DIFFERENTIAL ALGEBRA SPACE TOOLBOX* FOR NONLINEAR UNCERTAINTY PROPAGATION IN SPACE DYNAMICS

6th International Conference on Astrodynamics Tools and Techniques (ICATT)
Darmstadt, Germany

M. Rasotto, A. Morselli, A. Wittig, M. Massari,
P. Di Lizia, R. Armellin, C. Y. Valles and
G. Ortega

March 17th, 2016

** Developed under ITT AO/1-7570/13/NL/MH: Nonlinear Propagation of Uncertainties in Space Dynamics based on Taylor Differential Algebra"*

MOTIVATION

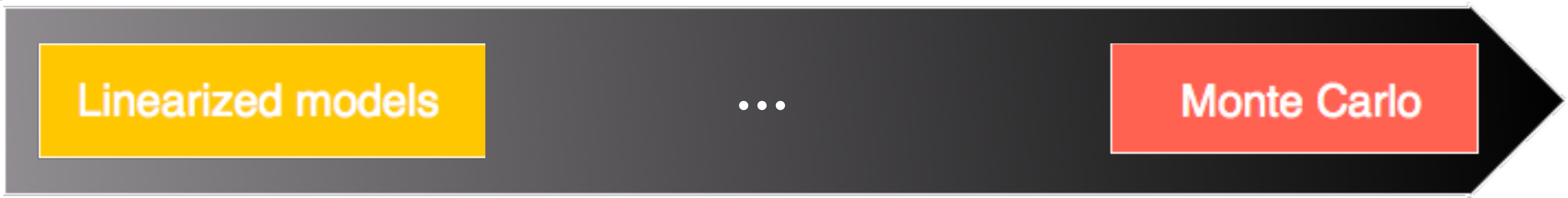
- Uncertainty propagation is a crucial issue in spaceflight dynamics
  - Space surveillance and tracking
  - Reentry and casualty area computation
  - Robust design of space trajectories and systems
  - …

- Most spaceflight mechanics problems involve nonlinear behavior

↓

Need of efficient tools for nonlinear propagation of uncertainties

PROPOSED APPROACH



- low computational burden
- low accuracy

- computationally intensive
- high accuracy

Can we find a compromise technique?

- We need a technique to:
  - Improve accuracy of linearized models
  - Reduce computational cost of classical Monte Carlo

Proposed Approach

Linearized models … Differential Algebra … Monte Carlo

- low computational burden
- low accuracy

- computationally intensive
- high accuracy

Can we find a compromise technique?

- We need a technique to:
  - Improve accuracy of linearized models
  - Reduce computational cost of classical Monte Carlo

Differential Algebra

AST

DIFFERENTIAL ALGEBRA

- Differential Algebra (DA) is an automatic differentiation technique

Algebra of real numbers → Algebra of Taylor polynomials

- DA can be implemented in a computer environment (DACE)
- Given any sufficiently regular function $f(x)$
  - Initialize x as a DA variable: $[x] = \overline{x} + \delta x$
  - Evaluate f in the DA framework:

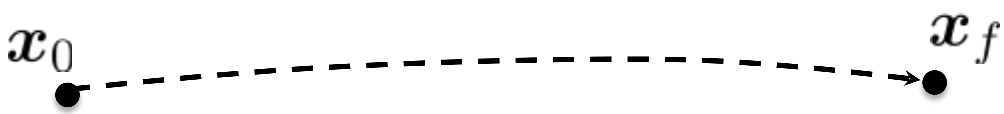
$$f([x]) = \mathcal{T}_f(\delta x)$$

→ Taylor expansion of f around $\overline{x}$ up to an **arbitrary order** k

Expansion of the Flow of ODEs

- Given any dynamics

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



- Any numerical integrator is based on the evaluation of f and its algebraic manipulation
  - Initialize x_0 as DA vector
  - Perform operations in DA

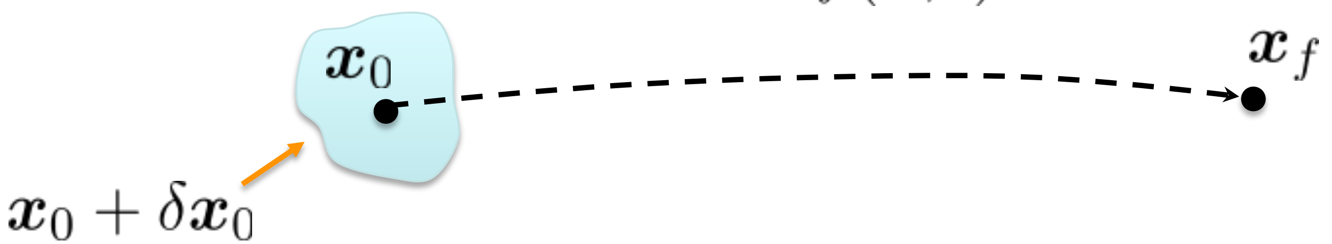
⇨ Taylor expansion of x_f w.r.t. x_0

- Uncertainty propagation can benefit from DA in different ways

EXPANSION OF THE FLOW OF ODES

- Given any dynamics

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



- Any numerical integrator is based on the evaluation of f and its algebraic manipulation
  - Initialize $\boldsymbol{x}_0$ as DA vector
  - Perform operations in DA

⇒ Taylor expansion of $\boldsymbol{x}_f$ w.r.t. $\boldsymbol{x}_0$

- Uncertainty propagation can benefit from DA in different ways

Expansion of the Flow of ODEs

- Given any dynamics

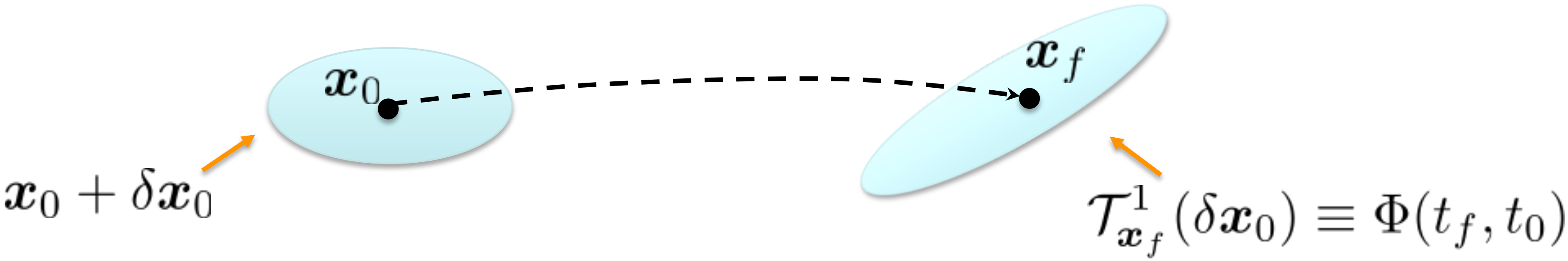
$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, t)$$

- Any numerical integrator is based on the evaluation of f and its algebraic manipulation
  - Initialize $\boldsymbol{x}_0$ as DA vector
  - Perform operations in DA

⇒ Taylor expansion of $\boldsymbol{x}_f$ w.r.t. $\boldsymbol{x}_0$

- Uncertainty propagation can benefit from DA in different ways

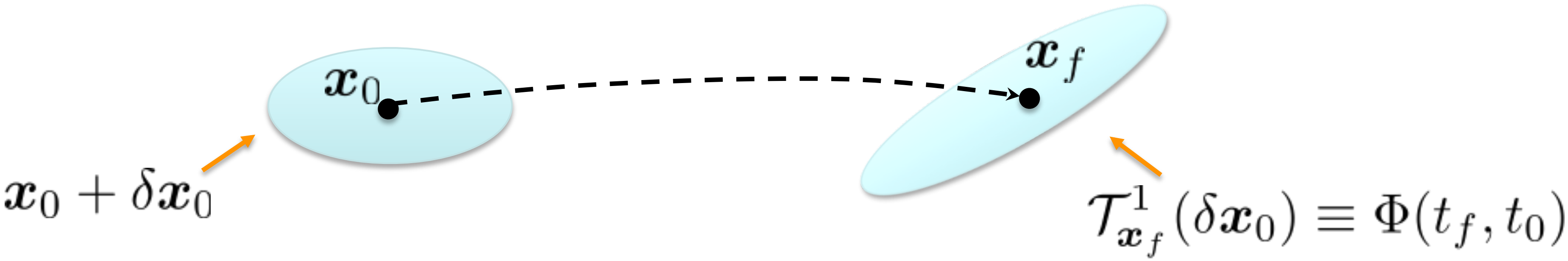
Linear Covariance Propagation



- If all computations are performed to order 1, the final Taylor expansion $\mathcal{T}^1_{x_f}(\delta x_0)$ coincides with the state-transition matrix $\Phi(t_f, t_0)$

Perform a single DA integration with expansion order 1 → Extract the STM from the Taylor polynomial map → Map the initial covariance: $C_f = \Phi\, C_0\, \Phi^T$

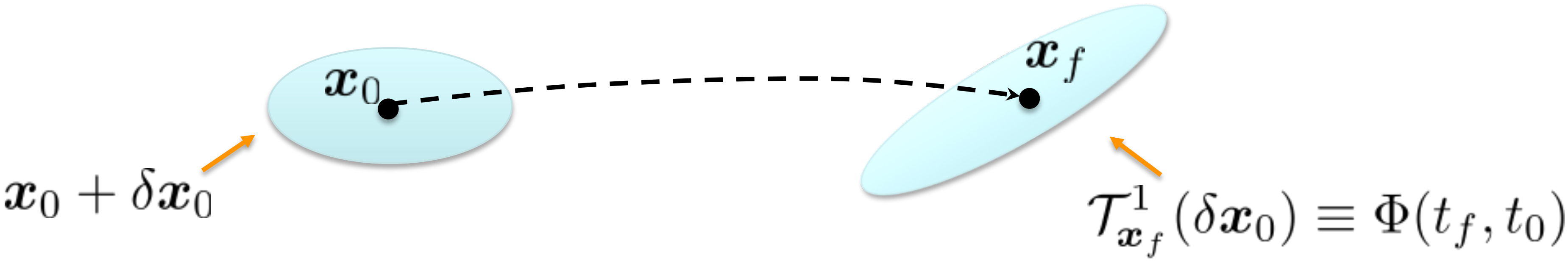
LINEAR COVARIANCE PROPAGATION



- If all computations are performed to order 1, the final Taylor expansion $\mathcal{T}^1_{x_f}(\delta x_0)$ coincides with the state-transition matrix $\Phi(t_f, t_0)$

Perform a single DA integration with expansion order 1 → **Extract the STM from the Taylor polynomial map** → **Map the initial covariance:** $C_f = \Phi \, C_0 \, \Phi^T$

