Uncertainty Propagation Using Differential Algebra

Armellin Roberto

with contributions by:

Pierluigi Di Lizia, Alexander Wittig, Alessandro Morselli, Monica Valli

Workshop on
Nonlinear Uncertainty Propagation using Differential Algebra
22nd September 2015

DINAMICA srl
Registered Office:
Piazza della Repubblica, 10 - 20121 - Milano (Italy)
Operational Headquarters:
Via Morghen, 13 - 20156 - Milano (Italy)
Phone +39 02 8342 2930
Fax +39 02 3206 6679
e-mail: dinamica@dinamicatech.com
website: www.dinamicatech.com

Outline

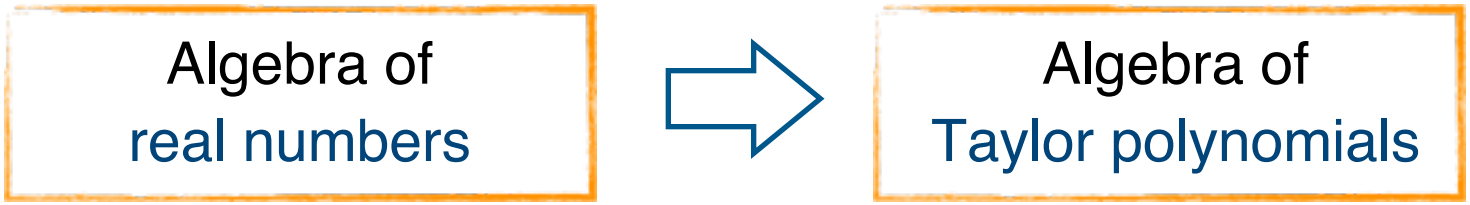
- Differential Algebra: basic concept
- Uncertainty propagation
  - ODEs flow expansion
  - Linear covariance propagation
  - DA-based Monte Carlo
  - DA-based statistical moments
  - Polynomial bounder
- Applications
  - SST: Impact probability computation
  - High-order filters
- Advanced methods
  - Automatic domain splitting

Outline

- Differential Algebra: basic concept
- Uncertainty propagation
  - ODEs flow expansion
  - Linear covariance propagation
  - DA-based Monte Carlo
  - DA-based statistical moments
  - Polynomial bounder
- Applications
  - SST: Impact probability computation
  - High-order filters
- Advanced methods
  - Automatic domain splitting

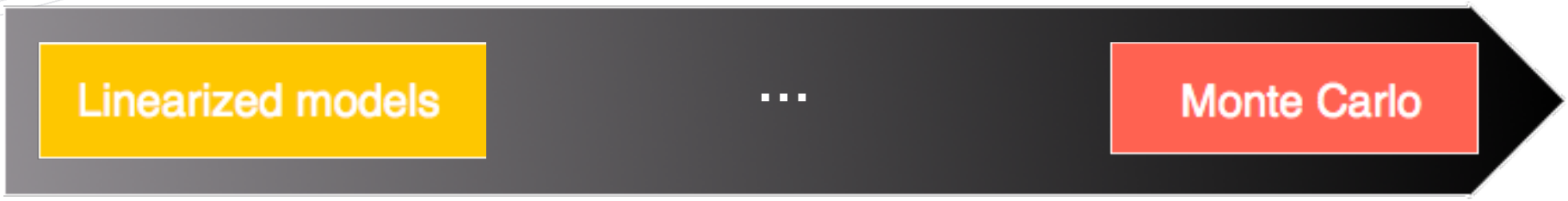
Differential algebra: basic concept

- Differential Algebra (DA) is an automatic differentiation technique



- Unlike standard automatic differentiation tools, the analytic operations of differentiation and antiderivation are introduced
- DA can be implemented in a computer environment (COSY-Infinity - Berz&Makino 1998, DACE, Jet Transport, JACK)
- Given any sufficiently regular function f of v, DA enables the computation of its Taylor expansion up to an arbitrary order k

Uncertainty propagation



- 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

Uncertainty propagation

low computational burden

low accuracy

computationally intensive

high accuracy

Can we find a compromise technique?

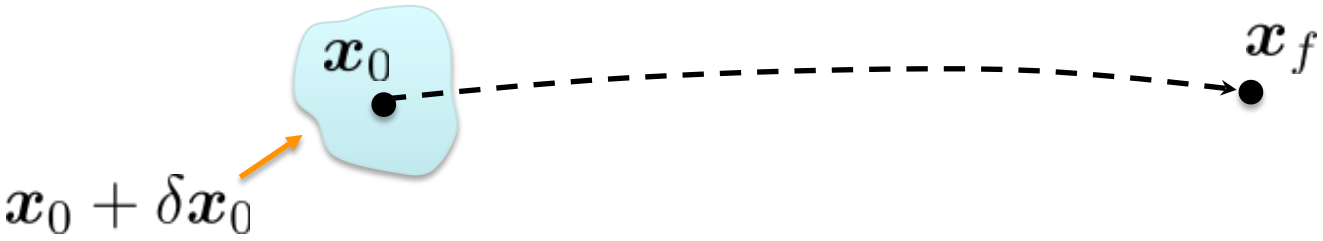
- We need a technique to:
  - Improve accuracy of linearized models
  - Reduce computational cost of classical Monte Carlo

Differential Algebra

Expansion of the flow of dynamics

- Given any dynamics

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



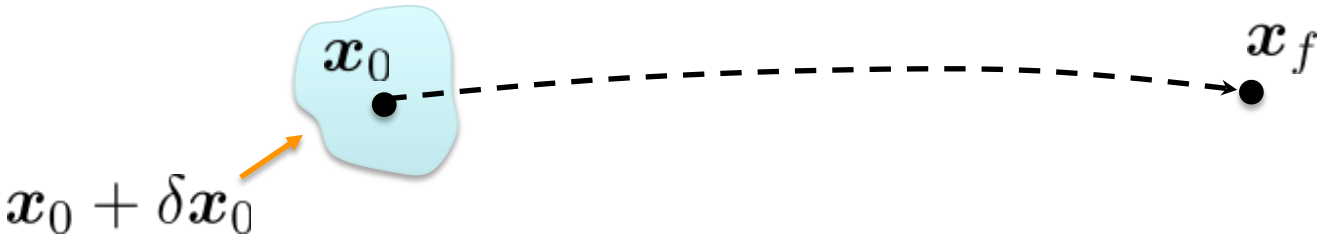
- Initialize initial conditions as a DA vector: $[\boldsymbol{x}_0] = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$
- Perform all operations of the integration scheme in the DA algebra

E.g.: Euler's scheme: $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{f}(\boldsymbol{x}_0) \cdot h$

Expansion of the flow of dynamics

- Given any dynamics

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



- Initialize initial conditions as a DA vector: $[\boldsymbol{x}_0] = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$
- Perform all operations of the integration scheme in the DA algebra

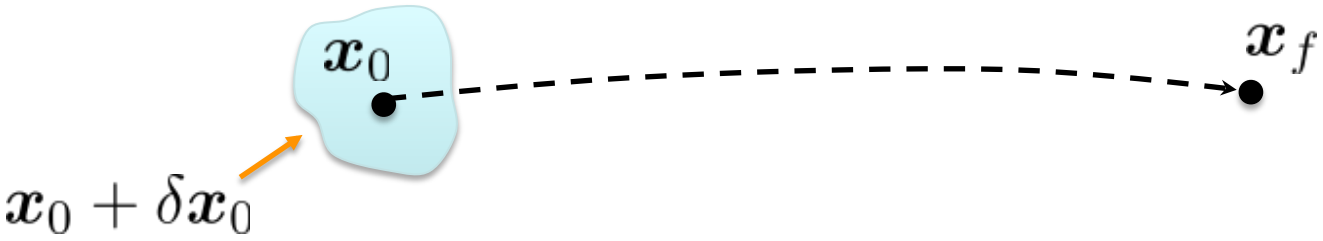
E.g.: Euler's scheme: $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{f}(\boldsymbol{x}_0) \cdot h$

$[\boldsymbol{x}_0]$

Expansion of the flow of dynamics

- Given any dynamics

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



- Initialize initial conditions as a DA vector: $[\boldsymbol{x}_0] = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$
- Perform all operations of the integration scheme in the DA algebra

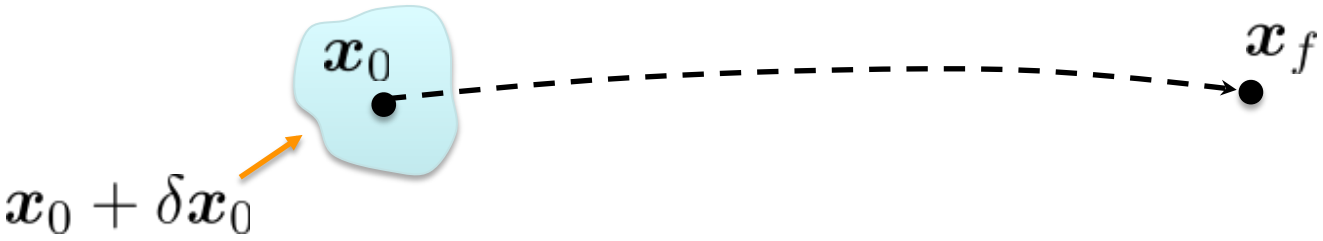
E.g.: Euler's scheme: $[\boldsymbol{x}_1] = [\boldsymbol{x}_0] + \boldsymbol{f}([\boldsymbol{x}_0]) \cdot h$

