive directions, the complexity of computing the second-order TDSTT derivative is reduced from  $O(n^4m^2)$  to  $O(m)$ .

### Derivatives of Eigenvalue-Eigenvector Pairs

The integration of TDSTT requires calculating the time evolution of the sensitive basis. The derivative of the  $k$ -th eigenvalue

$\lambda_k$  of the CGT is derived as:

$$\dot{\lambda}_k = (\dot{\mathbf{C}}\xi_k, \xi_k)$$

**Term Definitions:**

- $\dot{\lambda}_k$ : The time derivative of the  $k$ -th eigenvalue.
- $\dot{\mathbf{C}}$ : The time derivative of the Cauchy-Green Tensor.
- $\xi_k$ : The  $k$ -th eigenvector of the CGT.

The derivative of the  $k$ -th eigenvector  $\xi_k$  is represented linearly as:

$$\dot{\xi}_k = \sum_{p \neq k} B_{k,p} \xi_p$$

**Term Definitions:**

- $B_{k,p}$ : The coefficients of the eigenvector derivative, where  $B_{k,p} = \frac{1}{\lambda_p - \lambda_k} (\dot{\mathbf{C}}\xi_k, \xi_p)$ .

**Derivation of the TDSTT ODEs**

The governing ODE for a second-order TDSTT includes rotation terms ( $B$ ) representing the time derivative of the eigenvectors:

$$\dot{\varphi}_{i,p_1 p_2} = A_{i,\alpha} \varphi_{\alpha,p_1 p_2} + A_{i,\alpha\beta} \varphi_{\alpha,p_1} \varphi_{\beta,p_2} + \varphi_{i,\gamma_1 p_2} B_{p_1,\gamma_1} + \varphi_{i,p_1 \gamma_2} B_{p_2,\gamma_2}$$

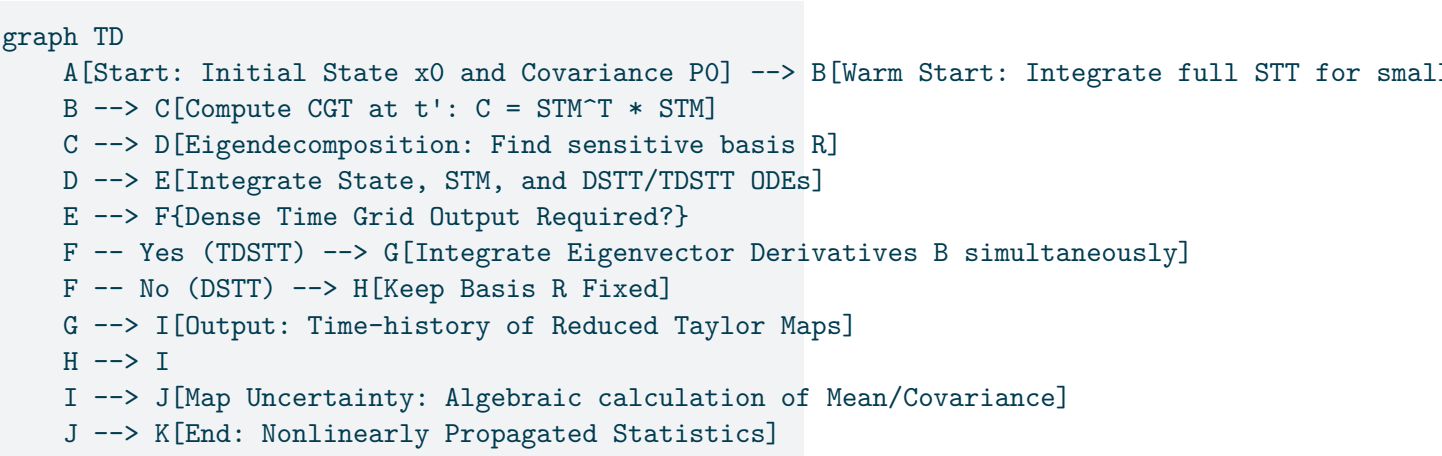
**Term Definitions:**

- $A_{i,\alpha}, A_{i,\alpha\beta}$ : Local dynamics tensors (Jacobian and Hessian).
- $B_{p_j,\gamma_j}$ : Coefficients derived from eigenvector derivatives.

**Multi-Directional Selection**

While many applications use only the primary sensitive direction ( $m = 1$ ), some scenarios involve complex dynamics where multiple directions are unstable. The DSTT framework supports  $R$  matrices built from the first  $k$  eigenvectors, balancing the tradeoff between accuracy and computational cost.

### Algorithm Flowchart for DSTT/TDSTT Implementation



(Ref: Boone and McMahon (2023) )

Summary Table: Contextual Positioning

| Feature                                 | State Transition Tensors (STT) | Directional STT (DSTT)      | Time-Varying DSTT (TDSTT)     |
|-----------------------------------------|--------------------------------|-----------------------------|-------------------------------|
| <b>Expansion Basis</b>                  | Cartesian coordinates          | Fixed unstable direction(s) | Dynamic unstable direction(s) |
| <b>Complexity</b><br>( $n = 6, P = 2$ ) | 258 variables                  | 48 variables                | 55 variables                  |
| <b>Epoch Flexibility</b>                | Valid at all points            | Predefined epoch only       | Valid at all points           |
| <b>Best Use Case</b>                    | Low-dimensional systems        | Known orbits (NRHO/Halo)    | New targets / Conjunctions    |

## Reduced State Transition Tensors (RSTT)

The **Reduced State Transition Tensors (RSTT)** method, primarily developed by Roa and Park (2021), serves as a strategic complexity-reduction technique. While standard **State Transition Tensors (STTs)** provide superior precision over linear models by capturing localized nonlinear motion through Taylor series expansions, they suffer from an exponential growth in computational burden as the expansion order increases. RSTT addresses this by isolating and capturing the **dominant secular terms** responsible for the growth of relative separation between neighboring trajectories.

### Theoretical Foundation: Stabilizing the Solution Flow

The fundamental methodology of RSTT is built on the premise that the primary instability in orbital motion—the Lyapunov instability—is relegated to the time domain, specifically the phasing of the spacecraft along its orbit.

- **Sundman Transformation:** To isolate secular trends, RSTT employs a time transformation (specifically a Sundman transformation of order  $\beta = 2$ ) to switch the independent variable from physical time  $t$  to the true anomaly increment  $\theta$ :

$$\frac{d\theta}{dt} \equiv \gamma(\mathbf{x}, t) = \frac{h}{r^2}$$

#### Term Definitions:

- $\theta$ : The true anomaly increment from the initial epoch  $t_0$ .
  - $\gamma$ : The angular frequency of the motion.
  - $h$ : The magnitude of the specific angular momentum.
  - $r$ : The magnitude of the spacecraft position vector.
- **Decomposition of the Integral Flow:** By transcribing the solution into the  $\theta$  domain, the state vector at a future time can be expressed as a stabilized solution  $G(\theta; \mathbf{x}_0)$  corrected by a time-phasing term  $\Theta(t; \mathbf{x}_0)$ :

$$\mathbf{x}(t) = \mathbf{F}(t; \mathbf{x}_0) = \mathbf{G}(\Theta(t; \mathbf{x}_0); \mathbf{x}_0)$$

This decomposition allows the partial derivatives of the flow (the tensors) to be split into a stabilized component and a secular correction component.

---

## Methodology for RSTT Approximation

The RSTT methodology reduces complexity by retaining only the highest-order secular terms (those growing like  $t^p$  for a  $p$ -th order tensor) while neglecting periodic and lower-order secular components.

### 1. First-Order Correction (The Secular STM)

The standard State Transition Matrix (STM)  $\Phi$  can be decomposed into a stabilized derivative and a first-order secular correction:

$$\Phi_{i;j} = \left. \frac{\partial G_i}{\partial x_{0,j}} \right|_{\theta} + \Gamma_j \frac{\partial G_i}{\partial \theta}$$

#### Term Definitions:

- $\Gamma_j$ : The partial derivative of the time transformation  $\Theta$  with respect to initial conditions  $\partial\Theta/\partial x_{0,j}$ .
- $\partial G_i/\partial\theta$ : The tangent vector to the orbit (related to velocity).