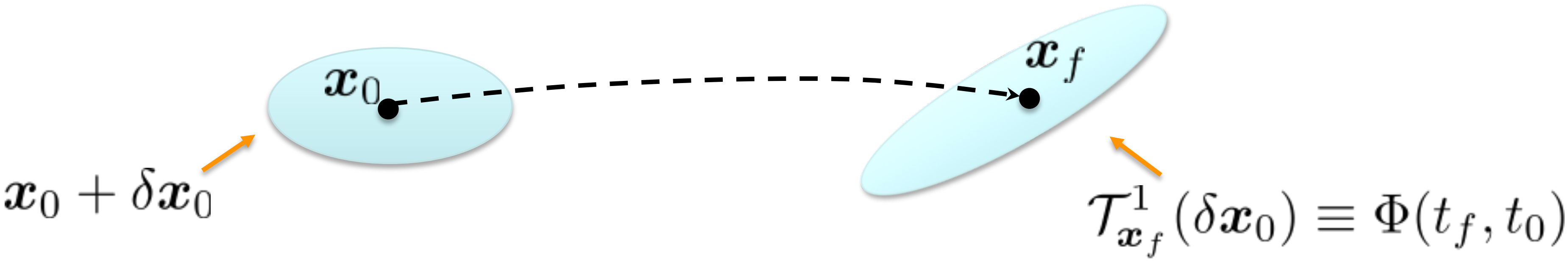
Linear Covariance Propagation



- If all computations are performed to order 1, the final Taylor expansion $\mathcal{T}^1_{x_f}(\delta x_0)$ coincides with the state-transition matrix $\Phi(t_f, t_0)$

Perform a single DA integration with expansion order 1 → **Extract the STM from the Taylor polynomial map** → **Map the initial covariance:** $C_f = \Phi \, C_0 \, \Phi^T$

Linear Covariance Propagation

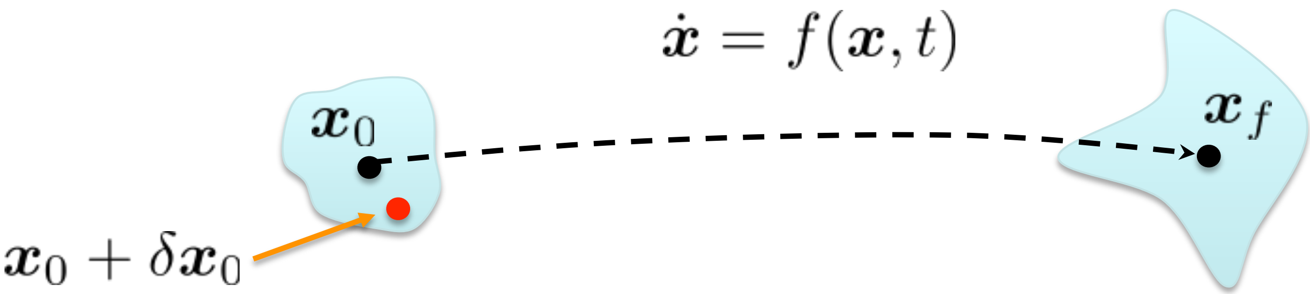


- If all computations are performed to order 1, the final Taylor expansion $\mathcal{T}^1_{\boldsymbol{x}_f}(\delta\boldsymbol{x}_0)$ coincides with the state-transition matrix $\Phi(t_f, t_0)$

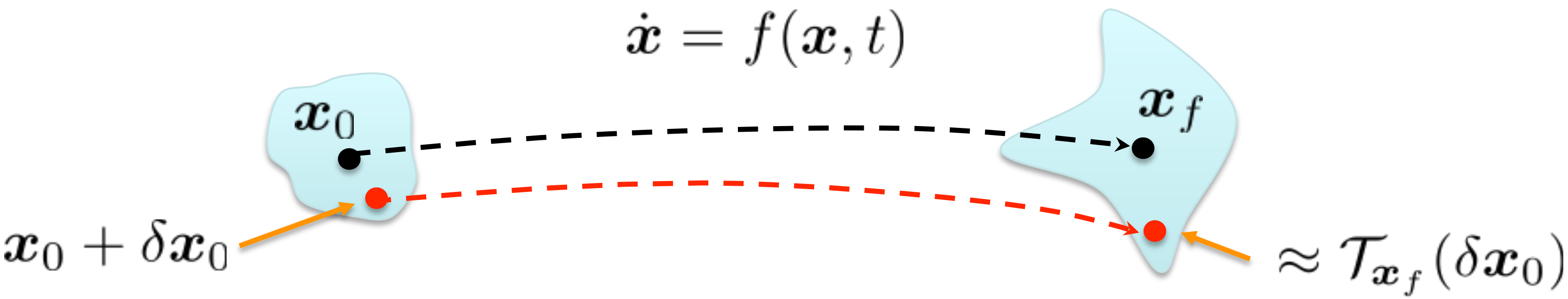
Perform a single DA integration with expansion order 1 → **Extract the STM from the Taylor polynomial map** → **Map the initial covariance:** $C_f = \Phi \, C_0 \, \Phi^T$

No need of variational equations!

DA-BASED MONTE CARLO



DA-BASED MONTE CARLO



- Any pointwise integration can be replaced by the evaluation of the polynomial $\mathcal{T}_{x_f}(\delta x_0)$

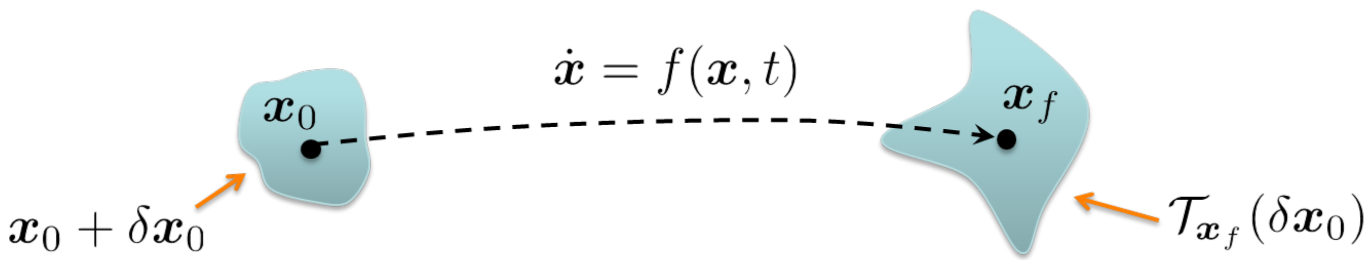
⇩

Saving in computational time w.r.t. classical MC

DA-BASED MONTE CARLO

Compute high order expansion of the flow → Evaluate expansion at samples → Compute moments of PDF

Generate random samples ↓ Evaluate expansion at samples



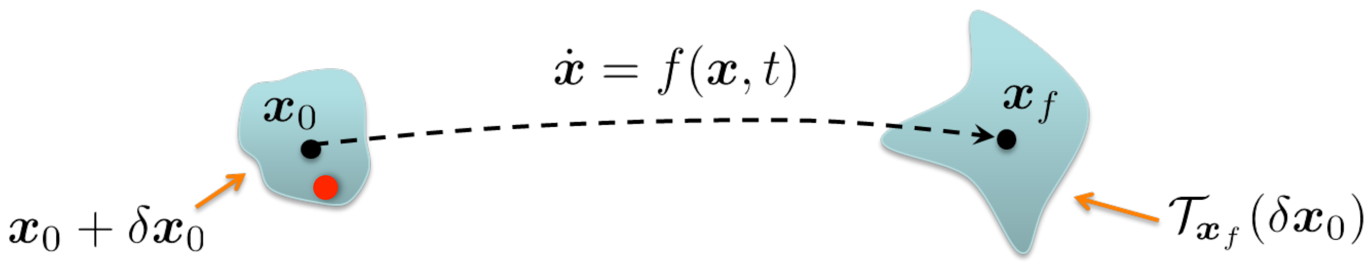
DA-based Monte Carlo

Generate random samples

Compute high order expansion of the flow

Evaluate expansion at samples

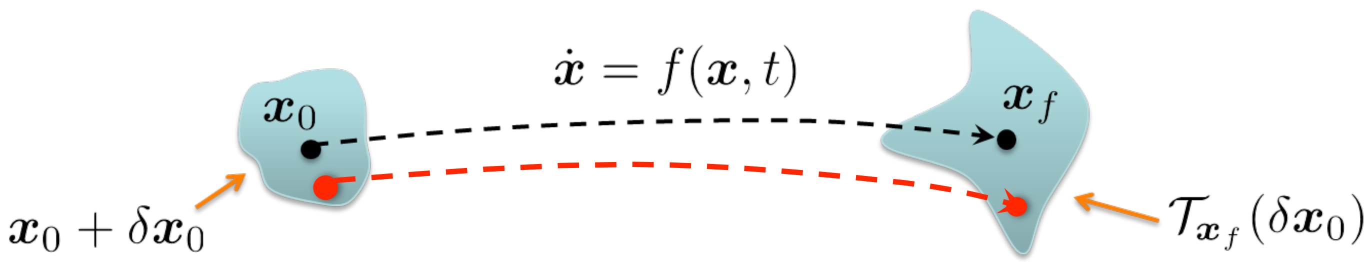
Compute moments of PDF



DA-based Monte Carlo

Generate random samples

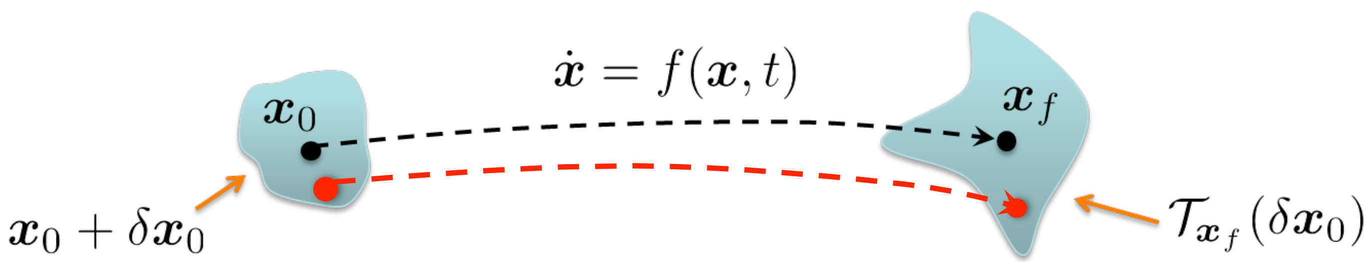
Compute high order expansion of the flow → **Evaluate expansion at samples** → Compute moments of PDF