$[\boldsymbol{x}_1] = \mathcal{T}_{\boldsymbol{x}_1}(\delta\boldsymbol{x}_0)$: *k*-th order Taylor expansion of the solution at t_1

Expansion of the flow of dynamics

- Given any dynamics

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



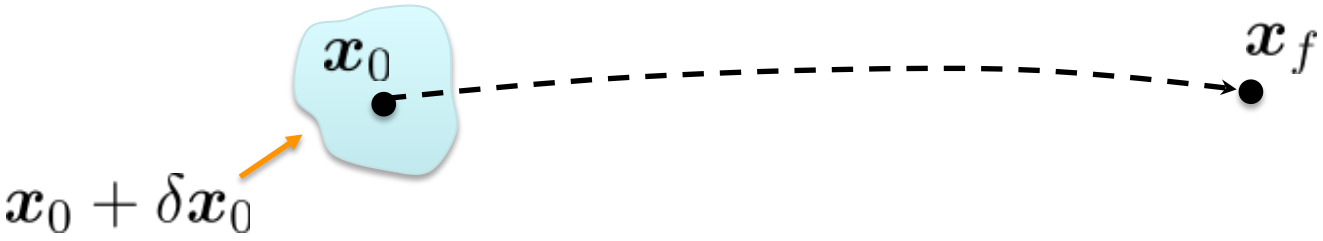
- Initialize initial conditions as a DA vector: $[\boldsymbol{x}_0] = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$
- Perform all operations of the integration scheme in the DA algebra

E.g.: Euler's scheme: $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{f}(\boldsymbol{x}_1) \cdot h$

Expansion of the flow of dynamics

- Given any dynamics

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



- Initialize initial conditions as a DA vector: $[\boldsymbol{x}_0] = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$
- Perform all operations of the integration scheme in the DA algebra

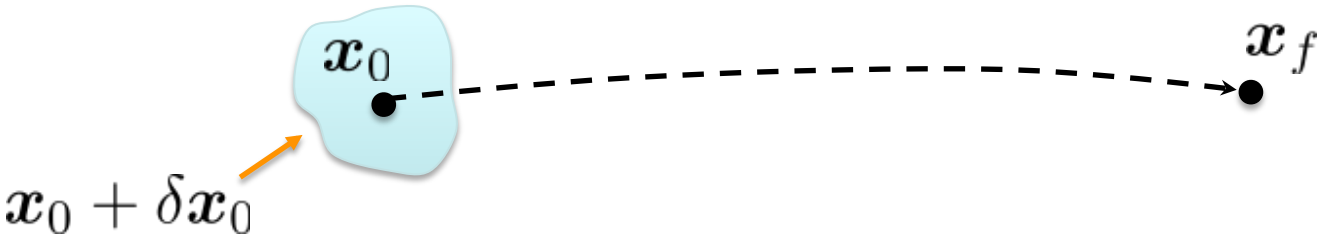
E.g.: Euler's scheme: $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{f}(\boldsymbol{x}_1) \cdot h$

$$[\boldsymbol{x}_1] = \mathcal{T}_{\boldsymbol{x}_1}(\delta\boldsymbol{x}_0)$$

Expansion of the flow of dynamics

- Given any dynamics

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



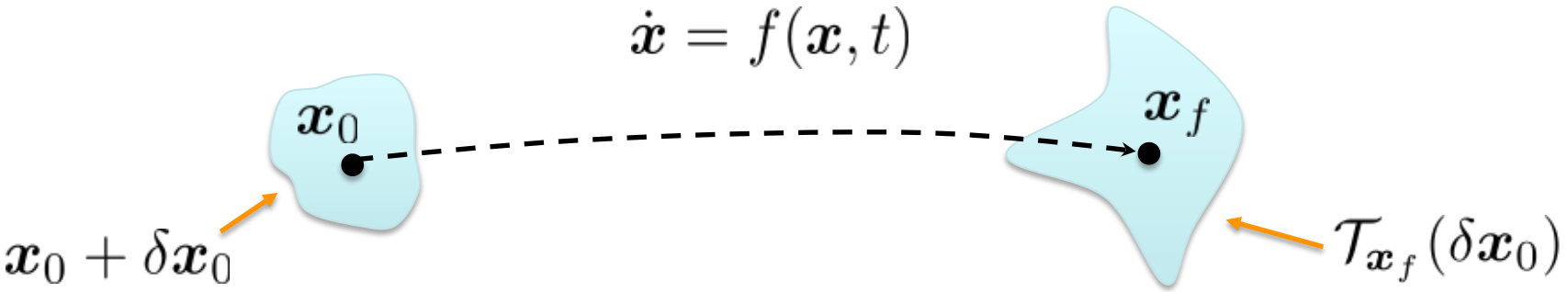
- Initialize initial conditions as a DA vector: $[\boldsymbol{x}_0] = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$
- Perform all operations of the integration scheme in the DA algebra

E.g.: Euler's scheme: $[\boldsymbol{x}_2] = [\boldsymbol{x}_1] + \boldsymbol{f}([\boldsymbol{x}_1]) \cdot h$

$[\boldsymbol{x}_2] = \mathcal{T}_{\boldsymbol{x}_2}(\delta\boldsymbol{x}_0)$: *k*-th order Taylor expansion of the solution at t_2

Expansion of the flow of dynamics

- Given any dynamics



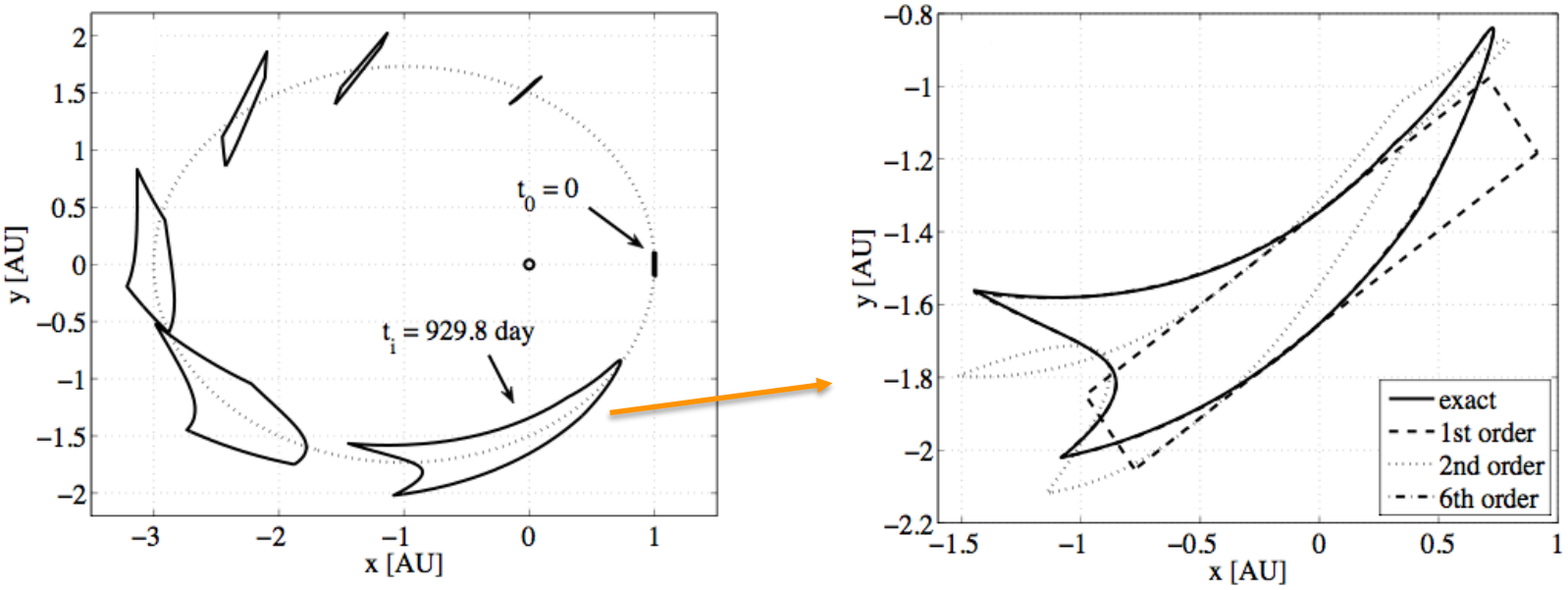
- Initialize initial conditions as a DA vector: $[\boldsymbol{x}_0] = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$
- Perform all operations of the integration scheme in the DA algebra

E.g.: Euler's scheme: $[\boldsymbol{x}_f] = [\boldsymbol{x}_{f-1}] + \boldsymbol{f}([\boldsymbol{x}_{f-1}]) \cdot h$