### 2. The Second-Order RSTT Approximation

The core “reduction” occurs at the second order. Instead of integrating 216 individual elements for an  $n = 6$  state, the second-order RSTT is approximated as:

$$\Phi_{i;jk} \approx \frac{r^4}{h^2} \Gamma_j \Gamma_k \frac{\partial^2 F_i}{\partial t^2}$$

#### Term Definitions:

- $\Phi_{i,jk}$ : The components of the second-order STT.
- $\Gamma_j, \Gamma_k$ : Sensitivities of the true anomaly to initial state variations.
- $\partial^2 F_i / \partial t^2$ : The time derivative of the dynamics (the acceleration/force function).

### 3. Generalization to Higher Orders

Higher-order STTs ( $p \geq 3$ ) are similarly reduced to a product of  $\Gamma$  vectors and higher-order time derivatives of the flow, growing like  $t^p$ :

$$\Phi_{i;k_1 \dots k_p} \approx \frac{1}{\gamma^p} \Gamma_{k_1} \dots \Gamma_{k_p} \frac{\partial^p F_i}{\partial t^p}$$


---

### Algorithm for RSTT-Based Uncertainty Propagation

The following flowchart illustrates the implementation of RSTT for nonlinear covariance propagation.

```
graph TD
  A[Start: Initial Mean m0 and Covariance P0] --> B[Initialize Nominal State and STM]
  B --> C[Integrate Dynamics and Variational Equations: dx/dt, dSTM/dt]
  C --> D[Integrate Time-Sensitivity Equations: dGamma/dt]
  D --> E[Obtain Time-Phasing Gradient Gamma at tf]
  E --> F[Construct RSTT Coefficients via Secular Approximation: Eq. 31]
  F --> G[Map Statistics: Algebraic Summation for Mean and Covariance]
  G --> H[Result: Propagated Stats at tf]
```

---

## Advantages and Contextual Variations

RSTT exists as a middle ground between the accuracy of full STTs and the efficiency of directional methods.

- **Computational Efficiency:**
  - **Variable Reduction:** For a second-order expansion, RSTT requires integrating only **6 additional differential equations** (the elements of  $\Gamma$ ) in addition to the state and STM. This is a drastic reduction from the 216 elements required for a full second-order STT.
  - **Speedup:** Numerical tests in LEO scenarios indicate that RSTT reduces the computational time required to compute the STT by approximately **37%** while maintaining comparable long-term accuracy.
- **Accuracy Convergence:** While full STTs are more accurate in the short term (due to periodic and linear secular terms), the RSTT approximation converges to the exact STT solution as time advances and the  $t^p$  secular terms become dominant.
- **Comparison with Other Methods:**
  - **RSTT vs. Full STT:** RSTT avoids the “curse of dimensionality” and the manual derivation of complex Hessian/third-order Local Dynamics Tensors.
  - **RSTT vs. DSTT (Directional STT):** While DSTT reduces dimensionality by aligning with “sensitive directions” of the Cauchy-Green Tensor, RSTT reduces complexity by isolating the “secular time drift” common to all orbital perturbations.
  - **RSTT vs. UT (Unscented Transform):** RSTT-based filters (like the ASEKF) can be several times faster than the UKF because they replace sigma-point sampling with a single algebraic correction.

## Variations on the RSTT Method

- **ASEKF (Approximate Second-Order EKF):** A high-order Kalman filter that utilizes the RSTT approximation for the prediction step. It yields results comparable to the UKF but with roughly **50% faster** computation than an EKF using exact STTs.
- **Analytical RSTT for Two-Body Problems:** In purely Keplerian regimes, the  $\Gamma$  vector and time derivatives can be expressed in closed form, allowing for instantaneous non-linear UP without any numerical integration.
- **Vectorized DT (Distribution Transport):** A related evolution that uses the same Taylor foundation but transforms the tensor structure into a fixed **vector structure**, enabling parallel computing speedups of 14x to 64x.

## Event Transition Tensors (ETT)

In the field of high-order uncertainty propagation (UP), **Event Transition Tensors (ETTs)** represent a significant extension of the **State Transition Tensor (STT)** framework. While standard STTs are parametric tools used to map state deviations and statistical moments to a predefined future time, ETTs provide a rigorous mathematical description of system behavior on **differentiable event manifolds**.

This methodology is critical for aerospace applications where the epoch of interest is defined by a physical condition (e.g., atmospheric entry, planetary landing, or a surface of section crossing) rather than a fixed timestamp.

### 1. Theoretical Foundation: From Solution Flows to Event Maps

To understand ETTs, one must distinguish between the standard solution flow of an ordinary differential equation (ODE) and the specialized event map.

- **The Solution Flow ( $\Phi$ ):** In the STT framework, the flow represents all possible trajectories generated by varying initial conditions  $\mathbf{x}_0$  and parameters  $\theta$  over a fixed time  $t$ . The coefficients of the Taylor expansion of this flow are the STTs.
- **The Event Manifold ( $e$ ):** An event is implicitly defined by the condition  $e(\mathbf{x}, \theta, t) = 0$ , representing a target surface or a specific state condition.
- **The Event Map ( $\mathcal{F}$ ):** The event map is defined as the flow of the dynamical system evaluated at the **trigger time**  $t^*$ , where  $t^*$  is the time at which the trajectory intersects the manifold.
- **The Definition of ETTs:** ETTs are the coefficients of the multivariate Taylor expansion of the event map  $\mathcal{F}$ . Mathematically, if the order of the expansion is  $k$ , the state on the event manifold  $\mathbf{x}_e$  is approximated as:

$$\delta x_{ie} \approx \sum_{1 \leq |\alpha| \leq k} \frac{1}{\alpha!} [\partial^\alpha \mathcal{F}_i(\mathbf{x}_0, \theta)] \delta \mathbf{z}^\alpha$$

#### Term Definitions:

- $\delta x_{ie}$ : The  $i$ -th component of the state deviation on the event manifold.
- $\alpha$ : An  $n$ -tuple multi-index representing the partial derivative order.
- $\delta \mathbf{z}$ : The vector combining initial state perturbations  $\delta \mathbf{x}_0$  and parameter deviations  $\delta \theta$ .

## 2. Methodology: Computation via Augmented Systems and Inversion

The methodology for computing ETTs involves transforming a variable-time event problem into a fixed-time integration problem through the use of differential algebra or variational equations.

- **Step 1: System Augmentation** The system of ODEs  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \theta, t)$  is augmented with the derivative of the event function  $\epsilon = e(\mathbf{x}, \theta, t)$ . To recast the problem for a fixed integration interval, a normalized time variable  $\tau = t/T$  is introduced, where  $T$  is the unknown event trigger time.

$$\begin{cases} \dot{\mathbf{x}} = T\mathbf{f}(\mathbf{x}, \theta, \tau) \\ \dot{\epsilon} = T \left( \frac{\partial e}{\partial t} + \nabla_{\mathbf{x}} e \cdot \mathbf{f} \right) \end{cases}$$

- **Step 2: Integration of Variational Equations** By integrating the  $k$ -th order variational equations of this augmented system up to  $\tau = 1$ , one obtains a polynomial representation of the final state  $\delta\mathbf{x}_f$  and the event variation  $\delta\epsilon_f$  as functions of initial perturbations and the time correction  $\delta T$ .
- **Step 3: Map Inversion** Utilizing the **implicit function theorem**, the polynomial system is partially inverted. By forcing the final event variation  $\delta\epsilon_f$  to zero, the variation in the trigger time  $\delta T$  is expressed as a function of the initial perturbations. Substituting this result back into the state polynomial yields the ETTs, which define the system state constrained strictly to the event manifold.

### 3. Algorithm Flowchart for ETT Propagation

graph TD

```

A[Start: Initial State x0, Parameters theta, and Event e=0] --> B[Augment System: Add Event]
B --> C[Normalize Time: tau = t/T]
C --> D[Integrate k-th Order Variational Equations to tau=1]
D --> E[Obtain Polynomials for State xf and Event Variation eps_f]
E --> F[Apply Map Inversion: Swap eps_f and Delta T]
F --> G[Constraint: Set eps_f = 0 to solve for Delta T]
G --> H[Substitute Delta T into State Polynomial]
H --> I[Result: Event Transition Tensors on Manifold]
I --> J[Map Statistics: Compute Propagated Moments at Event]

```

#### 4. Variations and Comparative Context

ETTs exist in a continuum with STTs and other high-order methods, with several variations depending on the application:

- **ETTs vs. STTs:** If the event manifold is defined as a reaching a predefined constant time  $T$ , the ETTs collapse into standard STTs.
- **Poincaré Maps:** In periodic systems, ETTs are used to construct discrete Poincaré maps [Poincaré return maps]. This is done by defining the surface of section (e.g.,  $x_{jf} - \bar{x}_{j0} = 0$ ) and inverting the map to find the period correction  $\delta t^*$  required for the return.
- **NeuralODEs expansion:** A novel variation involves using ETTs to analyze **Neural Ordinary Differential Equations**. Because NeuralODEs often function as “black-box” systems, ETTs provide a way to certify their reliability by expressing their complex learned dynamics through explicit, high-order mathematical structures on event boundaries.
- **Non-Gaussian Handling:** ETTs can be combined with **Moment Generating Functions (MGFs)** to propagate non-Gaussian uncertainties (e.g., uniform distributions for spacecraft mass or position) to an event manifold without requiring Monte Carlo sampling.