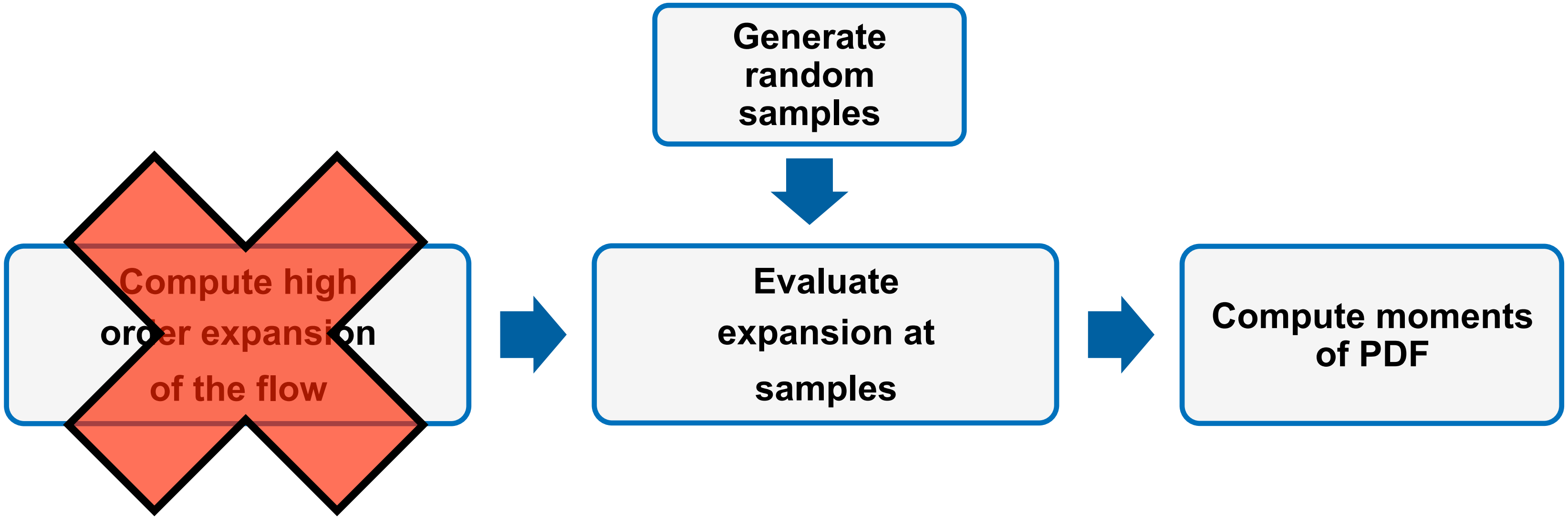
DA-BASED MONTE CARLO

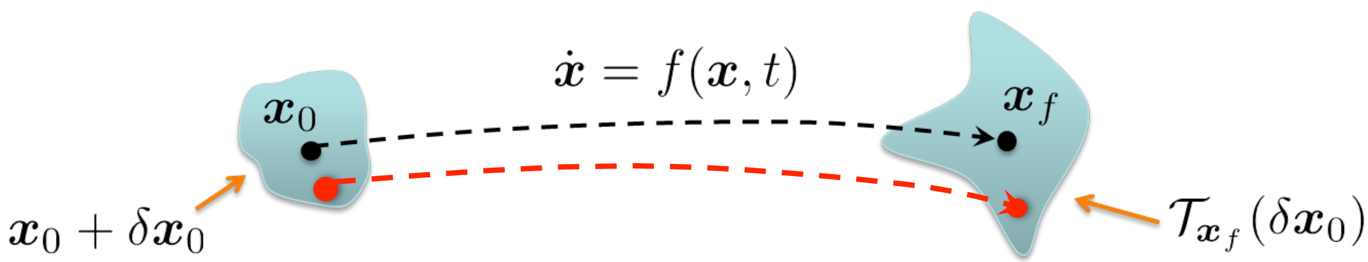
Generate random samples

Compute high order expansion of the flow → **Evaluate expansion at samples** → **Compute moments of PDF**



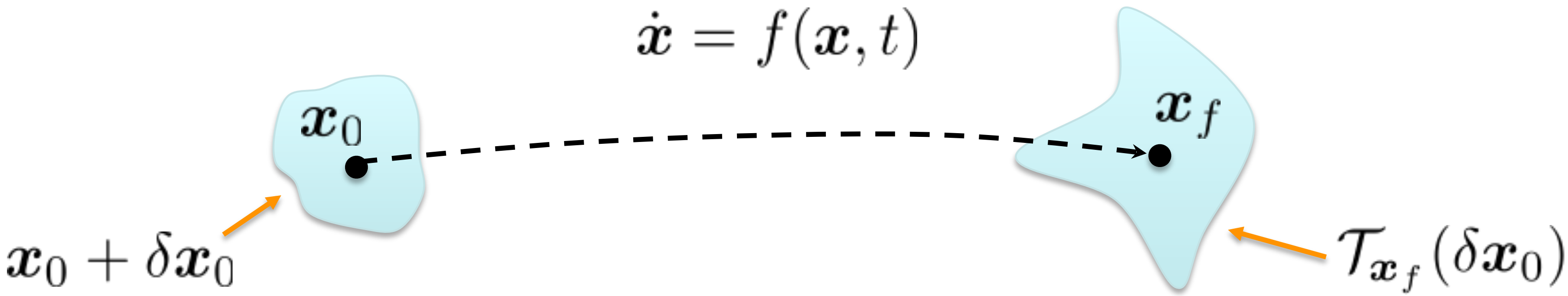
DA-BASED MONTE CARLO



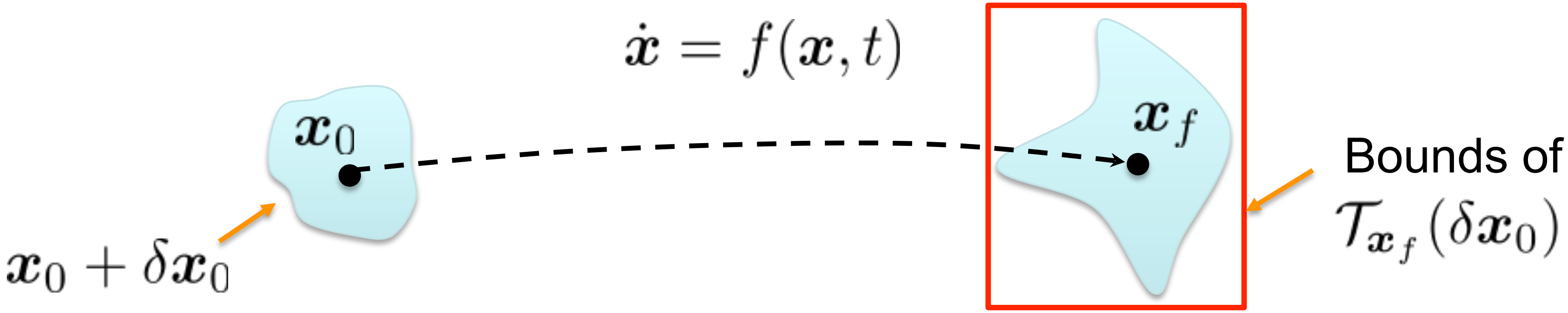


Same polynomial can be used to map different uncertainties

POLYNOMIAL BOUNDER

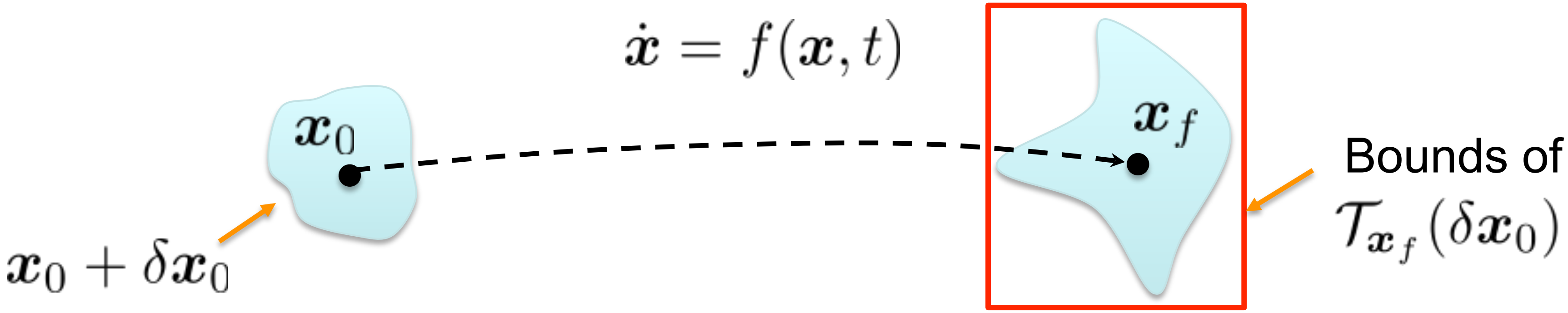


POLYNOMIAL BOUNDER



- Use polynomial bounders to estimate the range of the propagated uncertainties by bounding $\mathcal{T}_{\boldsymbol{x}_f}(\delta\boldsymbol{x}_0)$

POLYNOMIAL BOUNDER



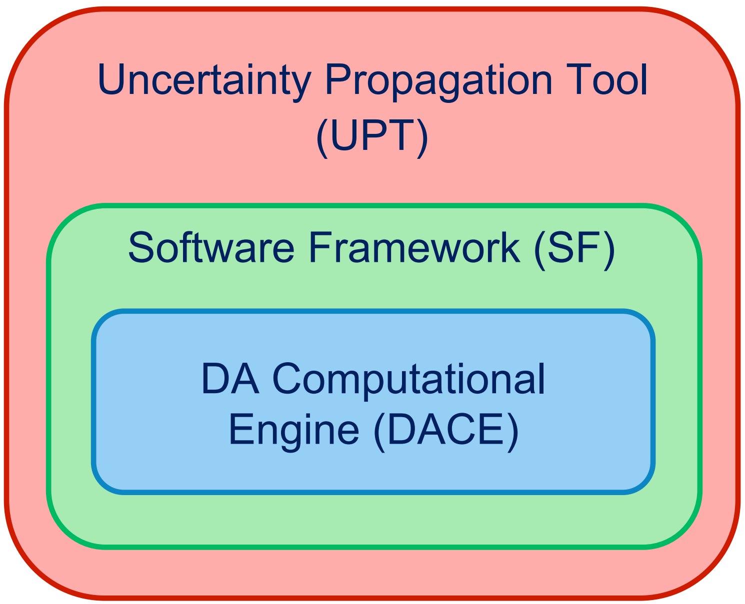
- Use polynomial bounders to estimate the range of the propagated uncertainties by bounding $\mathcal{T}_{x_f}(\delta x_0)$

Rescale the domain of each variable to [-1,1] → **Perform a single DA integration** → **Compute the range of the polynomial**

- Useful in some applications to verify constraints satisfaction (no need to map a statistical distribution)

DIFFERENTIAL ALGEBRA SPACE TOOLBOX

Differential Algebra
Space Toolbox



- **DA Computational Engine**

  Implements Taylor DA arithmetic to handle polynomial operations

- **Software Framework**

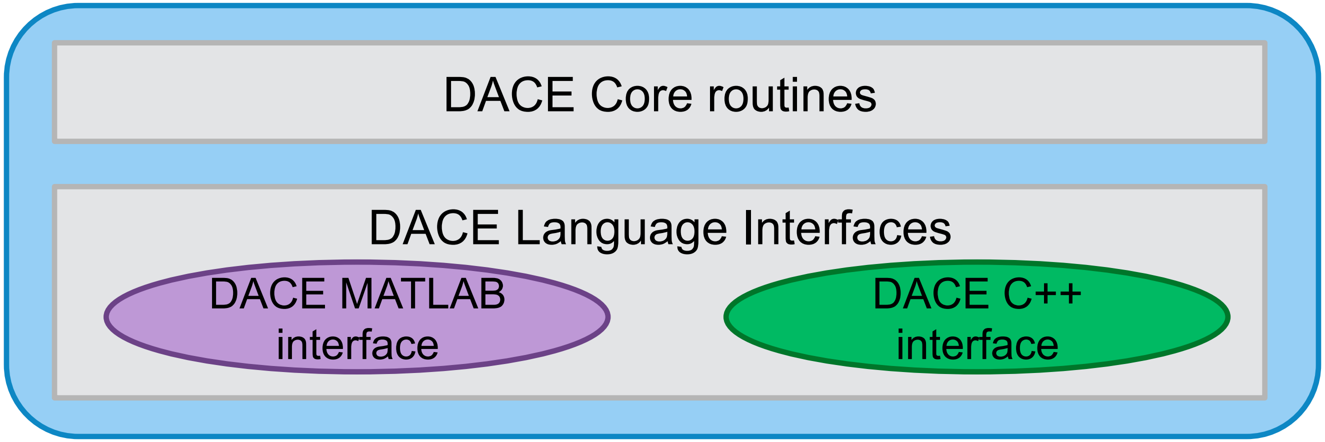
  Provides all routines to perform DA based propagation in astrodynamics