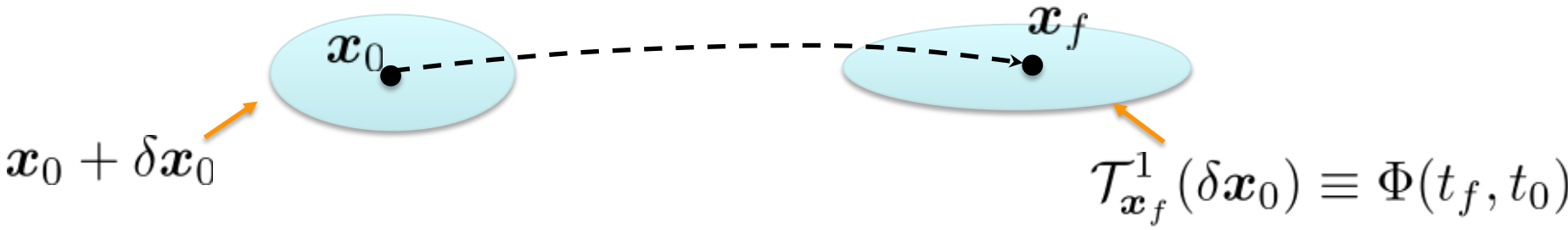
$[\boldsymbol{x}_f] = \mathcal{T}_{\boldsymbol{x}_f}(\delta\boldsymbol{x}_0)$: *k*-th order Taylor expansion of the solution at t_f

Expansion of the flow of dynamics

- Example: 2-body dynamics
  - Eccentricity: 0.5 – Starting point: pericenter
  - Integration scheme: Runge-Kutta (variable step, order 8)
  - DA-based ODE flow expansion order: 6
- Initial box: 0.008 AU in *x* and 0.08 AU in *y*



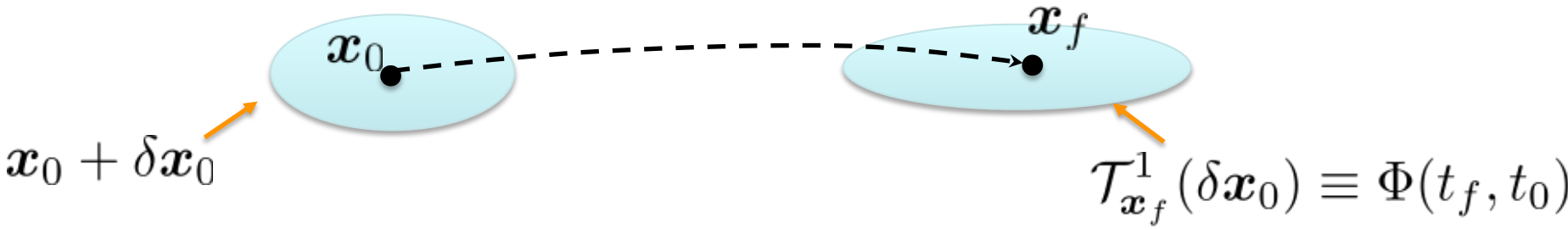
Linear covariance propagation



- 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** $\Phi(t_f, t_0)$

  ⇨ Initial covariance C_0 can be propagated as: $C_f = \Phi\, C_0\, \Phi^T$

Linear covariance propagation

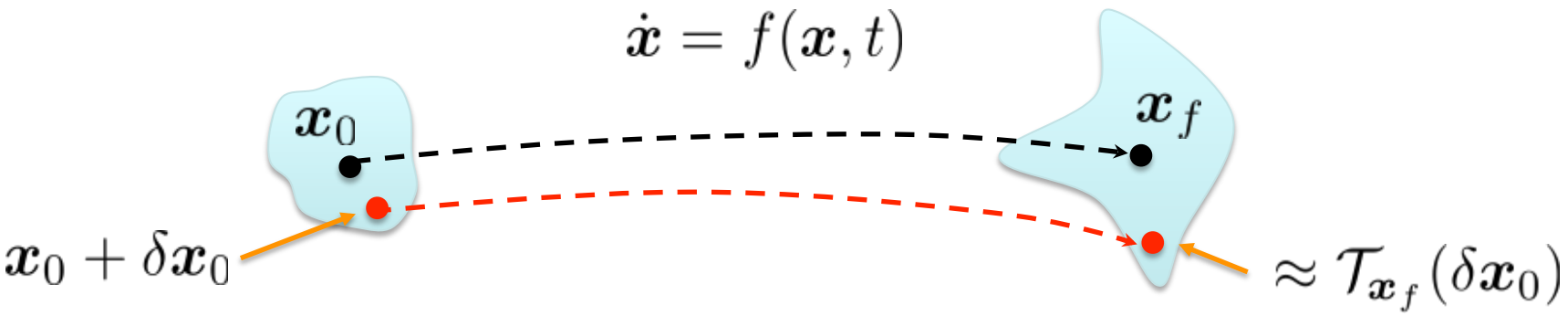


- If all computations are performed to **order 1**, the final Taylor expansion $\mathcal{T}^1_{\boldsymbol{x}_f}(\delta\boldsymbol{x}_0)$ coincides with the **state-transition matrix** $\Phi(t_f, t_0)$

⇨ Initial covariance C_0 can be propagated as: $C_f = \Phi\, C_0\, \Phi^T$

No need of variational equations!

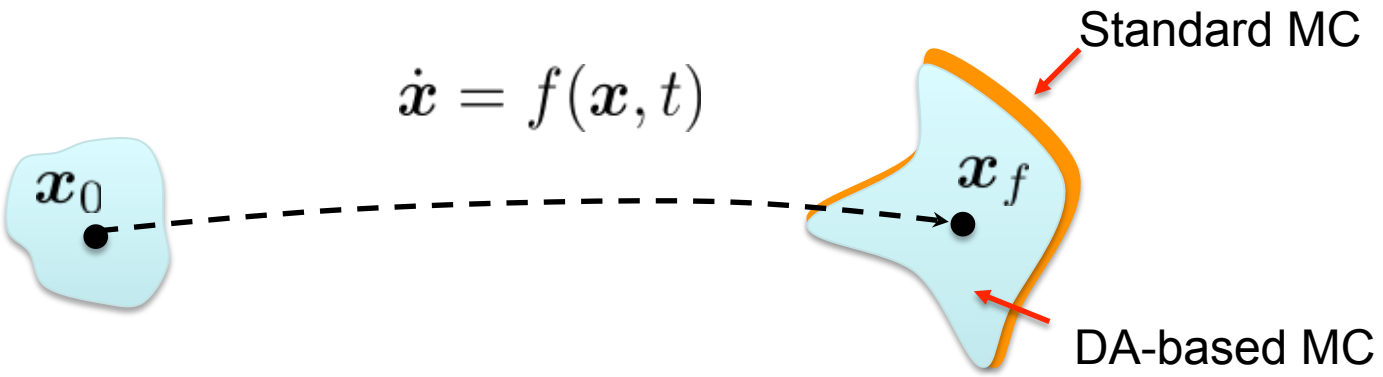
DA-based Monte Carlo



- 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

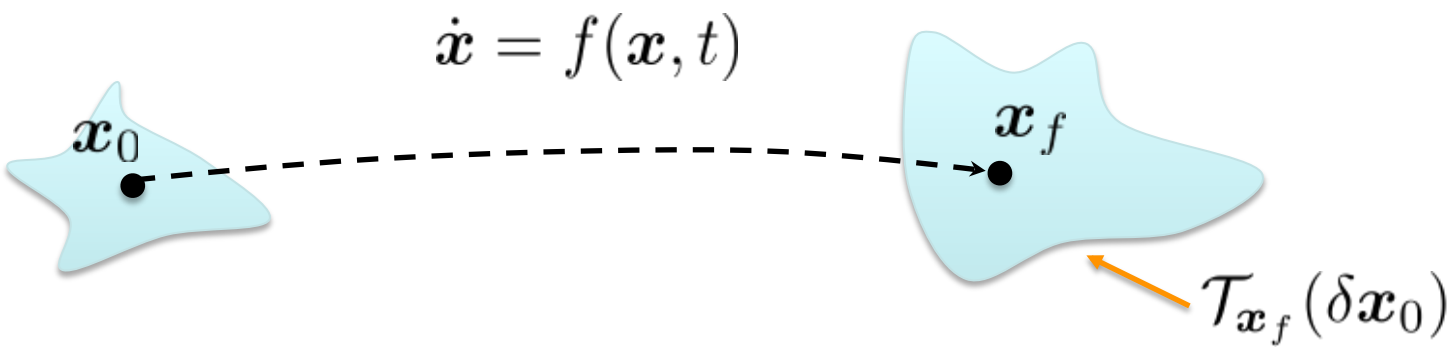
DA-based Monte Carlo



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

- Note:
  - The accuracy of Taylor expansion can be tuned with order
  - The same polynomial can be used to map different statistics

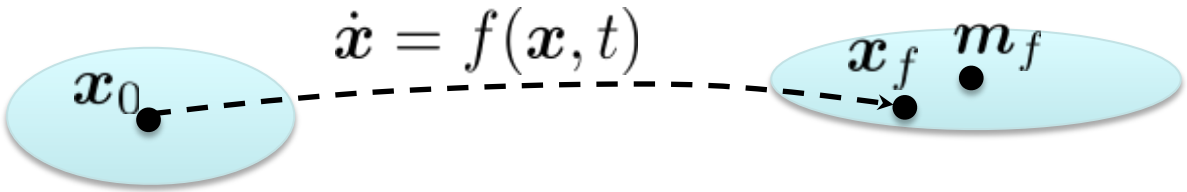
DA-based Monte Carlo



- Any pointwise integration can be replaced by the evaluation of the polynomial $\mathcal{T}_{\boldsymbol{x}_f}(\delta\boldsymbol{x}_0)$