#### 5. Implementation Advantages

- **Computational Efficiency:** Mapping statistical moments directly to an event manifold eliminates the need for “stopping” numerical integrators for each sample in a Monte Carlo run.
- **Rigorous Certification:** In safety-critical missions, such as landing a quadcopter through a gate or a spacecraft on an asteroid, ETTs allow for the analytical verification of performance requirements (e.g., relative velocity  $< 0.5$  m/s in 99% of cases) by bounding all possible configurations on the manifold.

- **Automation:** Modern pipelines, such as the open-source **heyoka** project, have automated the symbolic derivation and integration of these tensors, making the manual derivation of high-order event sensitivities unnecessary.

## Gaussian Mixture Models (GMM) Framework

In the larger landscape of **Uncertainty Propagation (UP)**, **Gaussian Mixture Models (GMM)** serve as a versatile “mixture model” framework that bridges the gap between computationally efficient linear methods and the complex, non-Gaussian realities of orbital mechanics. The foundational principle of the GMM framework is the “universal approximator” property, which asserts that any arbitrary probability density function (PDF) can be represented with sufficient accuracy as a weighted sum of multiple Gaussian kernels. This property allows GMMs to capture the intricate shapes of non-Gaussian distributions, such as the “banana” or “kidney” forms that emerge in long-term propagation through the highly nonlinear dynamics of orbital mechanics.

### Mathematical Architecture of the GMM Framework

The GMM framework represents a complex, non-Gaussian state  $\mathbf{x}$  as a linear combination of  $N_g$  multivariate Gaussian components.

#### The Joint GMM PDF

The non-Gaussian PDF of a state vector  $\mathbf{x}$  is defined as:

$$p(\mathbf{x}) = \sum_{i=1}^L \alpha_i \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_i, \mathbf{P}_i)$$

#### Terms Defined:

- $p(\mathbf{x})$ : The total non-Gaussian probability density function of the state.

- $L$ : The total number of Gaussian mixture elements (GMEs).
- $\alpha_i$ : The scalar weight of the  $i$ -th component, where  $0 \leq \alpha_i \leq 1$  and  $\sum_{i=1}^L \alpha_i = 1$ .
- $\mathcal{N}(\cdot)$ : The  $i$ -th multivariate Gaussian kernel, defined as:

$$\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_i, \mathbf{P}_i) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}_i|}} \exp \left( -\frac{1}{2} (\mathbf{x} - \hat{\mathbf{x}}_i)^T \mathbf{P}_i^{-1} (\mathbf{x} - \hat{\mathbf{x}}_i) \right)$$

Where,

- $\hat{\mathbf{x}}_i$ : The mean state vector of the  $i$ -th Gaussian component.
- $\mathbf{P}_i$ : The covariance matrix of the  $i$ -th Gaussian component.

## General Methodology for GMM Propagation

The propagation of uncertainty via GMMs typically follows a three-stage “split-propagate-recombine” paradigm.

- **Initial Splitting:** An initial Gaussian distribution (often from Orbit Determination) is replaced by  $L$  components. The effectiveness of this stage is highly dependent on the choice of splitting directions and the number of components, which are influenced by the algorithms used to initialize and manage the mixture components:
  - **Fixed vs. Adaptive Splitting:**
    - \* **Static Splitting:** The number of components is fixed at the initial epoch. This is common in the **Multidirectional GMM (MGMM)** approach, where a regular grid is formed by recursively applying the splitting library along multiple nonlinear directions.

- \* **Adaptive Splitting (AEGIS/AGM):** Splits are triggered online when a nonlinearity threshold is reached. The **Adaptive Entropy-based GMM (AEGIS)** monitors the deviation between linear and nonlinear differential entropy estimates to trigger splits.
- **The Horwood Suboptimal Algorithm:** This specific technique refines a multivariate Gaussian along a single direction, which is computationally faster than multi-directional splitting without significantly compromising accuracy.
- **Variance-Preserving (Homoscedastic) Splitting:** While homoscedastic libraries often discard skewness, variance-preserving libraries are optimized under the constraint of matching the first two moments of the original distribution exactly. This ensures that the global mean and covariance of the mixture are identical to the original Gaussian before any nonlinear propagation occurs.
- **Multidirectional GMM (MGMM):** This approach recursively applies the splitting library along multiple nonlinear directions, forming a regular grid of components. While this can capture complex non-Gaussian features, it is computationally expensive and may lead to an exponential increase in the number of components.
- **Coordinate Selection:** Splitting in Cartesian space is often inefficient because nonlinearity acts in all six dimensions. Instead, curvilinear coordinates (e.g., Equinoctial or Delaunay elements) are used, as nonlinearity is primarily restricted to the mean longitude direction, allowing for a 1D split that avoids the “curse of dimensionality”.
- **Component-wise Propagation:** Each component  $(\hat{\mathbf{x}}_i, \mathbf{P}_i)$  is evolved independently through the nonlinear dynamics using a chosen propagation method.
- **Recombination:** At the target epoch, the weighted sum of components provides the final non-Gaussian PDF for

tasks like Conjunction Assessment (CA).

## Algorithmic Methodology of Gaussian Mixture Modelling of uncertainty

The following flowchart illustrates the procedural logic for propagating orbital uncertainty using the GMM framework. Here we use the nonlinearity metric to pre-determine the splitting direction and the number of components to generate.

```
graph TD
    A[Initial Gaussian State & Covariance] --> B{Select Coordinate System}
    B -- Cartesian --> C[Calculate Nonlinearity Metric Stirling Interpolation]
    B -- Equinoctial/MEqOE --> D[Select Dominant Direction Mean Longitude]
    C --> E[Apply Splitting Library Rule 1/Homoscedastic]
    D --> E
    E --> F[Generate L Initial Gaussian Components]
    F --> G[Propagate Each Component independently]
    G -- Using UT/STT/DA/PCE/MF --> H[Sum Propagated Components via Weights]
    H --> I[Final Non-Gaussian Posterior PDF]
```

## Variations and Hybrid Architectures

The GMM framework is frequently combined with other UP methods to optimize the speed-accuracy Pareto front.

### GMM-UT (Unscented Transform)

The most common implementation, where each kernel's statistics are propagated via  $2n + 1$  sigma points.

- **Methodology:** Each component is represented by  $2n + 1$  deterministic sigma points  $\chi_j$ , which are propagated through the full nonlinear dynamics to reconstruct component statistics.
- **Limitation:** The cost scales linearly with the number of components  $L$  and the state dimension  $n$ , i.e.,  $O(L \cdot n)$ .

## GMM-MF (Multi-Fidelity)

A high-efficiency hybrid where the Multi-Fidelity method propagates the sigma points of the GMM kernels.

- **Methodology:** The entire ensemble of sigma points from all components is propagated via a computationally cheap low-fidelity (LF) model (e.g., SGP4). A pivoted Cholesky decomposition of the LF Gramian identifies a subset of “important samples” to be corrected by the HF model via stochastic collocation. See Section for a detailed discussion on the stochastic collocation multi-fidelity method.
- **Performance:** Achieves accuracy similar to GMM-UT but with an **80% reduction in computation time**.

## GMM-STT (State Transition Tensor)

### The Radius of Convergence Problem and GMM mitigation

STTs approximate the solution flow of a dynamical system using a high-order Taylor series expansion about a nominal trajectory. This expansion is localized and only valid within a finite neighborhood known as the **radius of convergence** [Park and Scheeres (2006)] [Acciarini et al. (2025)].