- **Uncertainty Propagation Tool**

  Provides all routines and an interface for DA-based propagation of uncertainties

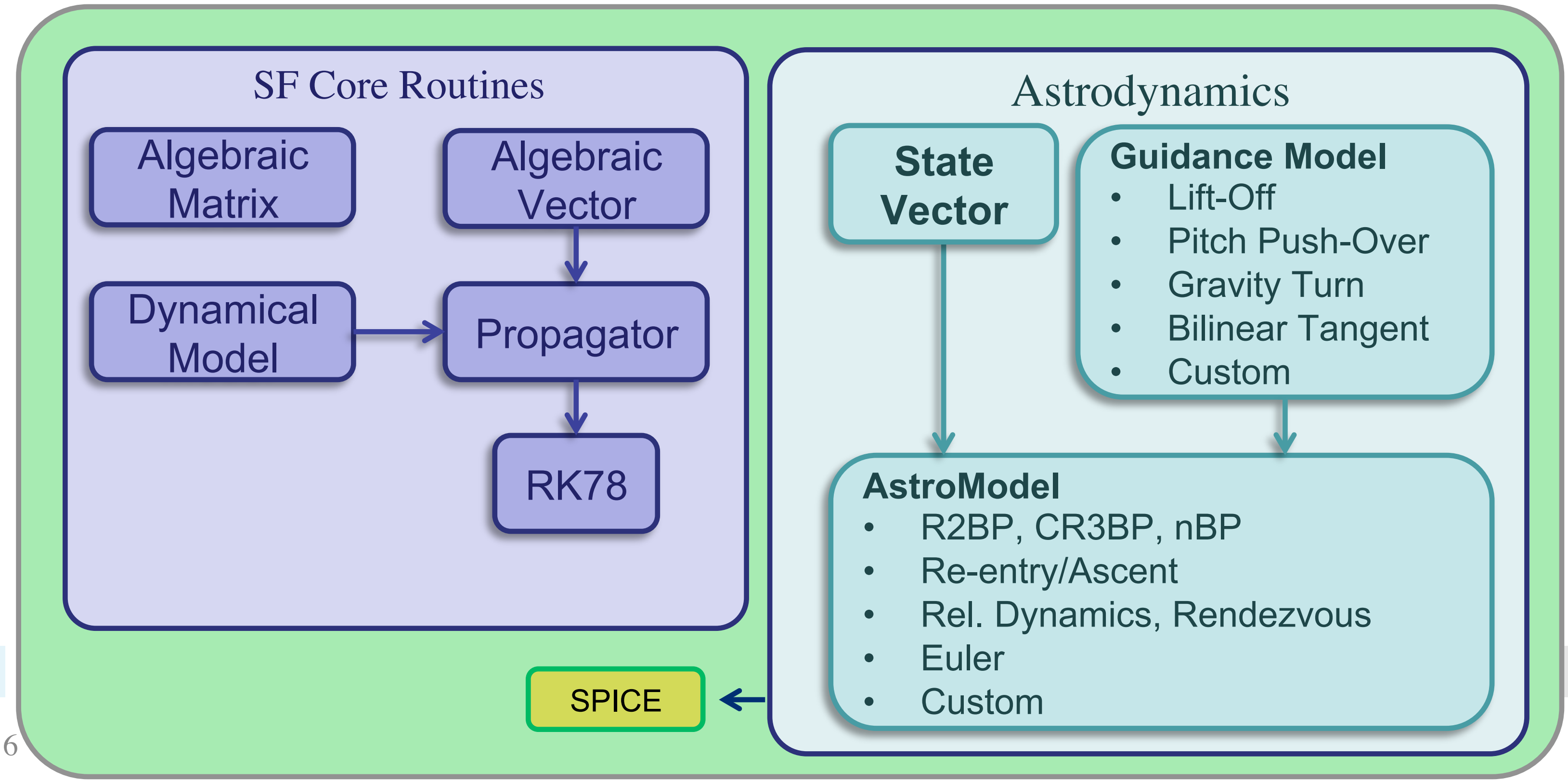
DACE Module Overview

- The DA Computational Engine (DACE) contains the implementation of the basic DA routines
- DACE Core routines: Fortran 95
  - Initialization, Memory management, error handling, DA operations
  - Each routine approximates the result of an operation by its Taylor exp.
- Interfaces: C++, MATLAB



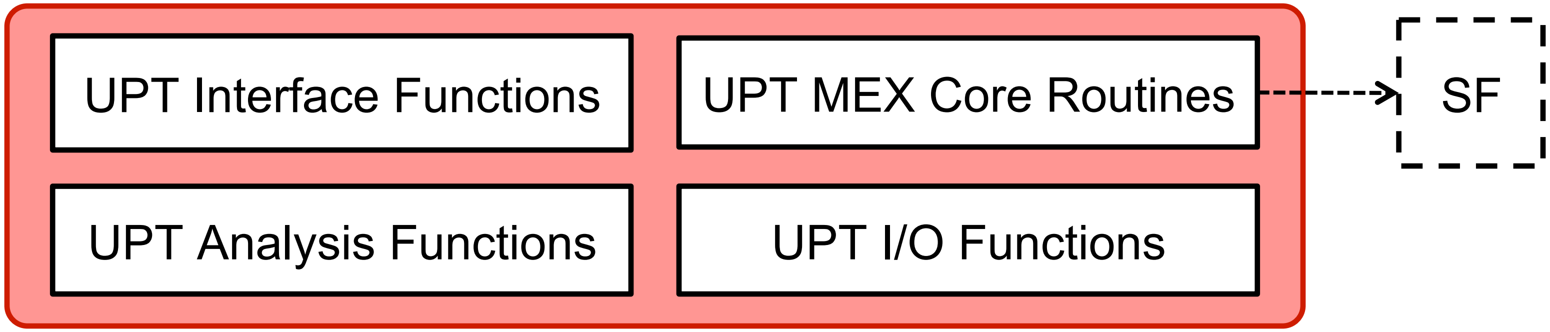
SF Module Overview

- The SF includes more advanced features:
  - DA operations between vectors and/or matrices of DA
  - DA implementation of the numerical integration schemes
  - Dynamical models and guidance models



UPT Module Overview

- The UPT is aimed at performing uncertainty propagations and statistical analyses
- The UPT comes with a set of Matlab routines to:
  - Easily interface with the SF (run DA based computations)
  - Managing simulation results
  - Easy graphical representations of the performed analyses



DIFFERENTIAL ALGEBRA SPACE TOOLBOX

Standard user

Uncertainty Propagation Tool (UPT)

Uncertainty propagation in various dynamical systems in space-related applications

Developer

Software Framework (SF)

Implementation of custom DA-based astrodynamics applications

DACE

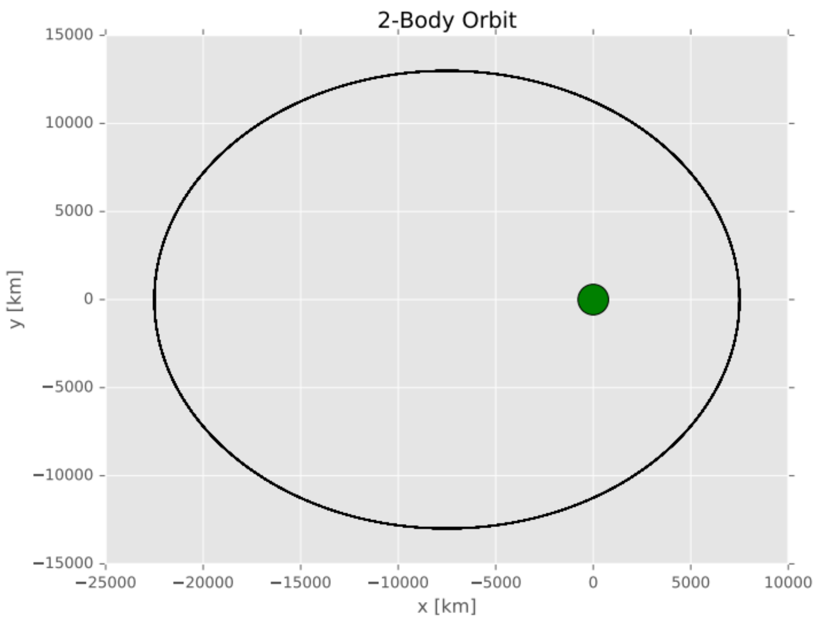
Use of DA operations for any custom application

Applications: Two-Body Dynamics

- Initial conditions

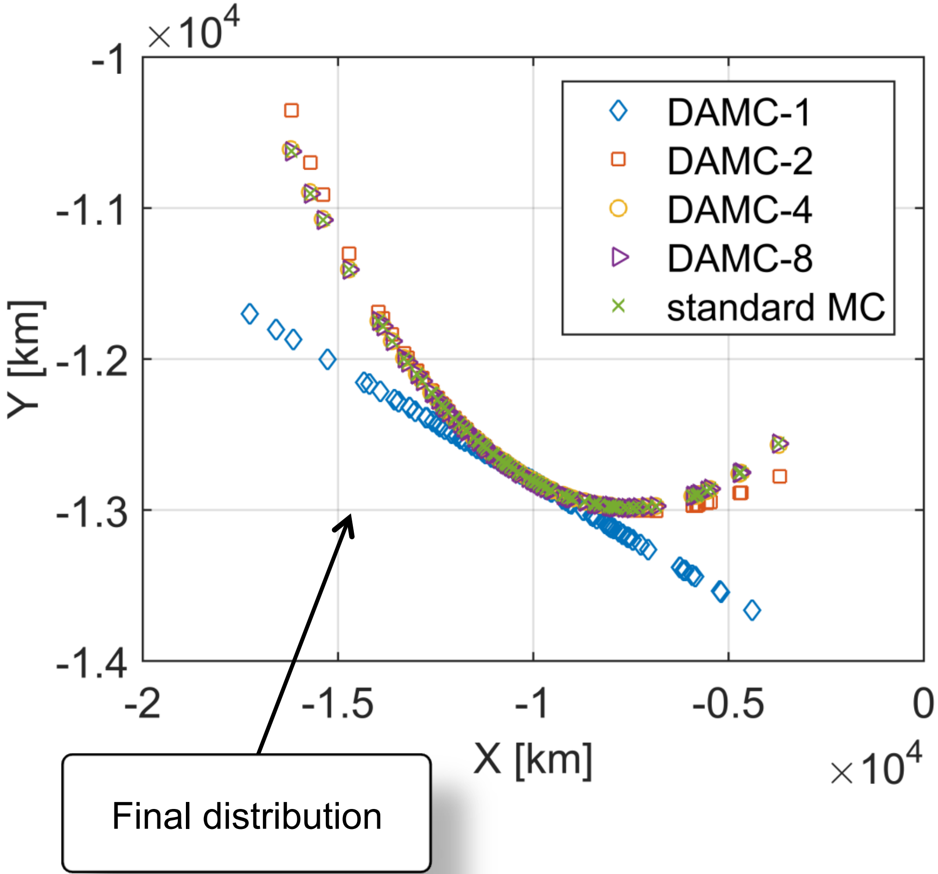
Initial Conditions			
State	Value	σ	Units
r_x	7.5e3	5e-2	km
r_y	0.0	5e-2	km
r_z	0.0	0.0	km
v_x	0.0	0.0	km/s
v_y	8.9286	0.0	km/s
v_z	0.0	0.0	km/s