- Note:
  - The accuracy of Taylor expansion can be tuned with order
  - The same polynomial can be used to map different statistics

DA-based statistical moments



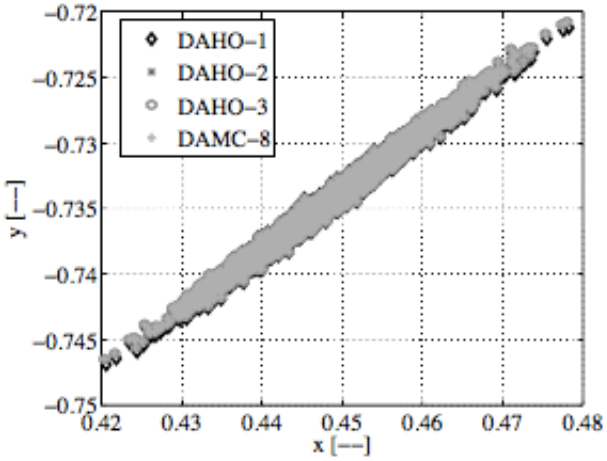
- Given the k-order polynomial map: $[\boldsymbol{x}_f] = \mathcal{T}_{\boldsymbol{x}_f}(\delta \boldsymbol{x}_0)$
- Moments of the pdf can be $\boldsymbol{x}_f$ obtained by computing the expectation of the Taylor expansion: for the mean

$$E[\boldsymbol{x}_f] = \sum_{p=1}^{k} \frac{1}{p!} \Phi_{1\ldots p} E[\delta x_{0,1} \ldots \delta x_{0,p}]$$

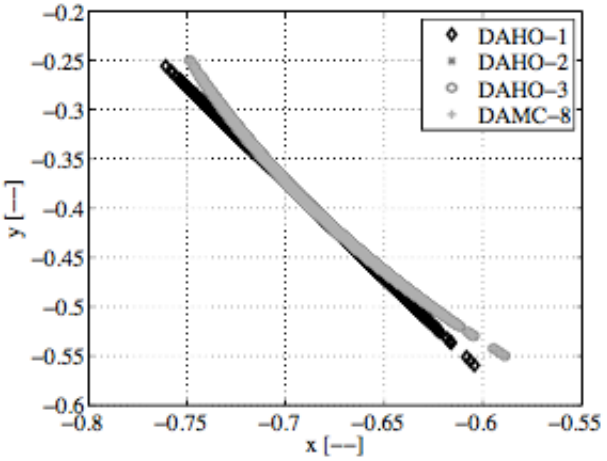
computed by Isserlis's formula

- Same approach for high order moments (covariance, skeweness, kurtosis,…)

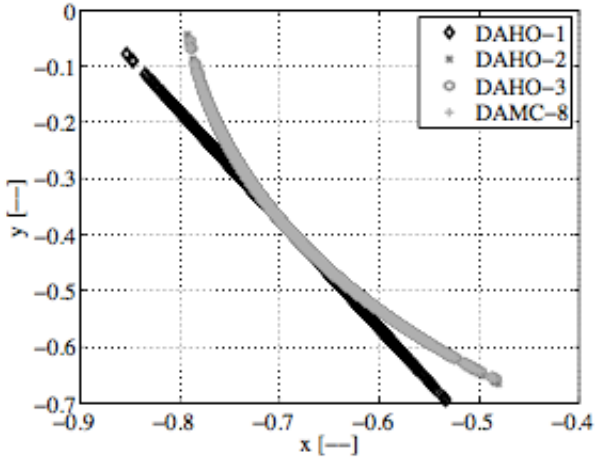
DA-based statistical moments



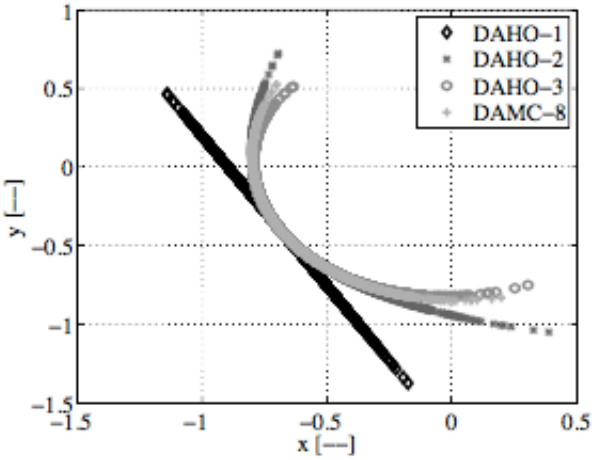
(a) Samples distribution after 0.8 orbit