- **Estimating Validity:** To compare how well the Taylor expansions can reconstruct the actual nonlinear dynamics, we can, as an initial analysis of the convergence radius, use results from basic calculus that study the convergence of power series [Acciarini et al. (2025)]. Tools like the **Cauchy–Hadamard theorem** are used as a proxy to estimate a convergence radius and check if the initial uncertainties are within the valid region.
- **Failure Mode:** If the initial state deviations are too large or if the tails of the probability density function (PDF) extend beyond this  $R_c$ , the STT expansion fails to accurately reconstruct the nonlinear dynamics, leading to significant errors.

The **Cauchy–Hadamard**

convergence radius of a Taylor expansion series  $\delta x_f^i \approx \mathcal{P}_i^k(\delta z) =$

$$\sum_{|\alpha|=1}^k \frac{1}{\alpha!} (\partial^\alpha x_f^i) \bigg|_{(\bar{x}_0, \bar{q})} \delta z^\alpha \text{ is given}$$

by:

$$R_c = \frac{1}{\lim_{k \rightarrow \infty} \sup_{|\alpha|=k} |a_\alpha \sqrt{\alpha! / k!}|^{1/k}}$$

Where  $a_\alpha$  is the  $\alpha$ -coefficient of the Taylor series, with  $\alpha$  being the n-tuple already introduced in the expansion.

GMMs address this limitation through a “divide and conquer” approach:

- **Decomposition into Sub-Gaussians:** The initial uncertainty distribution is split into a finite sum of smaller, weighted Gaussian components.
- **Ensuring Local Convergence:** By making each individual Gaussian component “small enough,” the GMM ensures that each component’s domain is fully contained within the STT’s **linear region of convergence** or its high-order radius of convergence.
- **Maintaining Phasing Accuracy:** In long-term propagation, the uncertainty stretches along the trajectory (the direction of maximum nonlinearity). Splitting the distribution—especially in coordinate spaces like **equinoctial elements**—allows these components to stay within accurate bounds even as global non-Gaussianity increases.
- **Mathematical Formula (State Deviation):**

$$\delta \hat{x}_i^k(t) \approx \sum_{p=1}^m \frac{1}{p!} \Phi_{j_1 \dots j_p}^k(t, t_0) \delta \hat{x}_{0,i}^{j_1} \dots \delta \hat{x}_{0,i}^{j_p}$$

- $\delta \hat{x}_i^k(t)$  is the  $k$ -th state component of the  $i$ -th Gaussian component;  $\Phi_{j_1 \dots j_p}^k$  is the  $p$ -th order State Transition Tensor.

### Benefits of Tandem GMM-STT Propagation

- **Reduced Expansion Order:** Because individual GMM components experience less severe “localized” nonlinearity than the original distribution, they can be accurately mapped using **lower-order STTs** (e.g., second-order). This provides high accuracy while avoiding the computational complexity of very high-order tensors.
- **Mitigation of Distribution Tails:** Using GMMs effectively mitigates issues arising from distribution tails that would otherwise extend beyond the  $R_c$ . This ensures the PDF has a compact domain fully contained within the valid expanded flow.

- **Adaptive Efficiency:** Techniques like **Automatic Domain Splitting (ADS)** or **AEGIS** can detect when a component is nearing the edge of its valid convergence region (using entropy or truncation error thresholds) and trigger an online split, keeping the propagation statistically consistent.

### GMM-DA (Differential Algebra)

The core innovation of GMM-DA is that the high-order Taylor expansion of the solution flow is computed only once. The solution flow is expanded into a polynomial up to an arbitrary order  $k$  without manual derivation of variational equations. All individual Gaussian kernels are then evaluated through this single polynomial map, turning the propagation of thousands of components into a series of efficient algebraic evaluations

- **Advantage:** Self-adaptive to any regular nonlinear system without requiring manual derivation of derivatives.
- **Variant (DA-LOADS):** Uses a **Nonlinearity Index (NLI)** to trigger adaptive splitting only when the Taylor expansion's truncation error exceeds a threshold.

### GMM-PCE (Polynomial Chaos Expansion)

This approach is analogous to **hp-refinement** in Finite Element Methods.

- **Methodology:** Splitting (h-refinement) reduces the domain of approximation for each component, allowing for lower-order polynomials (p-refinement) to be used effectively.
- **Benefit:** It reduces the factorial growth of PCE terms in high-dimensional states while maintaining the ability to capture full shapes of non-Gaussian distributions.

## GMM-Moment Method

This semi-analytical framework propagates the first raw moment (mean) and the second central moment (variance) of each Gaussian component through a nonlinear system using the first/or second-order Taylor expansion of the explicit flow function of the dynamical system. Section discusses the mathematical formulation and implementation of the moment method in detail.

## Advanced Variations on the GMM Method

### Adaptive GMMs (AEGIS and MDAES)

The **Adaptive Entropy-based Gaussian-mixture Information Synthesis (AEGIS)** method does not rely on a fixed number of components.

- **Mechanism:** It monitors component nonlinearity by comparing the linear evolution of its differential entropy against a nonlinear estimate (e.g., via UKF).
- **Directional Splitting:** To combat the “curse of dimensionality,” GMM splitting is often restricted to the eigenvector corresponding to the maximum eigenvalue ( $V_{max}$ ) or directions maximizing a nonlinearity metric.
  - **Trigger:** If the entropy deviation exceeds a threshold, the component is split online to maintain linearity.
- **Coordinate Frame Synergy:** Pairing GMMs with **Modified Equinoctial Elements (MEqOE)** is highly efficient, as uncertainty in MEqOE is dominated by a single dimension (true longitude), requiring fewer kernels than Cartesian space.
- **Variation:** **MDAES** combines initial multi-directional splitting with AEGIS to prevent correlation buildup in highly deformed distributions.

Despite its robustness, the sources highlight critical drawbacks:

- **The “Centred Collection” Effect:** In Cartesian frames, individual GMM components can end up clumping toward the center of the distribution at terminal time, failing to capture the thin “tails” of the true non-Gaussian distribution.
- **Combinatoric Growth:** The number of components in a batch GMM can scale factorially with the number of measurements ( $N_{\nu,0} \times \dots \times N_{\nu,L}$ ), necessitating aggressive pruning and merging techniques.
- **Tail Sensitivity:** While GMMs improve upon single Gaussians, they may still require an intractable number of kernels to accurately resolve the  $10^{-5}$  or  $10^{-6}$  tail probabilities required for actionable conjunction screening.

### Multidirectional GMM (MGMM)

Standard GMMs often split along a single axis. **MGMMs** recursively apply splitting libraries along multiple directions to form a regular grid in probability space.

- **Nonlinearity Metric:** Directions are ranked using a second-order divided difference (Stirling’s interpolation) to identify where splits provide the most accuracy gain.

$$\phi = \frac{f(\mathbf{m} + h\sigma\mathbf{a}) + f(\mathbf{m} - h\sigma\mathbf{a}) - 2f(\mathbf{m})}{2h^2}$$

- **Term Definitions:**  $\mathbf{a}$  is the splitting unit vector and  $h = \sqrt{3}$  is the recommended step size.

### Gaussian Mixture Batch Processor (GMBP)

Applied to **Initial Orbit Determination (IOD)**, the GMBP translates a batch of noisy measurements into a GMM-based PDF using Bayes’ rule.

- **Methodology:** It disperses linearization errors across an ensemble of Gaussian components, allowing for accurate reconstruction of “crescent-shaped” posteriors common in cislunar tracking.

### Role in Conjunction Assessment (CA)

In the field of collision analysis, GMMs enable **all-on-all analysis**. Rather than evaluating millions of individual sample pairs as in Monte Carlo simulation, every Gaussian component of a primary object is evaluated against every component of a secondary object. This allows for the use of more advanced analytical or numerical collision probability ( $P_c$ ) computation methods on each individual component pair.

### Adaptive splitting in the Context of Gaussian Mixture Models (GMM)

Bos’s thesis (Bos 2025) defines Adaptive Entropy-based Gaussian-mixture Information Synthesis (AEGIS) as an advanced framework that utilizes automatic domain splitting to maintain an accurate representation of a probability density function (PDF) as its non-Gaussianity increases during propagation.

While standard GMMs often approximate a non-Gaussian PDF by splitting an initial distribution into a fixed number of weighted components at the start of propagation ( $t_0$ ), **AEGIS performs this splitting online**. The method treats the uncertainty as a sum of Gaussian “kernels” that can be individually propagated using efficient techniques like the **Unscented Transform** or **Linearised Covariance**.

The core philosophy of AEGIS is that by repeatedly splitting a Gaussian component into smaller sub-components when non-linear effects become too high, the method reduces the local impact of those nonlinearities, allowing individual components to remain near-Gaussian for a longer duration.