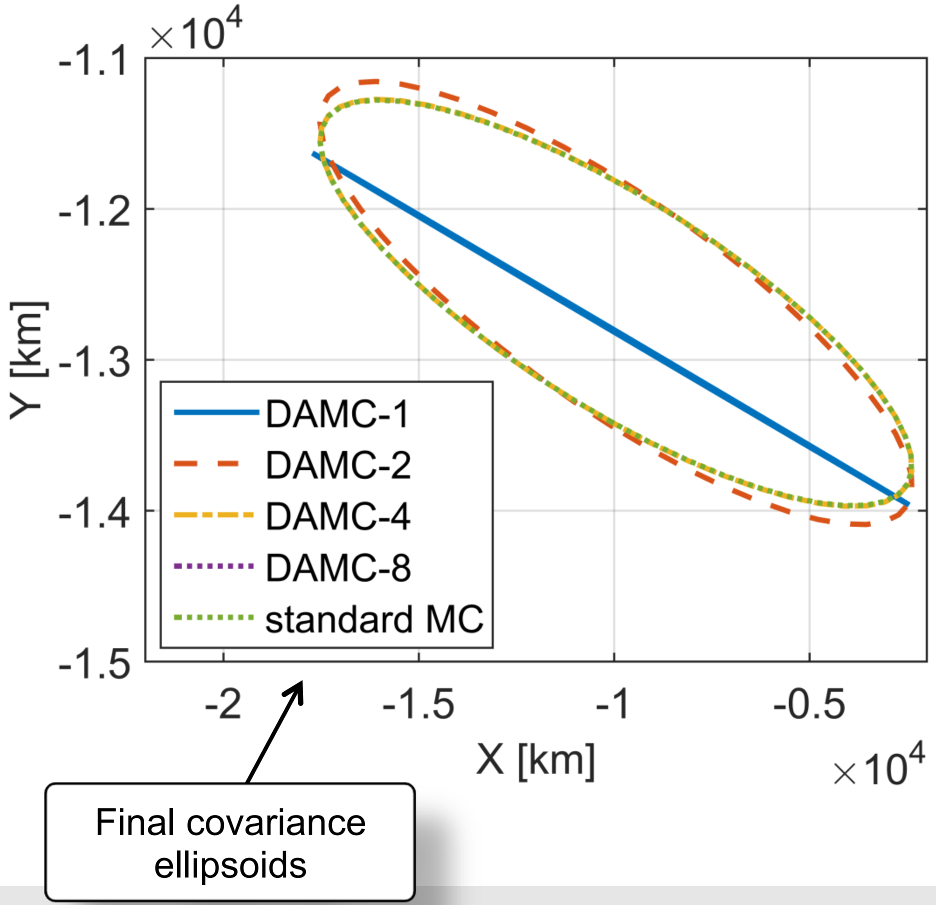
Propagation Time
$T_0 = 0.0$
T_F = 30.8 orbital periods



Applications: Two-Body Dynamics

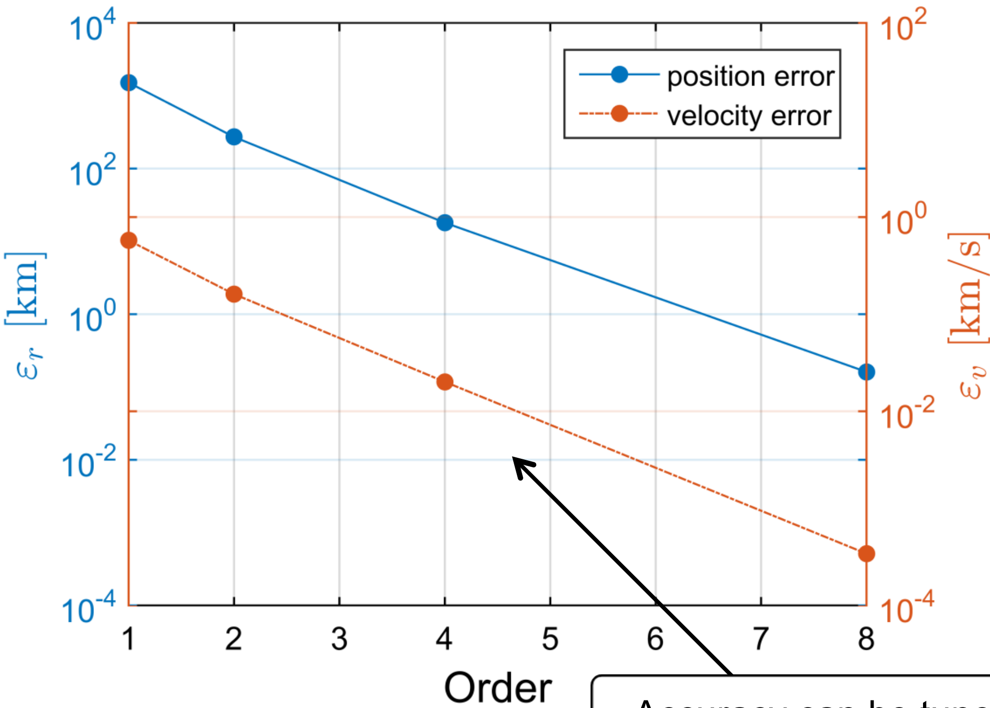
- DA based Monte Carlo

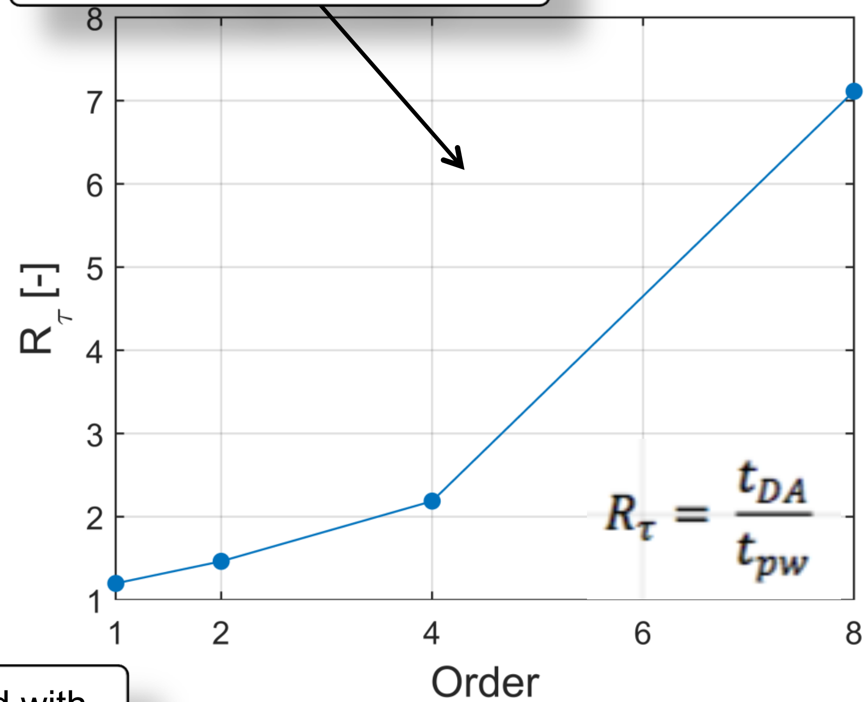




Applications: Two-Body Dynamics

- DA based Monte Carlo



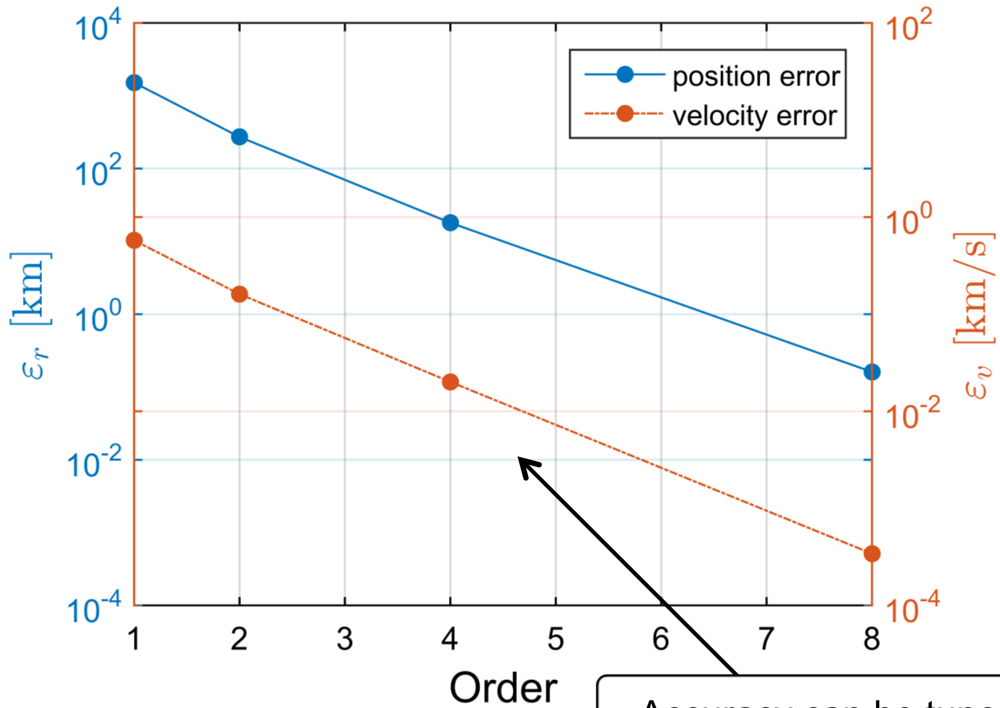


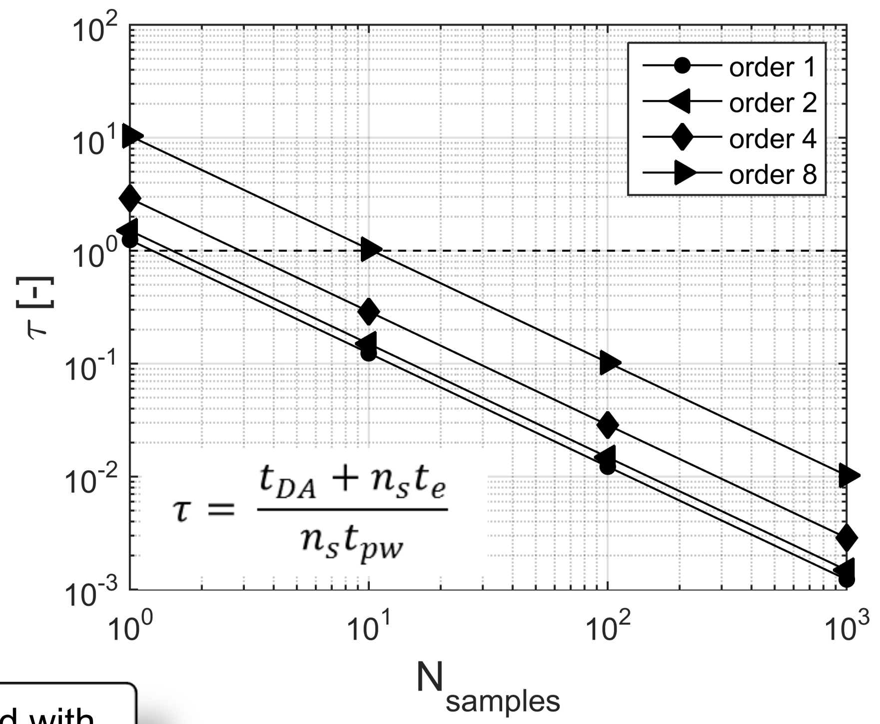
Computational cost increases for higher orders