(b) Samples distribution after 5 orbits

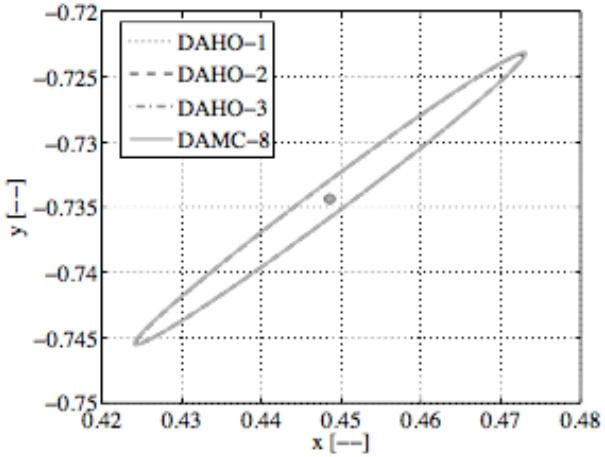


(c) Samples distribution after 10 orbits

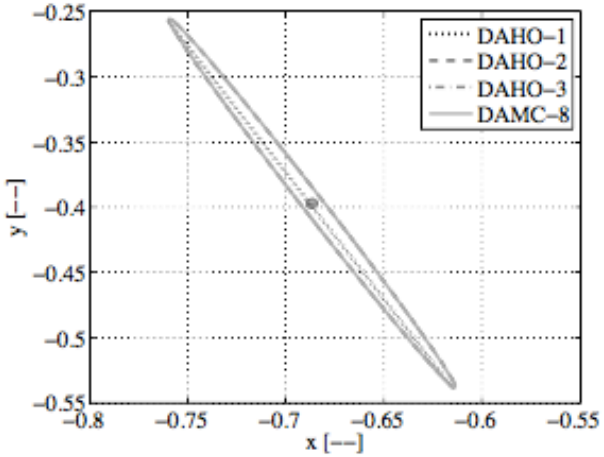


(d) Samples distribution after 30 orbits

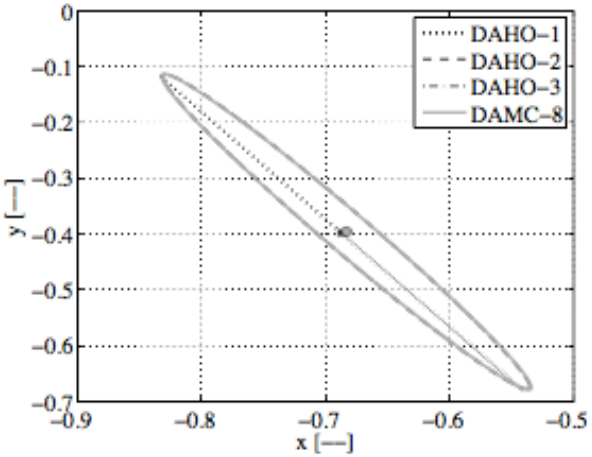
DA-based statistical moments



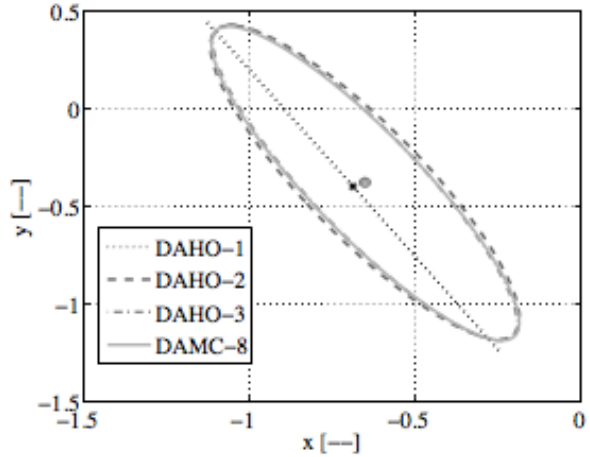
(a) Uncertainty ellipse after 0.8 orbit



(b) Uncertainty ellipse after 5 orbits

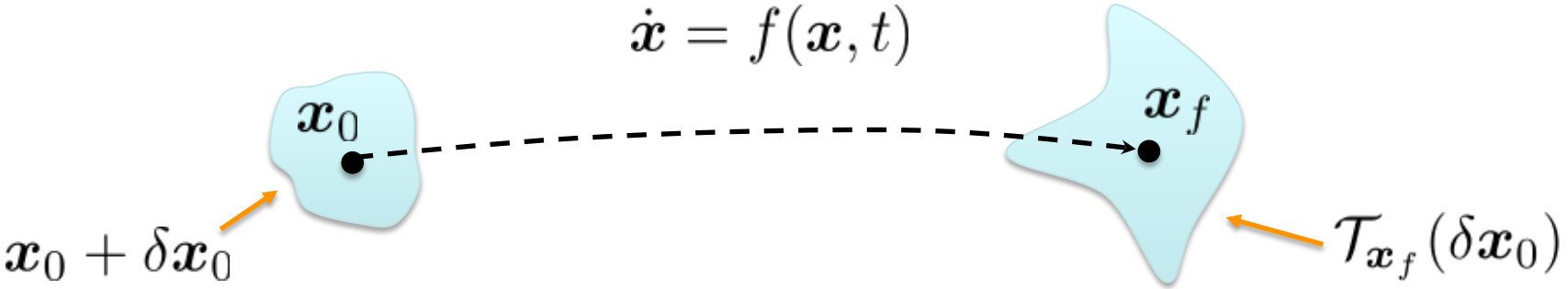


(c) Uncertainty ellipse after 10 orbits

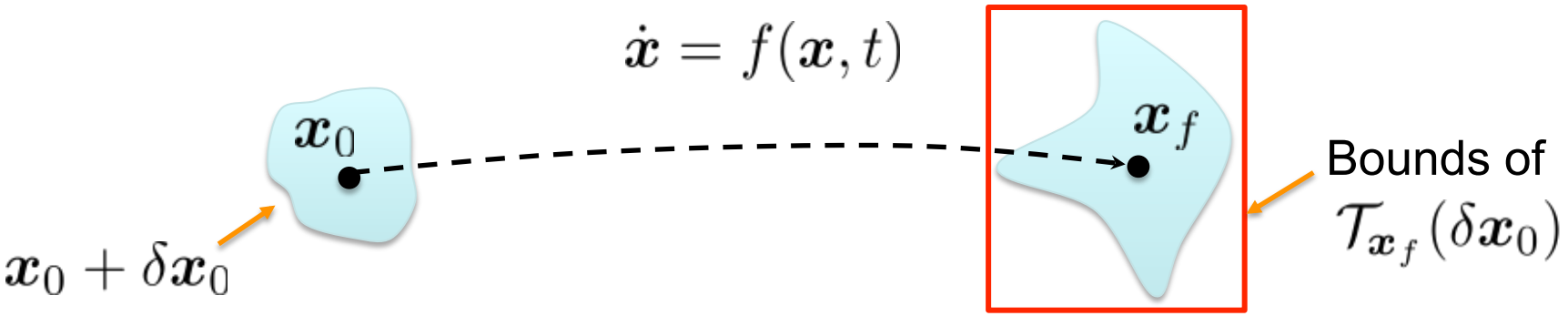


(d) Uncertainty ellipse after 30 orbits

Polynomial bounder

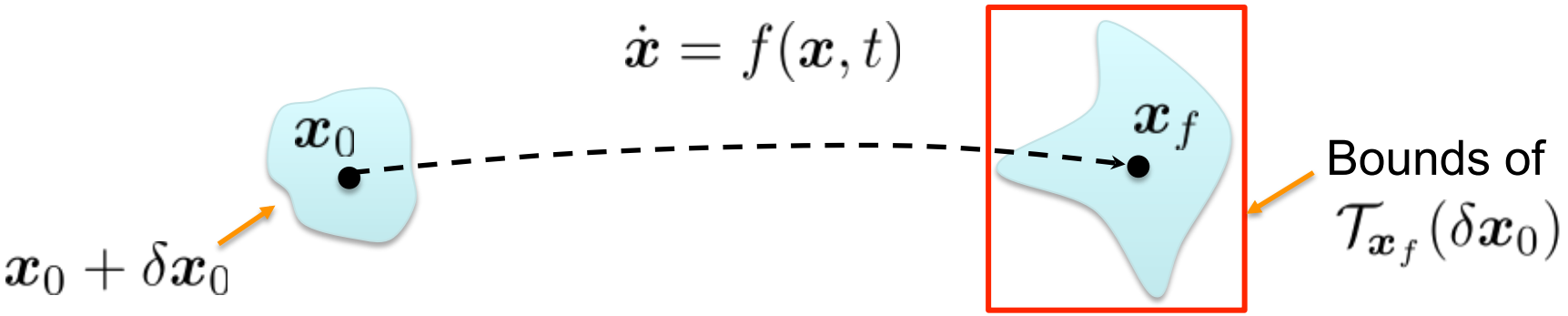


Polynomial bounder



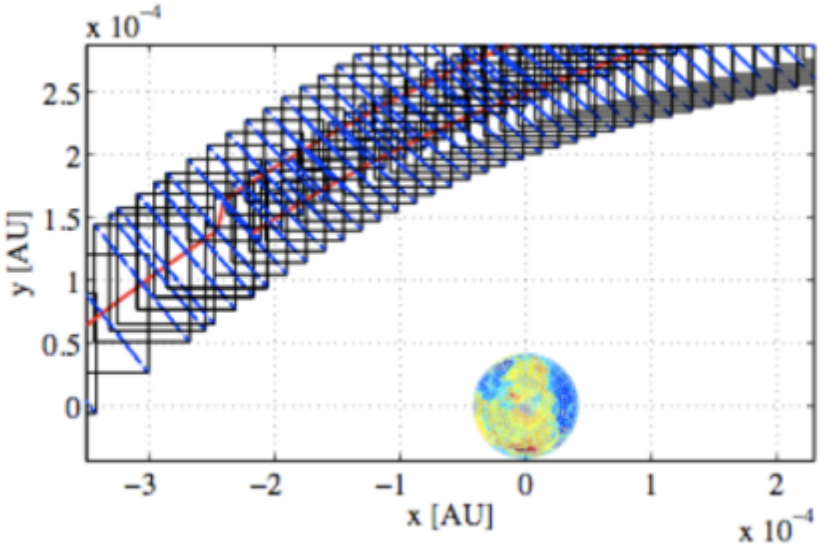
- Polynomial bounders can immediately estimate the range of $\boldsymbol{x}_f$

Polynomial bounder



- Polynomial bounders can immediately estimate the range of x_f

- Example: very useful to discard the occurrence of an event
  - Impact of NEOs with Earth

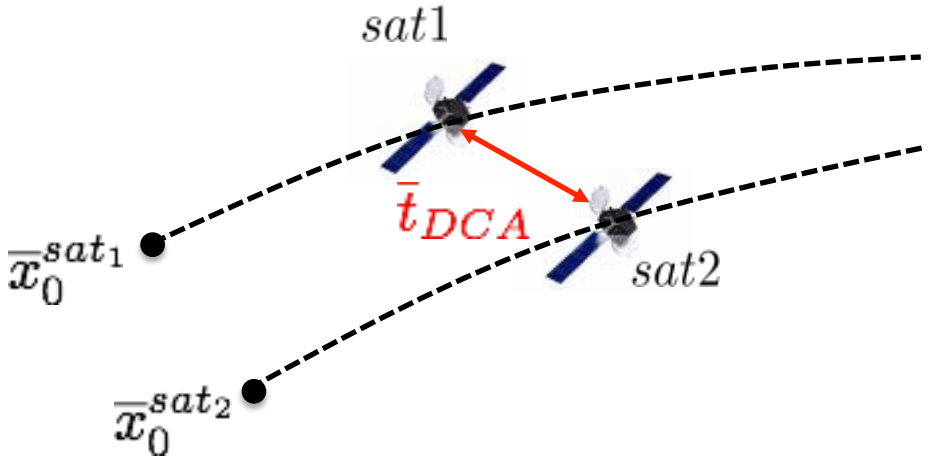


Outline

- Differential Algebra: basic concept
- Uncertainty propagation
  - ODEs flow expansion
  - Linear covariance propagation
  - DA-based Monte Carlo
  - DA-based statistical moments
  - Polynomial bounder
- Applications
  - SST: Impact probability computation
  - High-order filters
- Advanced methods
  - Automatic domain splitting