## Mechanics of Adaptive Splitting

AEGIS triggers a split based on a comparison between **linear and nonlinear differential entropy**:

- **Linear Entropy:** This is calculated by integrating the time derivative of the differential entropy, which is a function of the system's **Jacobian matrix** (the basis of the Linearized Covariance method).
- **Nonlinear Entropy:** This is calculated directly from the covariance matrix after propagating the component using a nonlinear technique, typically the **Unscented Transform**.
- **The Splitting Threshold:** When the normalized difference between these two entropy calculations exceeds a user-defined threshold ( $\epsilon_{ent}$ ), the propagation is paused, the component is split into smaller Gaussians, and the entropies are recalculated before propagation continues.

## Performance and Computational Challenges

Despite its theoretical potential, the study highlight significant practical drawbacks to the AEGIS method in orbital scenarios:

- **Extreme Computational Intensity:** AEGIS was found to be the most computationally expensive method tested. Because it requires propagating the State Transition Matrix (STM) and calculating differential entropy at every time step, it is slower than a standard UT even when **no splitting occurs**.
- **Exponential Component Growth:** For long propagation horizons, the number of components can increase exponentially as splits occur at increasingly frequent intervals. This leads to **unreachable computation times** or out-of-memory errors, making the method impractical for scenarios spanning many orbital revolutions.

- **Accuracy Decay:** While AEGIS approximates distributions well for short durations (e.g., 2 revolutions), its accuracy decays sharply in long-term Low Earth Orbit (LEO) cases. In these scenarios, it is generally outperformed by the **Multi-Fidelity (MF)** and **Polynomial Chaos Expansion (PCE)** methods.

### Impact of Coordinate Frames

The study notes that the poor performance of AEGIS was partly due to operating in **Cartesian coordinates**, where nonlinearity affects all dimensions. The sources suggest that combining AEGIS with **Modified Equinoctial Elements (MEqOE)** would be more effective; in MEqOE, the uncertainty primarily distributes along a single direction (true longitude), which would likely require far fewer splits and components to achieve high accuracy.

## Multi-Fidelity (MF) Method of Uncertainty Propagation

In the hierarchy of uncertainty propagation (UP) methodologies, the **Multi-Fidelity (MF) Method** represents a data-adaptive, **efficiency-optimized** paradigm designed to reconcile the computational burden of high-fidelity (HF) orbital dynamics (meaning force-models) with the large-sample requirements of robust statistical characterization. MF methods bridge the gap between “weak” approximation sampling (Monte Carlo) and “strong” spectral expansions (Polynomial Chaos), utilizing a hierarchy of dynamical models to construct a surrogate of the propagated probability density function (PDF) at a fraction of the cost.

### Theoretical Framework and Assumptions

The fundamental premise of MF methods is that a lower-fidelity model can provide a sufficient basis for the space of propagated

samples, which is then corrected by a small subset of higher-fidelity realizations. This approach is particularly effective in astrodynamics because the “main problem” (two-body dynamics plus  $J_2$  perturbations) captures the bulk of orbital motion, while HF perturbations (higher-order gravity, atmospheric drag, solar radiation pressure) act as refinements.

## Assumptions and Definitions

- **Quantity of Interest (QoI):** These are the variables of interest whose uncertainties are needed to be propagated through the dynamics, such as the orbital state vector  $x \in \mathbb{R}^6$ .
- **Model Hierarchy:** In astrodynamics, the LF model typically employs two-body Keplerian dynamics or  $J_2$  perturbations, while the HF model incorporates high-degree gravity fields, atmospheric drag, and solar radiation pressure.
- **Surrogate Basis:** A large ensemble of  $m$  LF samples (say a sample  $\mathbf{x}$  is a vector  $\in \mathbb{R}^n$ ) defines a basis for the space of propagated quantities of interest (QoI). From this, a small subset  $r$  of “important samples” (where  $r < n \ll m$ ) is identified for HF evaluation. ‘
- **Correction Mapping:** The information gained from the  $r$  HF evaluations is used to “lift” the remaining  $m - r$  LF samples toward the HF truth using a stochastic collocation surrogate.

## Mathematical Architecture

### Method 1: Stochastic Collocation (SC) Multi-Fidelity

Developed by Jones and Weisman (2019), this non-intrusive method uses a large ensemble of LF samples to identify a subset of “important” samples for HF propagation.

- **Initial Ensemble:** A set of  $m$  random inputs  $\Xi = \{\xi_i\}_{i=1}^m$  is generated from the initial probability density function (PDF).
- **LF Basis Construction:** The entire ensemble is propagated using an LF model (e.g., SGP4 or two-body dynamics) to define a subspace  $X_L(\Xi) \subseteq \mathcal{X}$ .
- **Important Sample Selection:** Identifying the  $r$  nodes is an optimization problem solved via a **greedy algorithm** that maximizes the distance between the next selected node and the existing basis. This is implemented by solving the pivoted Cholesky decomposition of the Gramian matrix:

$$[X_L]^T G_L [X_L] = A^T C_h C_h^T A \quad (13)$$

**Terms defined:**

- $X_L$ : Matrix of LF propagated samples.
- $G_L$ : Gramian matrix, where  $[G_L]_{i,j} = x_L(\xi_i) \cdot x_L(\xi_j)$  ( $x_L(\xi)$  is the LF propagated state vector for sample  $\xi$ ).
- $A$ : Permutation/pivot matrix that identifies the indices of the important samples.
- $C_h$ : Lower triangular Cholesky factor.

See Section for a detailed derivation of the pivoted Cholesky algorithm and its application to MF uncertainty propagation.

- $c_\ell(\xi)$  **Coefficient Computation:** The expansion coefficients  $c_\ell(\xi)$  in Equation 14 are computed by solving the linear system:

$$\mathbf{C}_h \mathbf{C}_h^T \mathbf{c}() = \mathbf{g}$$

where  $g_i = x_L(\xi) \cdot x_L(\xi_i)$  for  $i = 1, \dots, r$ .

These coefficients effectively map the LF realization of a general sample  $\xi$  to the HF space defined by the  $r$  important samples via Equation 14.

- **HF Correction:** The  $r$  samples are propagated through the HF model, and a surrogate is built using the LF-derived coefficients:

$$\hat{x}_H(\xi) = \sum_{l=1}^r c_l(\xi) x_H(\xi_l^*) \quad (14)$$

**Terms defined:**

- $\hat{x}_H(\xi)$ : The approximated HF state vector for a given input  $\xi$ .
- $x_H(\xi_l)$ : The exact HF state of the  $l$ -th “important sample”.
- $c_l(\xi)$ : Expansion coefficients derived from the LF model to relate the general sample  $\xi$  to the important nodes.
- $r$ : The rank of the surrogate (number of HF evaluations).

Equation 14 is what is referred to as the “low-rank” **stochastic collocation surrogate**. The coefficients  $c_l(\xi)$  are computed by solving a linear system derived from the LF model, ensuring that the surrogate accurately captures the HF dynamics while minimizing the number of expensive HF evaluations.

**Algorithmic Flowchart**

graph TD

```

A[Start: Initial PDF p(x0)] --> B[Generate random inputs ensemble Xi with size m]
B --> C[Propagate ensemble to tf using LF model]
C --> D[Construct matrix of LF samples XL]
D --> E[Compute Pivoted Cholesky Decomposition of XL Gramian]
E --> F[Select r important samples Xi* from pivot matrix A]
F --> G[Propagate only Xi* samples to tf using HF model]
G --> H[Compute surrogate coefficients c_l using LF basis L]
H --> I[Reconstruct/Correct entire HF ensemble using c_l and HF realizations]
I --> J[Result: Refined Propagated PDF p(xf)]
J --> K[Stop]

```

## Method 2: Differential Algebra (DA) and Adaptive GMM Multi-Fidelity

Developed by Fossà et al.(Fossà et al. 2024), this method combines the local accuracy of Taylor expansions with the global flexibility of Gaussian Mixture Models (GMMs).