Accuracy can be tuned with the polynomial order

APPLICATIONS: TWO-BODY DYNAMICS

- DA based Monte Carlo





Accuracy can be tuned with the polynomial order

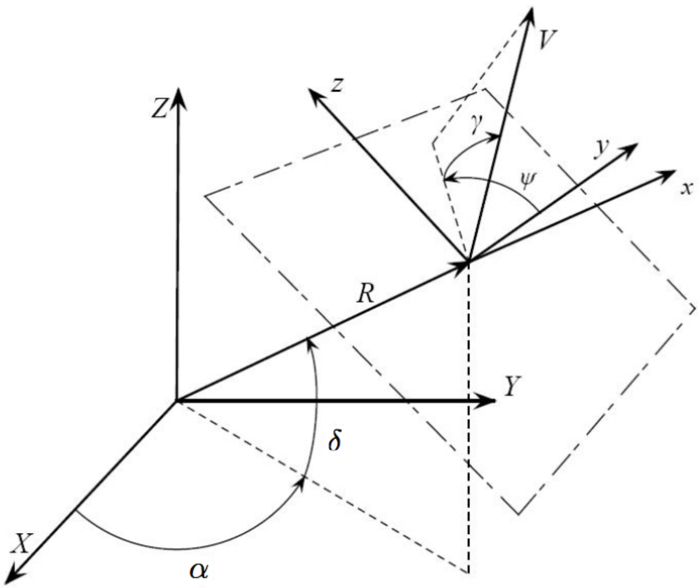
APPLICATIONS: RE-ENTRY DYNAMICS

- Re-entry dispersion analysis for a ballistic re-entry vehicle

Equations of motion

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

Reference frame



Exponential density model is used: $\rho = \rho_0 e^{-\frac{h}{\beta}}$ (NRLMSISE available)

Applications: Re-entry Dynamics

- Hayabusa re-entry*

Initial Conditions			
State	Value	σ	Units
h	201.992	0.1085	km
α	-124.28	0.0074	deg
δ	-27.33	0.008	deg
v	12.035	0.002	km/s
γ	-12.35	0.0044	deg/s
ψ	-22.06	0.0119	deg/s

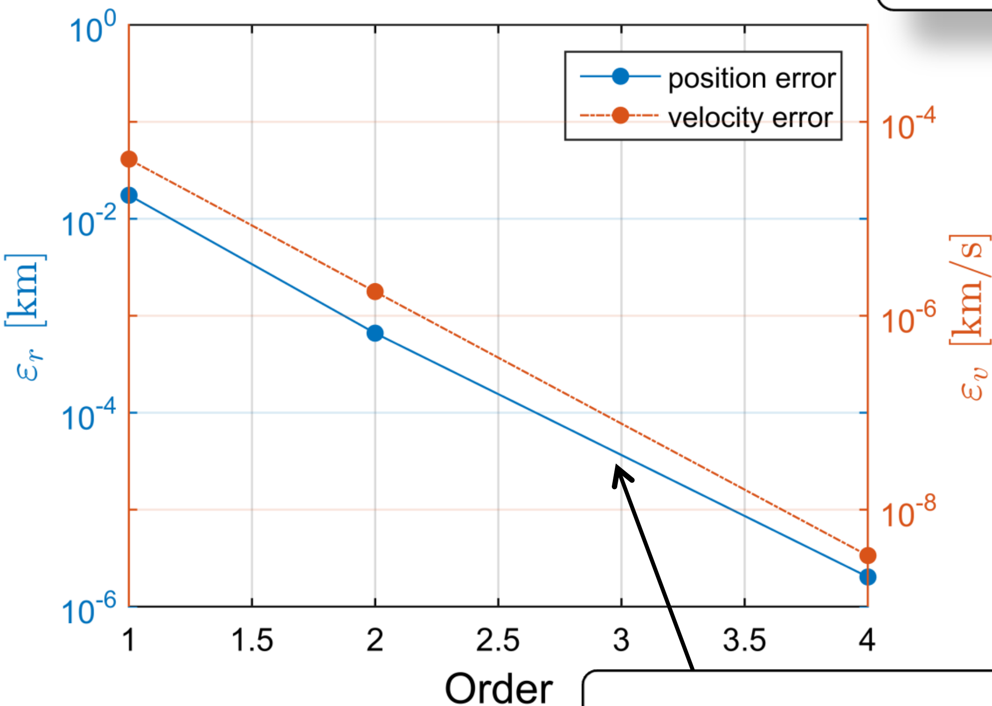
Model Parameters			
Parameter	Value	σ	Units
C_D	1.30	3.3%	-
C_L	0	-	-
m	18	-	kg
S	0.126	-	m^2
ρ_0	1.217	6.6%	kg/m^3
β	8.5	-	km

Propagation Time
T_0 = 2010-06-13, 13:51:11.47 UTC
T_F = 2010-06-13, 15:00:00.00 UTC

*Cassel et al., 2011

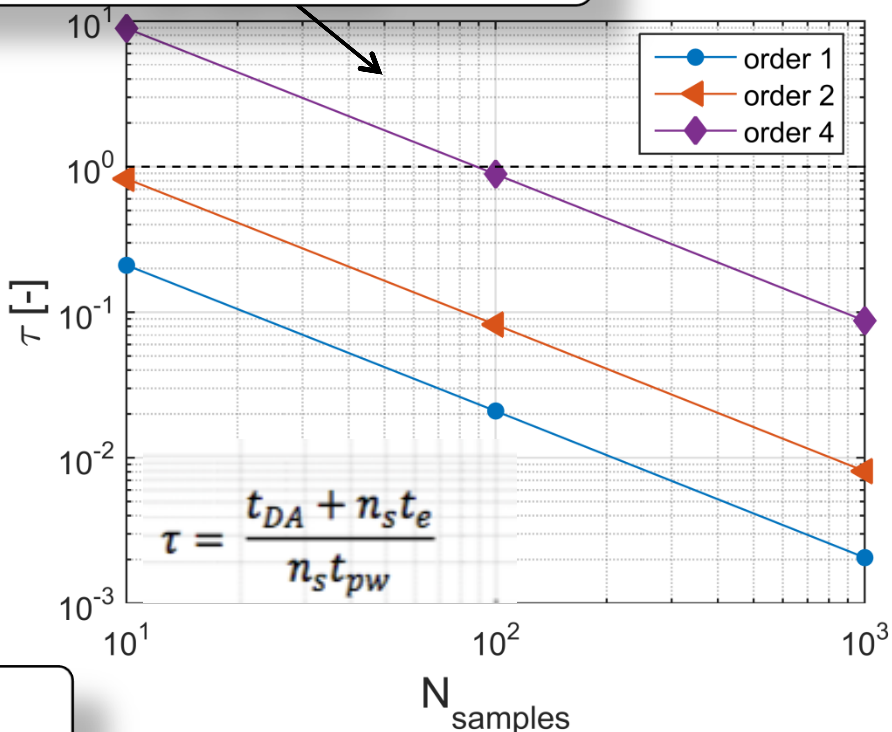
Propagation stopped at 25 km!

- DA based Monte Carlo



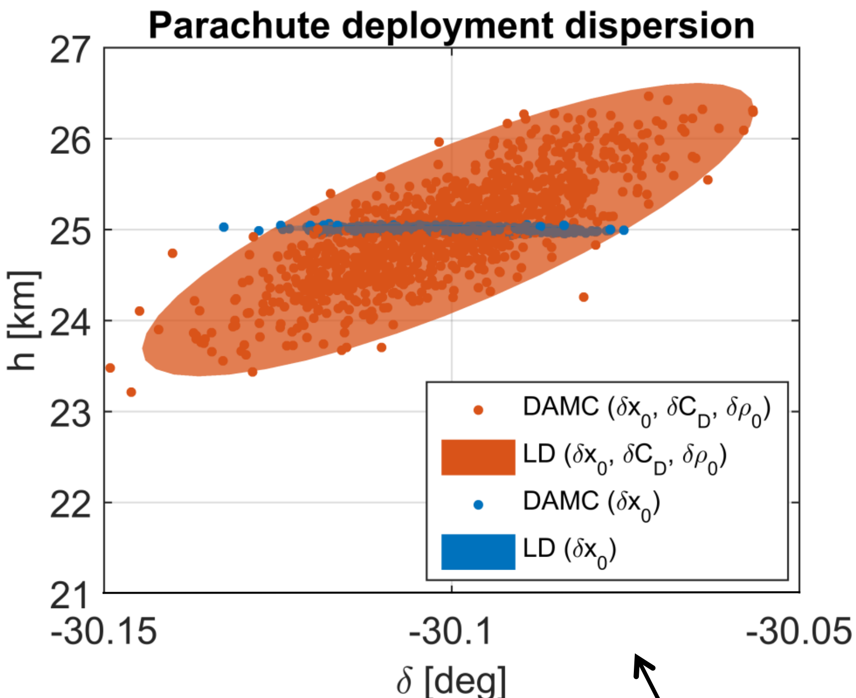
Errors decrease with higher orders

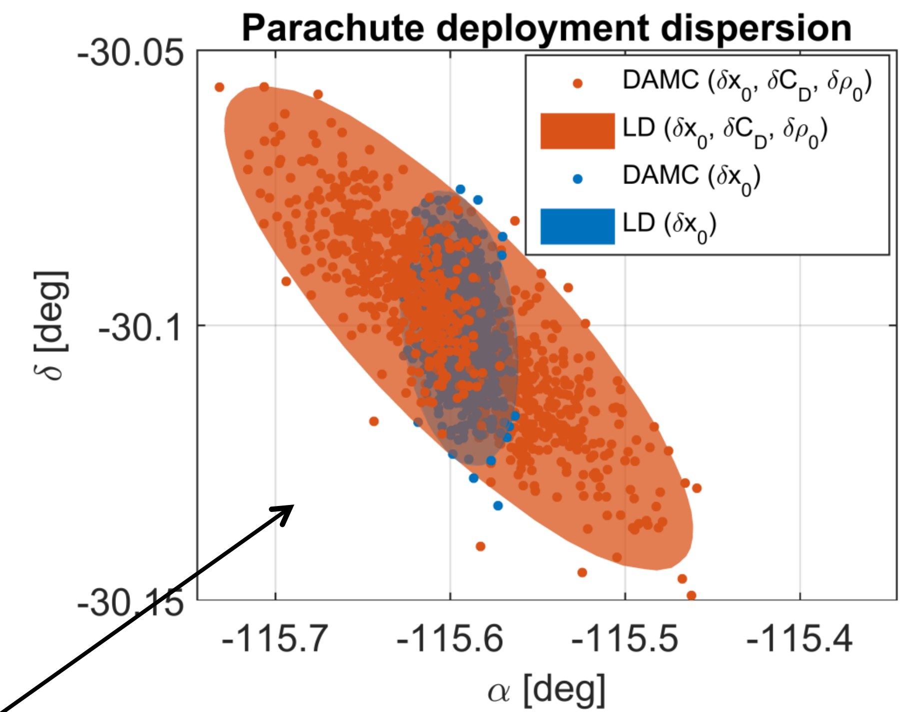
DAMC-k more efficient than classical MC for large number of samples