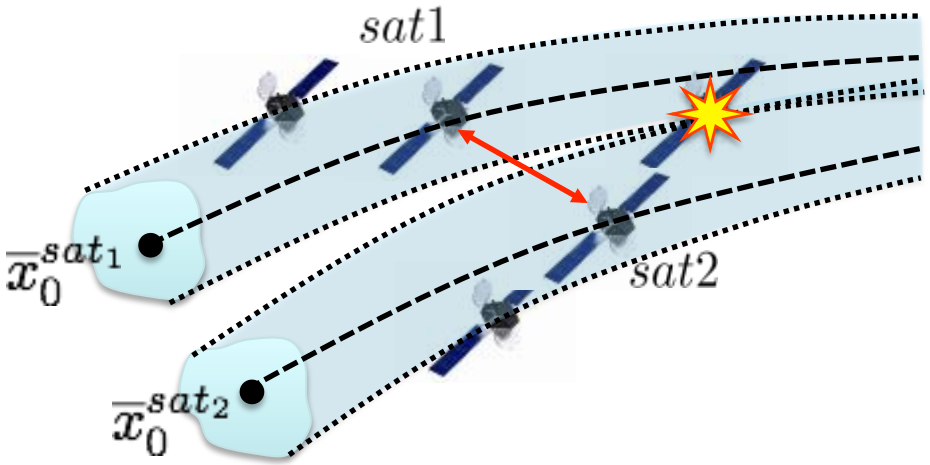
SST: impact probability computation

- Suppose no impact occurrence for nominal initial states



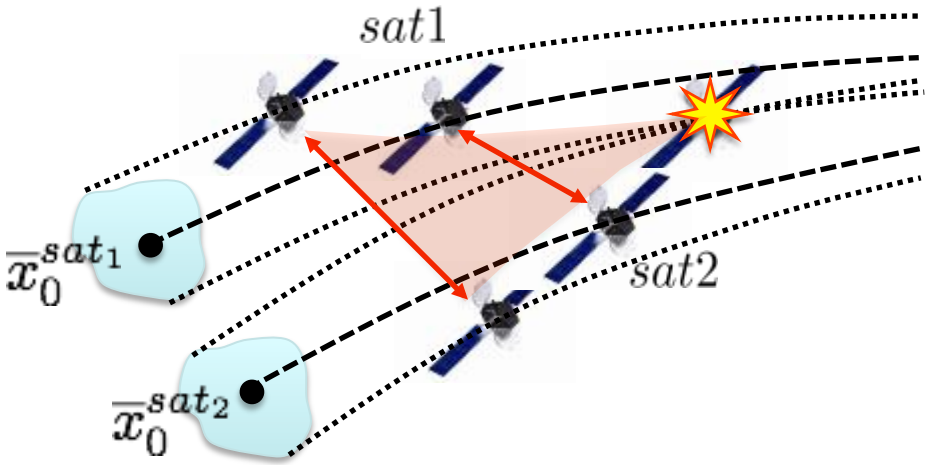
SST: impact probability computation

- Suppose no impact occurrence for nominal initial states
- Impact conditions may occur for $x_0^{sat_i} \neq \overline{x}_0^{sat_i}$ at $t_{DCA} \neq \bar{t}_{DCA}$



SST: impact probability computation

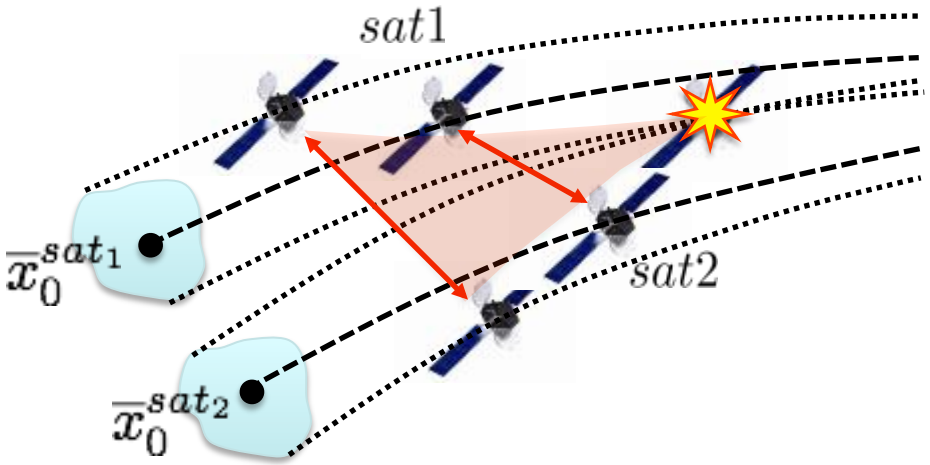
- Suppose no impact occurrence for nominal initial states
- Impact conditions may occur for $x_0^{sat_i} \neq \overline{x}_0^{sat_i}$ at $t_{DCA} \neq \bar{t}_{DCA}$
- DA can be used to compute the Taylor expansion of the distance of closest approach (DCA) w.r.t. initial states:



SST: impact probability computation

- Suppose no impact occurrence for nominal initial states
- Impact conditions may occur for $x_0^{sat_i} \neq \bar{x}_0^{sat_i}$ at $t_{DCA} \neq \bar{t}_{DCA}$
- DA can be used to compute the Taylor expansion of the distance of closest approach (DCA) w.r.t. initial states:
  - Compute: $[x_f^{sat_i}] = \mathcal{T}_{x_f^{sat_i}}(\delta x_0^{sat_i}, \delta t_f)$ for $i = 1, 2$

$$\Rightarrow [d^2] = \mathcal{T}_{d^2}(\delta x_0^{sat_i}, \delta t_f) \Rightarrow [\partial d^2 / \partial t_f] = \mathcal{T}_{d^2}(\delta x_0^{sat_i}, \delta t_f)$$



SST: impact probability computation

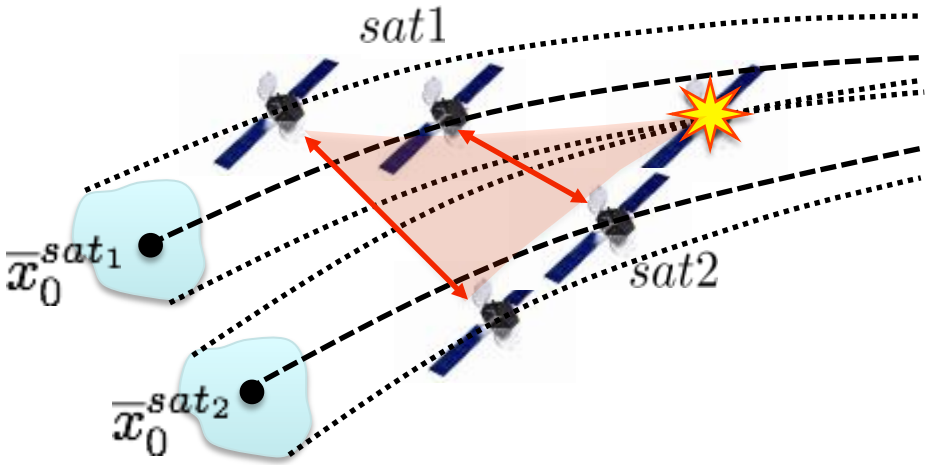
- Suppose no impact occurrence for nominal initial states
- Impact conditions may occur for $x_0^{sat_i} \neq \bar{x}_0^{sat_i}$ at $t_{DCA} \neq \bar{t}_{DCA}$
- DA can be used to compute the Taylor expansion of the distance of closest approach (DCA) w.r.t. initial states:
  - Compute: $[x_f^{sat_i}] = \mathcal{T}_{x_f^{sat_i}}(\delta x_0^{sat_i}, \delta t_f)$ for $i = 1, 2$

    $\Rightarrow [d^2] = \mathcal{T}_{d^2}(\delta x_0^{sat_i}, \delta t_f) \Rightarrow [\partial d^2/\partial t_f] = \mathcal{T}_{d^2}(\delta x_0^{sat_i}, \delta t_f)$
  - Invert:

    $[\delta t_f] = \mathcal{T}_{d^2}(\delta x_0^{sat_i}, \partial d^2/\partial t_f)$
  - Impose $\partial d^2/\partial t_f = 0$:

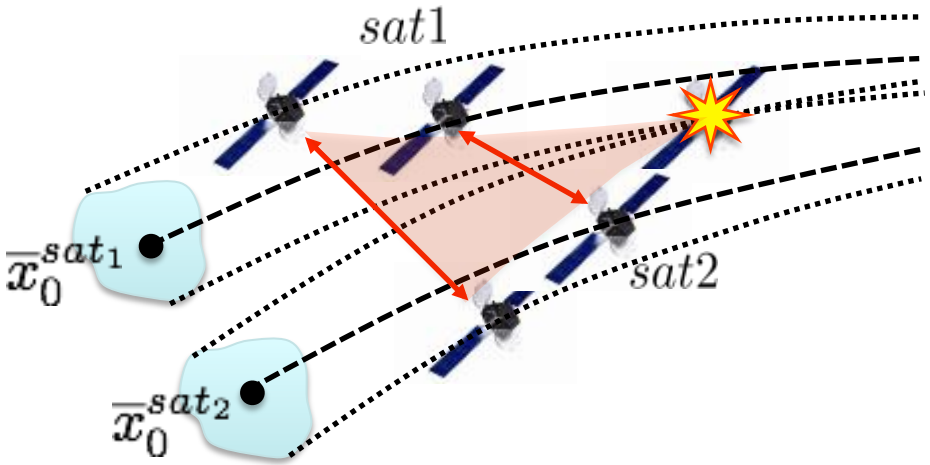
    $[t_{DCA}] = \mathcal{T}_{\partial d^2=0}(\delta x_0^{sat_i})$

    $[d_{DCA}] = \mathcal{T}_{DCA}(\delta x_0^{sat_i})$



SST: impact probability computation

$$[d_{DCA}] = \mathcal{T}_{DCA}(\delta x_0^{sat_i})$$



- DA-based fast Monte Carlo simulations can be run on $\mathcal{T}_{DCA}$ to compute # of samples with