- **Initialization:** The initial PDF is represented as a GMM where each kernel is initialized in the DA framework as a Taylor polynomial.
- **Adaptive Splitting (LOADS):** The Low-Order Automatic Domain Splitting (LOADS) algorithm monitors a DA-based nonlinearity index ( $NLI$ ) during LF propagation. If the  $NLI$  exceeds a threshold  $\epsilon_\nu$ , the polynomial is split into child kernels to maintain quasi-linearity.
- **LF Propagation:** Statistics are mapped through the LF dynamics (e.g., SGP4) via polynomial evaluation.
- **HF Rectification:** The constant part (center) of each Taylor expansion is propagated point-wise in HF dynamics. The accuracy is restored by re-centering the LF nilpotent part around the HF trajectory:

$$[x_{MF}^{(l)}(t_f)] = \mu_{HF}^{(l)}(t_f) + \{[x_{LF}^{(l)}(t_f)] - \bar{x}_{LF}^{(l)}(t_f)\}$$

where  $[x_{MF}^{(l)}]$  is the MF polynomial,  $\mu_{HF}^{(l)}$  is the HF-propagated mean, and the term in braces is the LF nilpotent (varying) part.

### Algorithmic Flowchart

The following flowchart illustrates the bi-fidelity uncertainty propagation using Differential Algebra and Adaptive GMM as described in the sources:

```
graph TD
    A["A[Start: Initial PDF p(x0)]"] --> B["B[Initialize DA variables or Particles]"]
    B --> C["C[LF Propagation: Map uncertainty through cheap model]"]
    C --> D["D{Nonlinearity Check?}"]
    D -- Yes --> E["E[LOADS/AEGIS: Split domains/kernels]"]
```

```

E --> C
D -- No --> F[Identify Important Samples/Kernel Means]
F --> G[HF Propagation: Point-wise propagation of select points]
G --> H[Polynomial/Surrogate Correction: Re-center or Re-weight]
H --> I[Result: Refined Propagated PDF  $p(\mathbf{x}_f)$ ]
I --> J[Stop]

```

## Hybrid and Efficiency-Optimized Frameworks

The sources highlight several advanced hybridizations and methodological variations that leverage MF's scalability:

- **GMM-MF (Gaussian Mixture Multi-Fidelity):** (from Jones and Weisman (2019)) The GMM-MF framework combines the **Gaussian Mixture Model (GMM)** with **stochastic collocation MF method** to handle non-Gaussian distributions. The GMM decomposes the initial uncertainty into multiple Gaussian kernels. Then sigma points ( $\zeta$ ) are deterministically generated for each kernel, which correspond to random inputs  $\xi$  that are then propagated through multi-fidelity stochastic collocation. This allows for accurate propagation of non-Gaussian distributions while maintaining computational efficiency.
- **Tri-Fidelity Models:** (from Jones and Weisman (2019)) This extension introduces a medium-fidelity (MF) model. LF is used to identify important samples, MF is used to calculate the expansion coefficients  $c$ , and HF provides the final collocation points, further optimizing the accuracy-to-cost ratio.
- **Stochastic Acceleration Propagation (PLASMA):** (from Fossà (2024)) To account for poorly modeled forces (process noise), the PLASMA algorithm propagates polynomial approximations of the moments of the PDF through stochastic differential equations (SDEs). In an MF framework, the initial uncertainty is handled by GMM-DA, while the process noise moments are propagated independently and merged *a posteriori*.
- **Analytical vs. Semianalytical Coupling:**

- *Analytical LF*: Uses closed-form solutions like SGP4 or  $J_2$ -perturbed Lagrangian coefficients for near-instantaneous mapping.
- *Semianalytical LF*: Uses mean element theories (e.g., DSST, STELA) where short-period variations are removed. This allows larger integration steps while maintaining better fidelity than purely analytical models.
- **Credibilistic Filtering**: (from Jones et al. (2019)) Because the surrogate introduces systematic (epistemic) error rather than random error, recent frameworks use **Outer Probability Measures (OPMs)** and possibility functions  $f(x)$  to bound these errors. This provides a “less over-descriptive” representation of uncertainty for establishing custody of newly detected objects.

## Multi-Fidelity in Orbit Propagation

A critical nuance in multi-fidelity orbit propagation is the **state trajectory expansion**.

- **The  $n=6$  Limitation**: Standard MF surrogates are bounded by the state dimension ( $n = 6$ ). To use more nodes ( $r > 6$ ) for higher accuracy, Jones and Weisman (2019) expand the Quantity of Interest (QoI) vector to include the **state trajectory**.
- **Trajectory Augmentation**: By defining the QoI as the state at  $T$  different time instances, the effective dimensionality becomes  $6T$ , allowing for a significantly higher surrogate rank and better representation of the nonlinear manifold.

## Performance and Conjunction Assessment Significance

- **Computational Speedup**: MF achieves a speedup of up to **four orders of magnitude** ( $10^4$ ) compared to brute-force Monte Carlo for high-fidelity LEO propagation.

- **Handling Non-Gaussianity:** Unlike linear methods, MF captures “banana-shaped” warpings and asymmetric tails. In conjunction assessment, as few as **8 important samples** allow MF to produce collision probabilities ( $P_C$ ) that fall within the 95% confidence interval of a full MC simulation.
- **Dimensionality custom:** To avoid limits on  $r$  based on state size ( $n = 6$ ), researchers use the **state trajectory** (state at multiple time points) to increase the effective dimension of the Quantity of Interest (QoI) vector, enabling a more accurate basis.

## Taylor Map Diffusion Method for Closed-Form solution to Fokker-Planck equation

### Fokker-Planck Equation in Orbital Mechanics

The **Fokker-Planck Equation (FPE)** is a partial differential equation that governs the time evolution of the probability density function (PDF) of a dynamical system subject to both deterministic forces and stochastic process noise. In the context of orbital mechanics and conjunction assessment, the FPE is the fundamental tool for transitioning from a point-estimate of a spacecraft’s position to a complete statistical characterization of its possible states.

The FPE describes how a probability cloud “flows” through state space while simultaneously “spreading” due to unmodeled perturbations.

- **Advection-Diffusion Duality:** The equation is composed of two primary operators: the **drift (advection)** term, which moves the PDF along deterministic trajectories (like Keplerian motion), and the **diffusion** term, which accounts for stochastic forcing such as atmospheric drag fluctuations or thrust errors.

- **The Governing Equation:** For a system  $dX = f(X)dt + \sigma dW_t$ , the PDF  $p(x, t)$  evolves as:

$$\frac{\partial p}{\partial t} = -\nabla \cdot (fp) + \frac{1}{2}\text{Tr}(\Sigma_0 \nabla^2 p)$$

Where  $f$  is the deterministic drift and  $\Sigma_0$  is the diffusion tensor.

- **The Curse of Dimensionality:** Traditional numerical solvers for the FPE rely on spatial grids. In a 6D orbital state, even a coarse grid of 50 points per axis requires approximately  $10^{10}$  evaluations per time step, making grid-based FPE solutions computationally intractable for real-time space operations.

By providing a closed-form solution to the FPE, TMD addresses the long-standing “process noise gap” in non-intrusive orbital uncertainty propagation.

## Theoretical Mechanics of the TMD Framework

The TMD method is built upon the **Quadratic Form Conservation Theorem**, which establishes that if a distribution starts with a specific structural form, it will maintain that form throughout its evolution.

- **The Structural Ansatz:** The PDF is represented as  $p(x, t) = N(t)\mu(x, t) \exp(-F(x, t)^T Q(t) F(x, t))$ . This formulation separates the distribution into three primary components:
  - $F(x, t)$ : A nonlinear Taylor map that captures the **shape** of the distribution (curvatures, “banana-shapes,” and asymmetries).
  - $Q(t)$ : A precision matrix (inverse of covariance) that captures the **concentration** and stochastic broadening.
  - $\mu(x, t)$ : A scalar field accounting for the **divergence** of non-conservative forces like drag.

- **Augmented ODE System:** Instead of solving a high-dimensional PDE, TMD reduces the problem to a compact system of coupled Ordinary Differential Equations (ODEs). For a 6D state, this involves only **279 scalar ODEs**, which can be solved in less than a second on standard hardware.
- **Automatic Differentiation (AD):** The implementation leverages AD (e.g., via JAX) to extract the necessary Jacobian and Hessian information from the orbital dynamics, eliminating the need for manual derivation of complex 6D force models.