$$\tau = \frac{t_{DA} + n_s t_e}{n_s t_{pw}}$$

- Covariance Propagation





Uncertainty on density ρ_0 and drag coefficient C_D ➡ increased dispersion

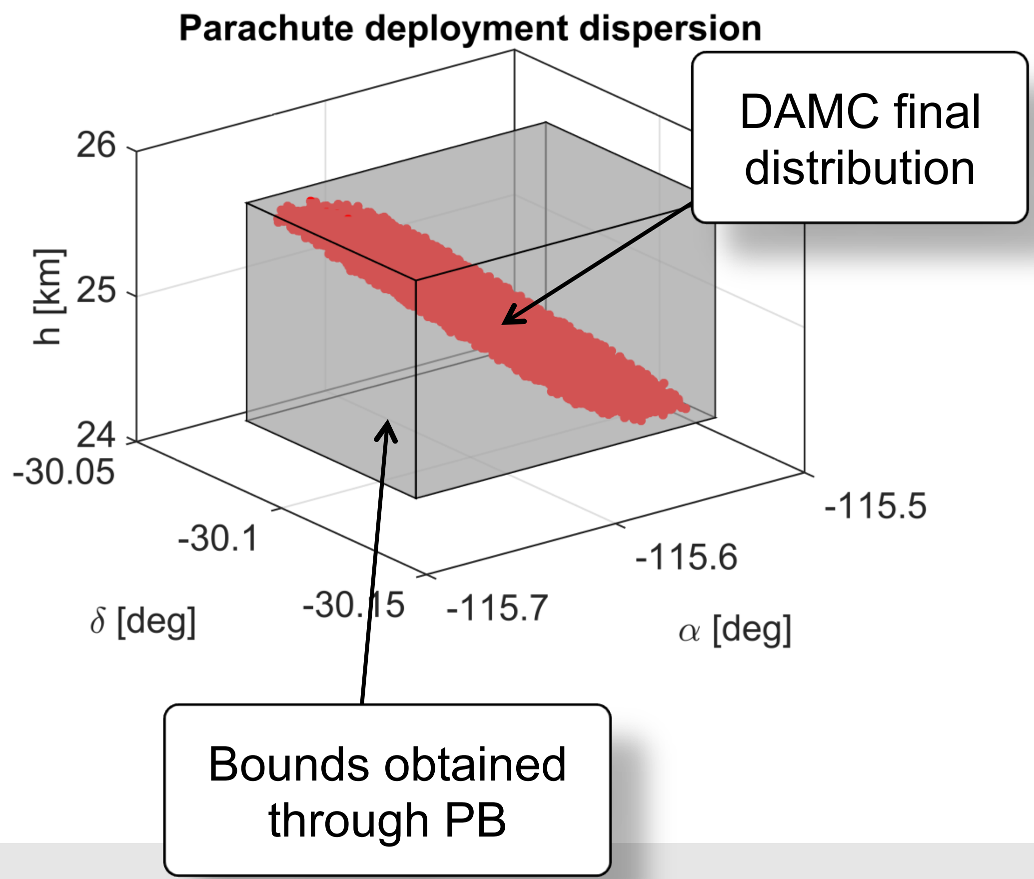
Applications: Re-entry Dynamics

- Polynomial Bounder

$$\varepsilon_{LB} = \frac{\left|\min\left(x_{f,DAMC}\right) - x_{f,LB}\right|}{x_{f,LB}}$$

$$\varepsilon_{UB} = \frac{\left|\min\left(x_{f,DAMC}\right) - x_{f,UB}\right|}{x_{f,UB}}$$

#samples	max(ε_{LB})	max(ε_{UB})
10^1	8.457e-03	4.104e-03
10^2	3.215e-03	1.979e-03
10^3	1.338e-03	8.025e-04
10^4	7.859e-04	4.634e-04
10^5	3.355e-04	4.457e-04

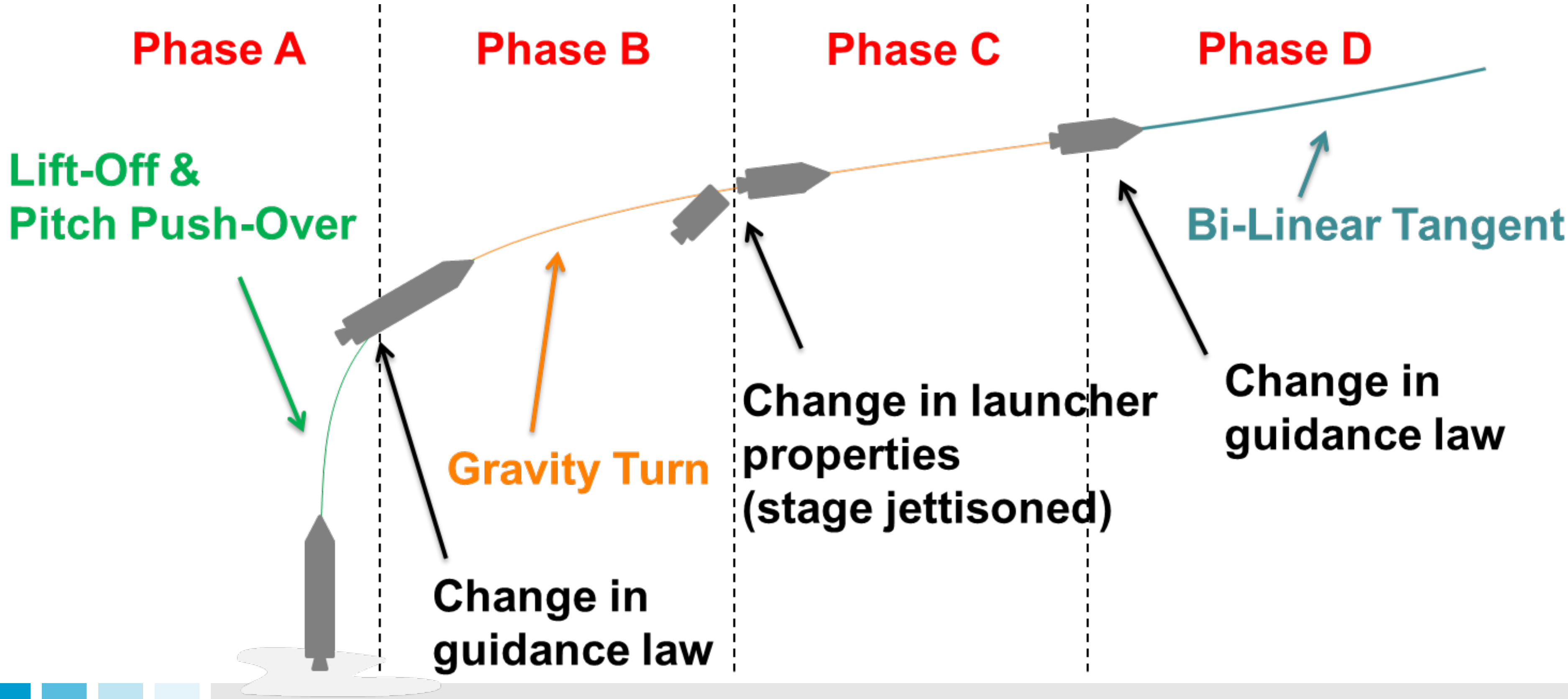


Available Dynamical Models

- The dynamical models available in the DAST are:
  - Two-Body
  - Three-Body
  - N-body
  - Re-entry
  - Relative
  - Attitude
  - Ascent
  - Rendezvous
  - Custom

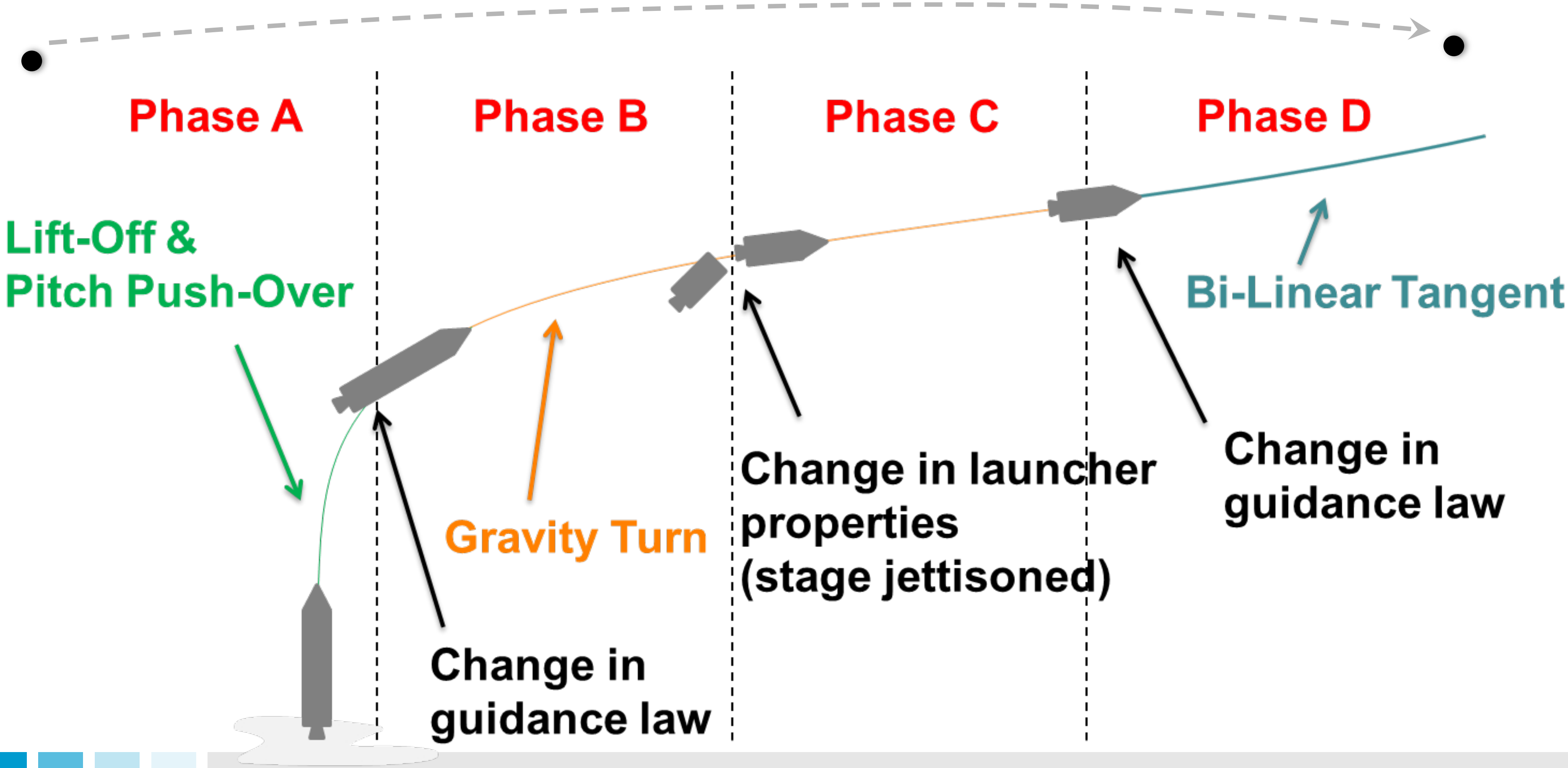
Ascent, Rendezvous → *Multi-phase approach*