  $\mathcal{T}_{DCA} <$ threshold

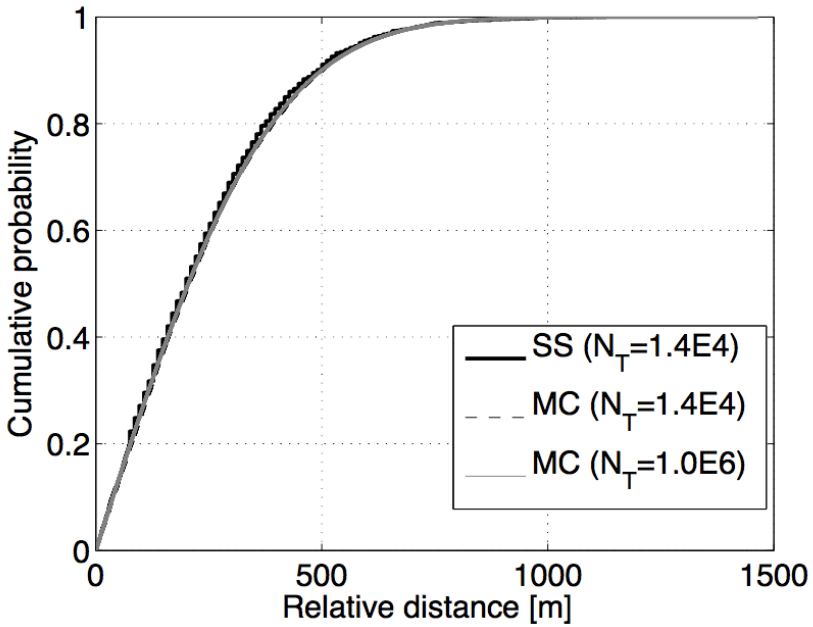
- The map $\mathcal{T}_{DCA}(\delta x_0^{sat_i})$ was used for:
  - DA-based Monte Carlo (DAMC)
  - DA-based Line Sampling (DALS)
  - DA-based Subset Simulation (DASS)

  ⇨ more efficient for low probabilities

- Different orbital regimes: LEO, MEO, GEO

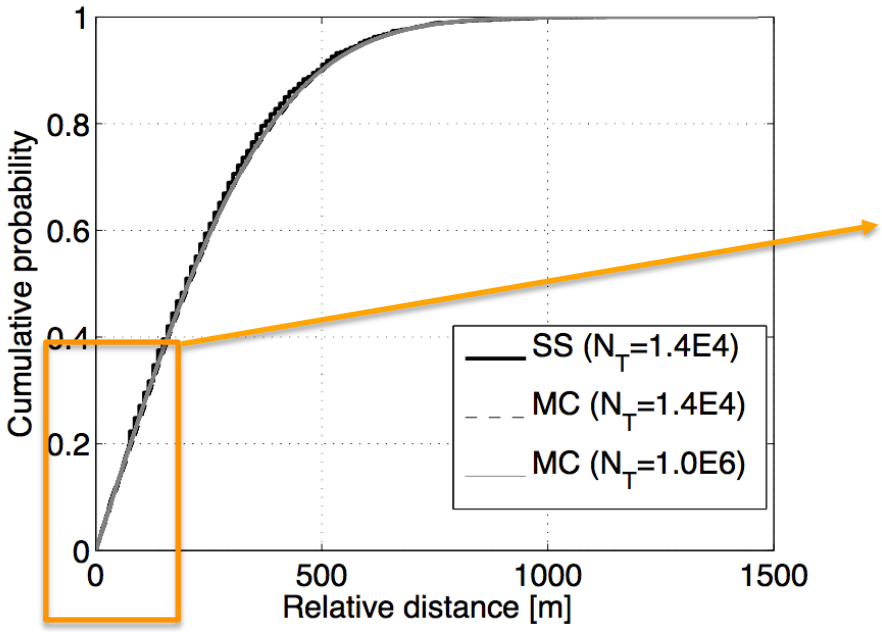
SST: impact probability computation

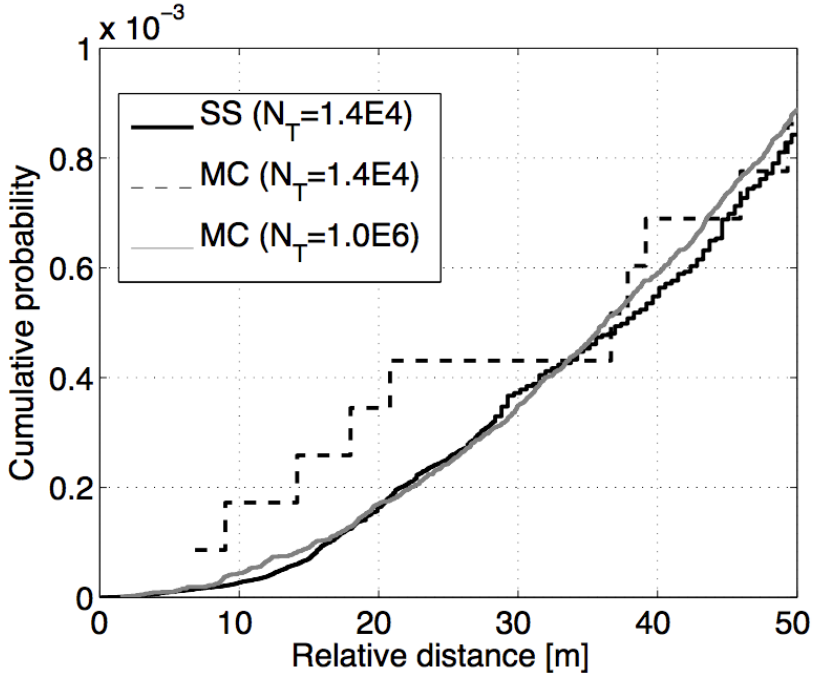
- Efficiency of LS and SS over standard Monte Carlo for low probabilities:
  - Cumulative impact probability distribution for a close conjunction
  - # of samples: 14000



SST: impact probability computation

- Efficiency of LS and SS over standard Monte Carlo for low probabilities:
  - Cumulative impact probability distribution for a close conjunction
  - # of samples: 14000
  - No samples generated by standard MC below 5 m distance





SST: impact probability computation

- Efficiency of DAMC, DASS, and DALS over standard MC
- Three test cases (Alfano, 2009)
  - Test 5: linear relative motion ($P_c = 4.440\text{E-}2$)
  - Test 6: boundary of linear relative motion ($P_c = 4.324\text{E-}3$)
  - Test 7: nonlinear relative motion ($P_c = 1.580\text{E-}4$)

Test case	Method	P_c [-]	N_T	t_c [s]
5	MC	4.452E-2	1.0E+5	4.75
	DAMC-3	4.459E-2	1.0E+5	0.67
	DALS-3	4.451E-2	5.0E+3	2.53
	DASS-3	4.450E-2	2.0E+4	0.13
6	MC	4.339E-3	1.0E+6	43.21
	DAMC-3	4.350E-3	1.0E+6	6.67
	DALS-3	4.341E-3	5.0E+3	2.58
	DASS-3	4.328E-3	4.0E+4	0.27
7	MC	1.615E-4	2.7E+7	1155.36
	DAMC-3	1.612E-4	2.7E+7	179.34
	DALS-3	1.621E-4	5.0E+3	1.43
	DASS-3	1.626E-4	6.0E+4	0.43

* Intel Core i5 2500 @3.30GHz, 8Gb RAM

SST: impact probability computation

- Efficiency of DAMC, DASS, and DALS over standard MC
- Three test cases (Alfano, 2009)
  - Test 5: linear relative motion ($P_c = 4.440\text{E-}2$)
  - Test 6: boundary of linear relative motion ($P_c = 4.324\text{E-}3$)
  - Test 7: nonlinear relative motion ($P_c = 1.580\text{E-}4$)

Test case	Method	P_c [-]	N_T	t_c [s]
5	MC	4.452E-2	1.0E+5	4.75
	DAMC-3	4.459E-2	1.0E+5	0.67
	DALS-3	4.451E-2	5.0E+3	2.53
	DASS-3	4.450E-2	2.0E+4	0.13
6	MC	4.339E-3	1.0E+6	43.21
	DAMC-3	4.350E-3	1.0E+6	6.67
	DALS-3	4.341E-3	5.0E+3	2.58
	DASS-3	4.328E-3	4.0E+4	0.27
7	MC	1.615E-4	2.7E+7	1155.36
	DAMC-3	1.612E-4	2.7E+7	179.34
	DALS-3	1.621E-4	5.0E+3	1.43
	DASS-3	1.626E-4	6.0E+4	0.43

* Intel Core i5 2500 @3.30GHz, 8Gb RAM

SST: impact probability computation

- Efficiency of DAMC, DASS, and DALS over standard MC
- Three test cases (Alfano, 2009)
  - Test 5: linear relative motion ($P_c = 4.440\text{E-}2$)
  - Test 6: boundary of linear relative motion ($P_c = 4.324\text{E-}3$)
  - Test 7: nonlinear relative motion ($P_c = 1.580\text{E-}4$)

Test case	Method	P_c [-]	N_T	t_c [s]
5	MC	4.452E-2	1.0E+5	4.75
	DAMC-3	4.459E-2	1.0E+5	0.67
	DALS-3	4.451E-2	5.0E+3	2.53
	DASS-3	4.450E-2	2.0E+4	0.13
6	MC	4.339E-3	1.0E+6	43.21
	DAMC-3	4.350E-3	1.0E+6	6.67
	DALS-3	4.341E-3	5.0E+3	2.58
	DASS-3	4.328E-3	4.0E+4	0.27
7	MC	1.615E-4	2.7E+7	1155.36
	DAMC-3	1.612E-4	2.7E+7	179.34
	DALS-3	1.621E-4	5.0E+3	1.43
	DASS-3	1.626E-4	6.0E+4	0.43

* Intel Core i5 2500 @3.30GHz, 8Gb RAM

SST: impact probability computation

- Efficiency of DAMC, DASS, and DALS over standard MC
- Three test cases (Alfano, 2009)
  - Test 5: linear relative motion ($P_c = 4.440\text{E-}2$)
  - Test 6: boundary of linear relative motion ($P_c = 4.324\text{E-}3$)
  - Test 7: nonlinear relative motion ($P_c = 1.580\text{E-}4$)