### Algorithm: The Taylor Diffusion Propagator

```
graph TD
    A["Initial State X0, Covariance P0, Process Noise Σ0"] --> B["Initialize: Trajectory x0, Map Q"]
    B --> C["Compute Precision Matrix Derivative Q"]
    C --> D["Extract Map Evolution Derivatives via Automatic Differentiation"]
    D --> E["Identify Taylor Coefficients: V0, J, H"]
    E --> F["Apply Zero-Tracking Condition: x0 = -J⁻¹V0"]
    F --> G["Integrate Augmented ODE System via RK4"]
    G --> H["Final Time Reached?"]
    H -- No --> C
    H -- Yes --> I["Output Analytic PDF: {x0, J, H, Q}"]
    I --> J["Instantaneous Evaluation of Tail Probabilities"]
```

## Appendix

### A: Pivoted Cholesky Decomposition for Multi-Fidelity Node Selection

In the architecture of **Multi-Fidelity (MF) Uncertainty Propagation**, the **pivoted Cholesky decomposition** serves as the mathematical engine for the “greedy” selection of **important samples** (nodes). This approach identifies a small

subset of points from a large low-fidelity (LF) ensemble that are most representative of the state space, ensuring that high-fidelity (HF) evaluations are spent where they provide the most information.

## The Objective: Information Maximization

The goal of node selection is to find a set of random inputs  $\Xi = \{\xi_1, \dots, \xi_r\}$  such that the span of their low-fidelity realizations,  $\mathcal{X}_L(\Xi)$ , provides the best possible approximation of the full ensemble  $X_L(\Xi)$ .

- **Greedy Selection Logic:** The algorithm iteratively picks the next node  $\xi_k$  by maximizing the distance between that sample and the space spanned by the previously selected nodes.
- **Intractability:** Solving this optimization problem directly via  $\arg \max \text{dist}(\mathbf{x}(\xi), \mathcal{X}(\Xi_{k-1}))$  is computationally expensive (intractable) for high-dimensional state spaces.
- **The Cholesky Proxy:** The pivoted Cholesky decomposition provides an efficient linear algebra substitute to this geometric search by operating on the **Gramian matrix** of the low-fidelity samples.

## Mathematical Architecture

The method leverages the relationship between the inner products of the propagated samples to identify the “rank” or “important” directions of the uncertainty volume.

## The Decomposition Formula

The decomposition is defined as:

$$[X_L]^T G_L [X_L] = A^T C_h C_h^T A$$

**Terms defined:**

- $X_L$ : The matrix of low-fidelity propagated samples ( $n \times m$ ).
- $G_L$ : The Gramian matrix where  $[G_L]_{i,j} = x_L(\xi_i) \cdot x_L(\xi_j)$ . It captures the similarity/redundancy between all sample pairs.
- $A$ : A **pivot matrix** (permutation matrix) that reorders the samples based on their importance.
- $C_h$ : A lower triangular Cholesky factor ( $m \times r$ ).

### Weight Update and Orthogonalization

During the decomposition, the algorithm maintains a vector of “weights”  $w_\ell$  representing the norm of each sample. At each step:

1. The sample with the largest remaining weight (the most unique point) is selected and “pivoted” to the front.
2. The weights of all other samples are updated by subtracting their projection onto the newly selected node:

$$w_t = w_t - C_{h(t,n)}^2$$

This step effectively “deflates” the importance of samples that are redundant with the current basis.

### Algorithm: Node Selection via Pivoted Cholesky

The following flowchart details the procedural logic for identifying  $r$  important samples from  $N$  low-fidelity evaluations:

```
graph TD
    A[Matrix of LF samples: X_L] --> B[Initialize Weights w: Square of norms of columns in X_L]
    B --> C[Set current rank n = 1]
    C --> D[Find index i with max weight w_i]
    D --> E[Is w_i < Tolerance?]
    E -- Yes --> F[Stop: System rank reached]
    E -- No --> G[Swap column n with column i in X_L and weight vector w]
```

```

G --> H[Compute diagonal element: Ch_n,n = sqrt w_n]
H --> I[Compute Ch_t,n for remaining samples t > n]
I --> J[Update remaining weights: w_t = w_t - Ch_t,n^2]
J --> K[n = n + 1]
K --> L{n <= r?}
L -- Yes --> D
L -- No --> M[Output indices of r Important Samples]

```

## B: Derivation of the differential equation for first and second-order State Transition Tensor (STT) propagation

### First-order STT

To derive the differential equation for the first-order STT  $\Phi_{i;a}$ , we need the equations Equation 10, Equation 11 and Equation 12 in first order and follow the steps outlined in Section to match the coefficients of  $\delta x_{0a}$  on both sides of the resulting equation.

#### 1. First order Taylor expansion of the state deviation (Equation 10)

$$\delta x_i(t) = \sum_a \Phi_{i;a} \delta x_{0a} \quad (15)$$

This is Equation 10 in first order.

#### 2. Time derivative of state deviation Taylor map

Since only the STTs depend on time,

$$\delta \dot{x}_i = \sum_a \dot{\Phi}_{i;a} \delta x_{0a} \quad (16)$$

which is Equation 12 in first order.

### 3. Dynamics deviation Taylor map first order (Equation 11)

Equation 11 in first order gives

$$\delta \dot{x}_i = \sum_{\gamma} f_{i;\gamma}^* \delta x_{\gamma} \quad (17)$$

where

$$f_{i;\gamma}^* = \frac{\partial f_i}{\partial x_{\gamma}},$$

evaluated along the reference trajectory.

### 4. Substitution of State deviation Taylor map into dynamics Taylor map

Replace every  $\delta x_{\gamma}$  by

$$\delta x_{\gamma} = \sum_a \dot{\Phi}_{\gamma;a} \delta x_{0a}$$

using Equation 15. Then the Equation 17 becomes

$$\delta \dot{x}_i = \sum_{\gamma} \sum_a f_{i;\gamma}^* \dot{\Phi}_{\gamma;a} \delta x_{0a}$$

### 5. Balancing coefficients

Now we equate both Equation 16 and the modified Equation 17. Since  $\delta x_{0a}$  are independent variables, their coefficients must be equal. Removing the common factor, we obtain the first-order STT differential equation:

$$\boxed{\dot{\Phi}_{i;a} = \sum_{\gamma} f_{i;\gamma}^* \Phi_{\gamma;a}} \quad (18)$$

## Second-order STT

To derive the differential equation for the second-order STT  $\Phi_{i;ab}$ , we need the equations Equation 10, Equation 11 and Equation 12 in second order and follow the steps outlined in Section to match the coefficients of  $\delta x_{0a}\delta x_{0b}$  on both sides of the resulting equation.

### 1. Second order Taylor expansion of the state deviation(Equation 10)

$$\delta x_i(t) = \sum_a \Phi_{i;a} \delta x_{0a} + \frac{1}{2} \sum_{a,b} \Phi_{i;ab} \delta x_{0a} \delta x_{0b} \quad (19)$$

This is Equation 10 truncated after second order.

### 2. Time derivative of state deviation Taylor map

Since only the STTs depend on time,

$$\delta \dot{x}_i = \sum_a \dot{\Phi}_{i;a} \delta x_{0a} + \frac{1}{2} \sum_{a,b} \dot{\Phi}_{i;ab} \delta x_{0a} \delta x_{0b} \quad (20)$$

which is Equation 12 in second order.

### 3. Dynamics deviation Taylor map second order (Equation 11)

Equation 11 truncated to second order gives

$$\delta \dot{x}_i = \sum_{\gamma} f_{i;\gamma}^* \delta x_{\gamma} + \frac{1}{2} \sum_{\gamma,\eta} f_{i;\gamma\eta}^* \delta x_{\gamma} \delta x_{\eta} \quad (21)$$

where

$$f_{i;\gamma}^* = \frac{\partial f_i}{\partial x_{\gamma}},$$

and

$$f_{i;\gamma\eta}^* = \frac{\partial^2 f_i}{\partial x_\gamma \partial x_\eta},$$

evaluated along the reference trajectory.

#### 4. Substitution of state deviation Taylor expansions to dynamics Taylor map

Substituting Equation 19 into the first  $\delta x_\gamma$  of Equation 21 gives

$$\delta \dot{x}_i = \sum_{\gamma} f_{i;\gamma}^* \left( \sum_a \Phi_{\gamma;a} \delta x_{0a} + \frac{1}{2} \sum_{a,b} \Phi_{\gamma;ab} \delta x_{0a} \delta x_{0b} \right) + \frac{1}{2} \sum_{\gamma,\eta} f_{i;\gamma\eta}^* \delta x_\gamma \delta x_\eta$$

##### First contribution

Expanding the first term gives

$$\sum_{\gamma,a} f_{i;\gamma}^* \Phi_{\gamma;a} \delta x_{0a} + \sum_{\gamma,a,b} \frac{1}{2} f_{i;\gamma}^* \Phi_{\gamma;ab} \delta x_{0a} \delta x_{0b}.$$

The first piece is first-order and is part of Equation 18, while the second piece contributes to the second-order equation.