MULTI-PHASE: ASCENT DYNAMICS

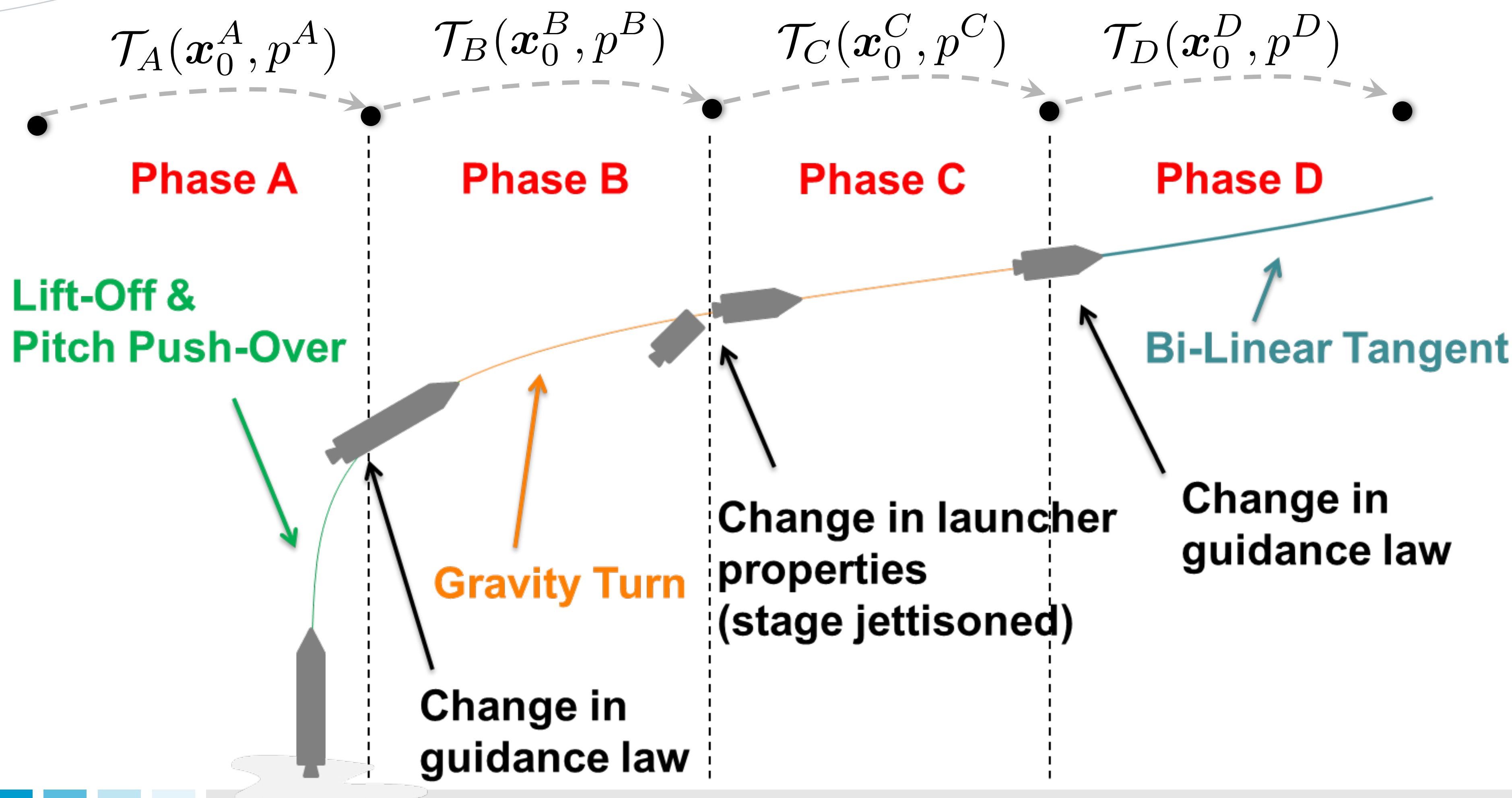


MULTI-PHASE: ASCENT DYNAMICS

$$\mathcal{T}(\boldsymbol{x}_0, p^A, p^B, p^C, p^D)$$

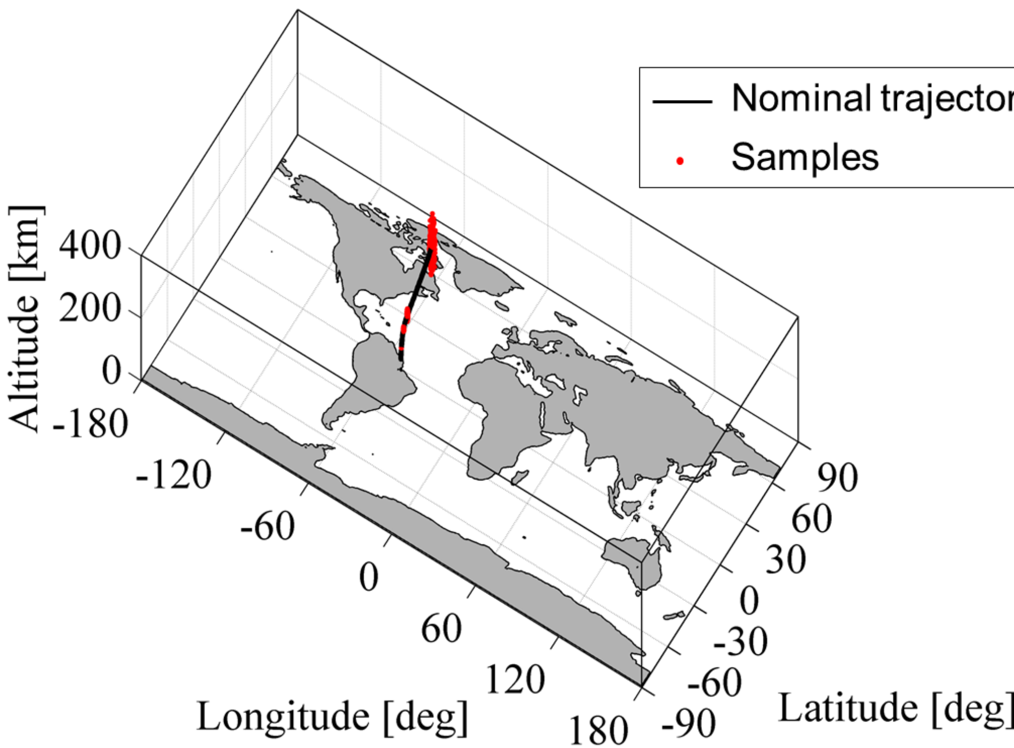


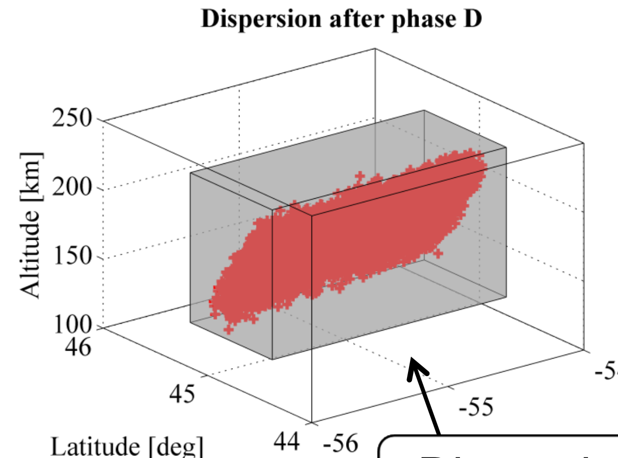
Multi-phase: Ascent Dynamics



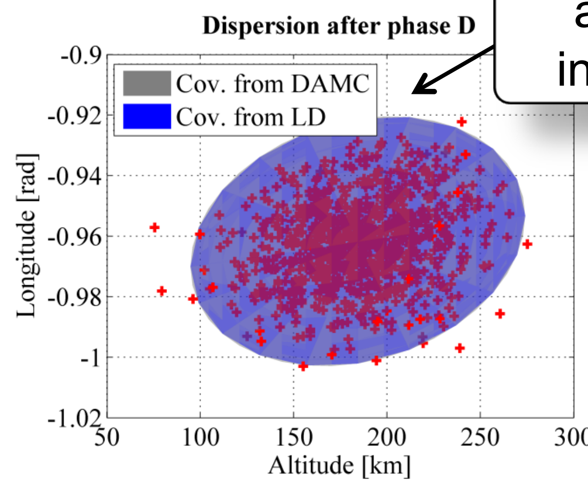
Multi-phase: Ascent Dynamics

- VEGA launcher





Dispersion at orbit insertion



CONCLUSIONS

- DAST is an efficient tool for nonlinear uncertainty propagation:
  - Propagations can be run on several ready-to-use dynamical models and any DA-compatible custom dynamical model
  - More efficient than standard Monte Carlo for typical number of samples
  - Analytical information available at the end of the propagation
- Note: method based on Taylor approximations ⇨ Size of uncertainty set and order shall guarantee sufficient accuracy
- If linear methods are sufficiently accurate for your application, you may not need to increase order, however…
- …DA relieves you from the "pain" of writing variational equations

DIFFERENTIAL ALGEBRA SPACE TOOLBOX FOR NONLINEAR UNCERTAINTY PROPAGATION IN SPACE DYNAMICS*

6th International Conference on Astrodynamics Tools and Techniques (ICATT)
Darmstadt, Germany

M. Rasotto, A. Morselli, A. Wittig, M. Massari, P. Di Lizia, R. Armellin, C. Y. Valles and G. Ortega

March 17th, 2016

** Developed under ITT AO/1-7570/13/NL/MH: Nonlinear Propagation of Uncertainties in Space Dynamics based on Taylor Differential Algebra"*