##### Second contribution

Now examine the second term:

$$\sum_{\gamma,\eta} \frac{1}{2} f_{i;\gamma\eta}^* \delta x_\gamma \delta x_\eta.$$

For deriving the second-order equation, only first-order parts of each state are needed because

$$(1st)(1st) = 2nd,$$

while

$$(1\text{st})(2\text{nd}) = 3\text{rd order}$$

and does not become part of the second-order equation. Therefore, we only need to substitute the first-order Taylor expansion of the states into the second contribution, which are

$$\delta x_\gamma = \sum_a \Phi_{\gamma;a} \delta x_{0a},$$

and

$$\delta x_\eta = \sum_b \Phi_{\eta;b} \delta x_{0b}.$$

Hence

$$\delta x_\gamma \delta x_\eta = \sum_{a,b} \Phi_{\gamma;a} \Phi_{\eta;b} \delta x_{0a} \delta x_{0b}$$

Substituting back into the second contribution gives,

$$\sum_{a,b} \frac{1}{2} f_{i;\gamma\eta}^* \Phi_{\gamma;a} \Phi_{\eta;b} \delta x_{0a} \delta x_{0b}.$$

This is the entire second-order contribution from the nonlinear dynamics.

### Collect all second-order terms

The complete right-hand side therefore contains

$$\sum_{a,b} \frac{1}{2} f_{i;\gamma}^* \Phi_{\gamma;ab} \delta x_{0a} \delta x_{0b} + \sum_{a,b} \frac{1}{2} f_{i;\gamma\eta}^* \Phi_{\gamma;a} \Phi_{\eta;b} \delta x_{0a} \delta x_{0b}.$$

The right-hand side from Equation 20 contains

$$\sum_{a,b} \frac{1}{2} \Phi_{i;ab} \delta x_{0a} \delta x_{0b}.$$

## 5. Balancing coefficients

We balance the coefficients of the second order terms  $\delta x_{0a}\delta x_{0b}$  on both sides of the equation, which gives

$$\frac{1}{2}\dot{\Phi}_{i;ab} = \sum_{\gamma,\eta} \left[ \frac{1}{2}f_{i;\gamma}^* \Phi_{\gamma;ab} + \frac{1}{2}f_{i;\gamma\eta}^* \Phi_{\gamma;a} \Phi_{\eta;b} \right]$$

Removing the common factor (1/2), we get the final equation for the second-order STT differential equation:

$$\boxed{\dot{\Phi}_{i;ab} = \sum_{\gamma,\eta} [f_{i;\gamma}^* \Phi_{\gamma;ab} + f_{i;\gamma\eta}^* \Phi_{\gamma;a} \Phi_{\eta;b}]}$$
 (22)

First order STT's  $\Phi_{i;a}$  and  $\Phi_{i;b}$  are obtained by solving the first-order STT differential equation, which is a prerequisite for solving the second-order equation.

Acciarini, Giacomo, Nicola Baresi, David J. B. Lloyd, and Dario Izzo. 2025. “Nonlinear Propagation of Non-Gaussian Uncertainties.” *Journal of Guidance, Control, and Dynamics* 48 (4): 903–13. <https://doi.org/10.2514/1.G008717>.

Adurthi, Nagavenkat, Puneet Singla, and Tarunraj Singh. 2012. “The Conjugate Unscented Transform — an Approach to Evaluate Multi-Dimensional Expectation Integrals.” *2012 American Control Conference (ACC)*, 5556–61. <https://doi.org/10.1109/ACC.2012.6314970>.

Bignon, Emmanuel, P Mercier, Vincent Azzopardi, and R Pinède. 2015. “Accurate Numerical Orbit Propagation Using Polynomial Algebra Computational Engine Pace.” *ISSFD 2015 Congress*.

Boone, Spencer, and Jay McMahon. 2023. “Directional State Transition Tensors for Capturing Dominant Nonlinear Effects in Orbital Dynamics.” *Journal of Guidance, Control, and Dynamics* 46 (3): 431–42. <https://doi.org/10.2514/1.G006910>.

- Bos, Justin. 2025. *Orbit Uncertainty Propagation in Space Situational Awareness Applied to Conjunction Assessment*.
- Detomaso, Roberto, Andrea Muciaccia, and Camilla Colombo. 2025. “Hybrid Gaussian Mixture Model and Unscented Transformation Algorithm for Uncertainty Propagation Within the PUZZLE Software.” *Advances in Space Research* 75 (10): 7226–41. <https://doi.org/https://doi.org/10.1016/j.asr.2025.03.026>.
- Fossa, Alberto, Daniel Alazard, Emmanuel Delande, and Roberto Armellin. 2024. “Multifidelity Orbit Uncertainty Propagation in the Presence of Stochastic Accelerations.” PhD thesis, Ph. D. Thesis, Inst. Supérieur de l’Aéronautique et de l’Espace, Toulouse .... [https://depozit.isae.fr/theses/2024/2024\\_Fossa\\_Alberto.pdf](https://depozit.isae.fr/theses/2024/2024_Fossa_Alberto.pdf).
- Fossà, Alberto. 2024. “Propagation Multi-Fidélité d’incertitude Orbitale En Présence d’accélération Stochastiques.” PhD thesis, Inst. Supérieur de l’Aéronautique et de l’Espace, Toulouse.
- Fujimoto, K., and D. J. Scheeres. 2015. “Tractable Expressions for Nonlinearly Propagated Uncertainties.” *Journal of Guidance, Control, and Dynamics* 38 (6): 1146–51. <https://doi.org/10.2514/1.G000795>.
- Izzo, Dario, Sebastien Origer, Giacomo Acciarini, and Francesco Biscani. 2025. “High-Order Expansion of Neural Ordinary Differential Equation Flows.” *Science Advances* 11 (51): eady1348. <https://doi.org/10.1126/sciadv.ady1348>.
- Jones, Brandon A., and Ryan Weisman. 2019. “Multi-Fidelity Orbit Uncertainty Propagation.” *Acta Astronautica* 155: 406–17. <https://doi.org/https://doi.org/10.1016/j.actaastro.2018.10.023>.
- Jones, Brandon, Emmanuel Delande, Enrico Zucchelli, and Moriba Jah. 2019. “Multi-Fidelity Orbit Uncertainty Propagation with Systematic Errors.” In *Advanced Maui Optical and Space Surveillance Technologies Conference*, edited by

S. Ryan.

Julier, Simon J., and Jeffrey K. Uhlmann. 1997a. “New Extension of the Kalman Filter to Nonlinear Systems.” *Defense, Security, and Sensing*. <https://api.semanticscholar.org/CorpusID:7937456>.

Julier, Simon J., and Jeffrey K. Uhlmann. 1997b. “New extension of the Kalman filter to nonlinear systems.” In *Signal Processing, Sensor Fusion, and Target Recognition VI*, edited by Ivan Kadar, vol. 3068. International Society for Optics; Photonics; SPIE. <https://doi.org/10.1117/12.280797>.

Park, Ryan S., and Daniel J. Scheeres. 2006. “Nonlinear Mapping of Gaussian Statistics: Theory and Applications to Spacecraft Trajectory Design.” *Journal of Guidance, Control, and Dynamics* 29 (6): 1367–75. <https://doi.org/10.2514/1.20177>.

Soman, Suyog. 2025. *Orbital Uncertainty Propagation for Collision Probability Computation*.

Sun, Pan, Kai Cao, and Shuang Li. 2026. “Recent Advances of Practical Techniques for Nonlinear Uncertainty Propagation in Orbital Dynamics.” *Science China Physics, Mechanics & Astronomy* 69 (1): 214501.

Tenne, D., and T. Singh. 2003. “The Higher Order Unscented Filter.” *Proceedings of the 2003 American Control Conference, 2003*. 3: 2441–2446 vol.3. <https://doi.org/10.1109/ACC.2003.1243441>.

Van der Merwe, R., and E. A. Wan. 2001. “The Square-Root Unscented Kalman Filter for State and Parameter-Estimation.” *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)* 6: 3461–3464 vol.6. <https://doi.org/10.1109/ICASSP.2001.940586>.

Zhou, Xingyu, Roberto Armellin, Dong Qiao, and Xiangyu Li. 2024. *Time-Varying Directional State Transition Tensor for*

*Orbit Uncertainty Propagation.* <https://arxiv.org/abs/2412.07060>.