Test case	Method	P_c [-]	N_T	t_c [s]
5	MC	4.452E-2	1.0E+5	4.75
	DAMC-3	4.459E-2	1.0E+5	0.67
	DALS-3	4.451E-2	5.0E+3	2.53
	DASS-3	4.450E-2	2.0E+4	0.13
6	MC	4.339E-3	1.0E+6	43.21
	DAMC-3	4.350E-3	1.0E+6	6.67
	DALS-3	4.341E-3	5.0E+3	2.58
	DASS-3	4.328E-3	4.0E+4	0.27
7	MC	1.615E-4	2.7E+7	1155.36
	DAMC-3	1.612E-4	2.7E+7	179.34
	DALS-3	1.621E-4	5.0E+3	1.43
	DASS-3	1.626E-4	6.0E+4	0.43

* Intel Core i5 2500 @3.30GHz, 8Gb RAM

High-order Kalman filters

- Given the discrete system model:

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k$$

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}, t_{k+1}) + \mathbf{v}_{k+1}$$

- Suppose to have the state $\mathbf{x}_k$, with mean $\mathbf{m}_k^+$ and covariance $\mathbf{P}_k^+$
- General filtering algorithm:
  - Prediction:

$$\mathbf{m}_{k+1}^- = E[\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k \mid \mathbf{z}_k]$$

$$\mathbf{P}_{k+1}^- = E\{[\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k][\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k]^T \mid \mathbf{z}_k\} - (\mathbf{m}_{k+1}^-)(\mathbf{m}_{k+1}^-)^T$$

$$\mathbf{n}_{k+1}^- = E[\mathbf{h}(\mathbf{x}_{k+1}, t_{k+1}) + \mathbf{v}_{k+1} \mid \mathbf{z}_k]$$

  - Update:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{xz}\left(\mathbf{P}_{k+1}^{zz}\right)^{-1} \Rightarrow \begin{cases} \mathbf{m}_{k+1}^+ = \mathbf{m}_{k+1}^- + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{n}_{k+1}^-\right) \\ \mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1}\mathbf{P}_{k+1}^{zz}\mathbf{K}_{k+1}^T \end{cases}$$

High-order Kalman filters

- Given the discrete system model:

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k$$

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}, t_{k+1}) + \mathbf{v}_{k+1}$$

- Suppose to have the state $\mathbf{x}_k$, with mean $\mathbf{m}_k^+$ and covariance $\mathbf{P}_k^+$
- General filtering algorithm:
  - Prediction:

classical approach: linearization

$$\mathbf{m}_{k+1}^- = E[\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k \mid \mathbf{z}_k]$$

$$\mathbf{P}_{k+1}^- = E\{[\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k][\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k]^T \mid \mathbf{z}_k\} - (\mathbf{m}_{k+1}^-)(\mathbf{m}_{k+1}^-)^T$$

$$\mathbf{n}_{k+1}^- = E[\mathbf{h}(\mathbf{x}_{k+1}, t_{k+1}) + \mathbf{v}_{k+1} \mid \mathbf{z}_k]$$

  - Update:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{xz}\left(\mathbf{P}_{k+1}^{zz}\right)^{-1} \Rightarrow \begin{cases} \mathbf{m}_{k+1}^+ = \mathbf{m}_{k+1}^- + \mathbf{K}_{k+1}\left(\mathbf{z}_{k+1} - \mathbf{n}_{k+1}^-\right) \\ \mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1}\mathbf{P}_{k+1}^{zz}\mathbf{K}_{k+1}^T \end{cases}$$

High-order Kalman filters

- DA provides high-order polynomials for

$$\mathbf{x}_{k+1} = \mathcal{T}_{\phi_{k,k+1}}(\delta\mathbf{x}_k) + \mathbf{w}_k$$
$$\mathbf{z}_{k+1} = \mathcal{T}_{\mathbf{h}_{k+1}}(\delta\mathbf{x}_{k+1}) + \mathbf{v}_{k+1}$$

- Expectations can then be computed on polynomials. E.g.:

$$\mathbf{m}^-_{k+1} = \sum_{p=1}^{m} \frac{1}{p!} \phi^{\gamma_1 \dots \gamma_p}_{k,k+1} E(\delta\mathbf{x}^{\gamma_1}_k \dots \delta\mathbf{x}^{\gamma_p}_k)$$

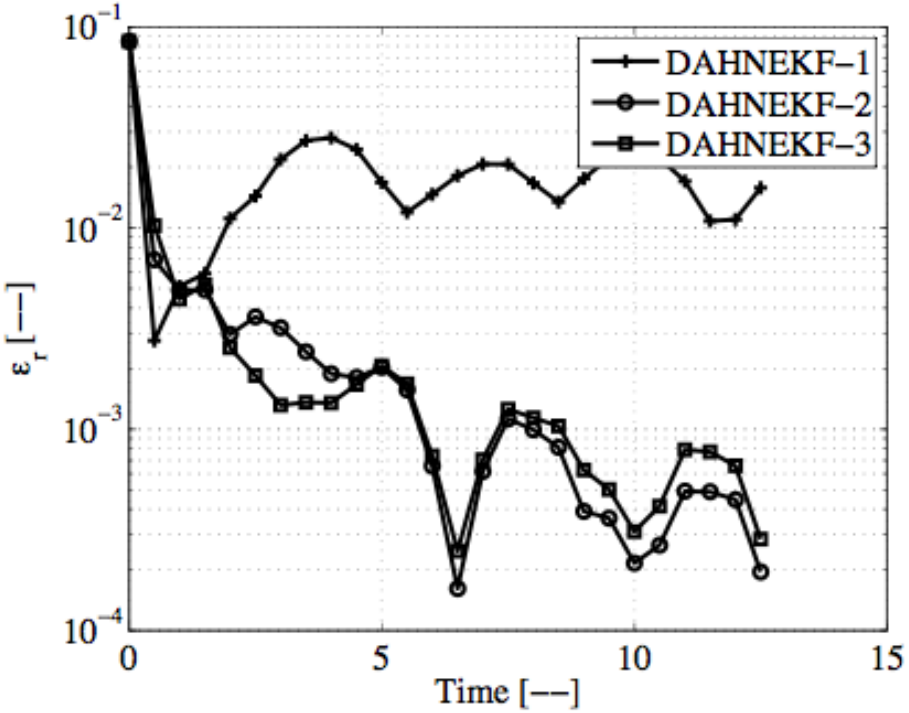
coefficients of the polynomials

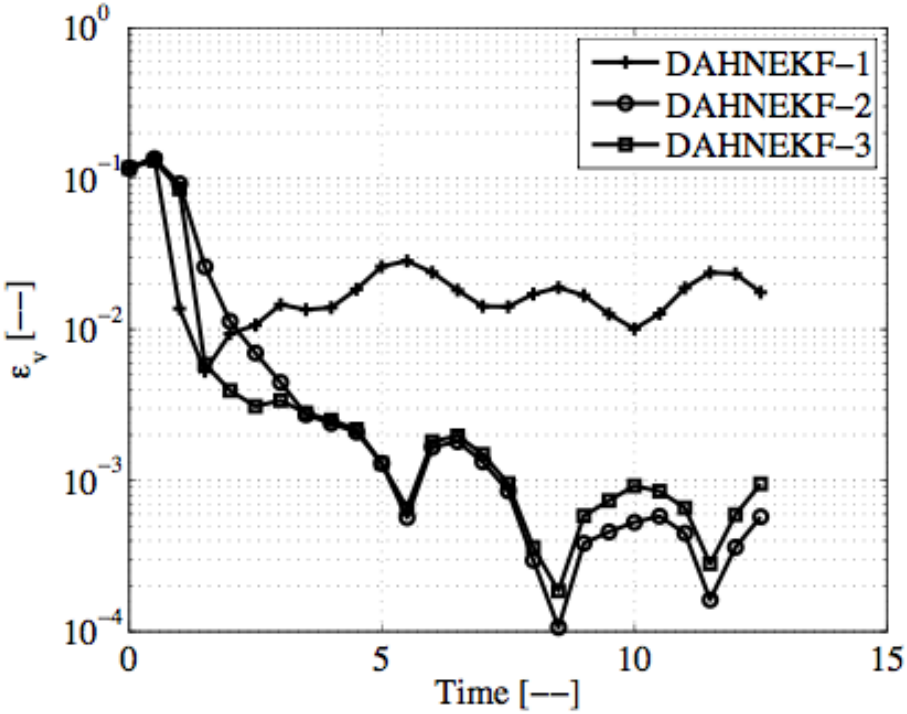
where $E(\delta\mathbf{x}^{\gamma_1}_k \dots \delta\mathbf{x}^{\gamma_p}_k)$ is computed using Isserlis' formula

- Prediction step is fully nonlinear:
  - Faster convergence
  - Possible reduction of measurements acquisition frequency

High-order Kalman filters

- Example: 2-Body Problem
  - State variables: S/C position and velocity
  - Uncertainty on initial conditions (10% off from true initial state) and no process noise
  - Nonlinear measurements with realistic measurement noise (radial position of the S/C wrt the Earth and the LOS direction to Earth)



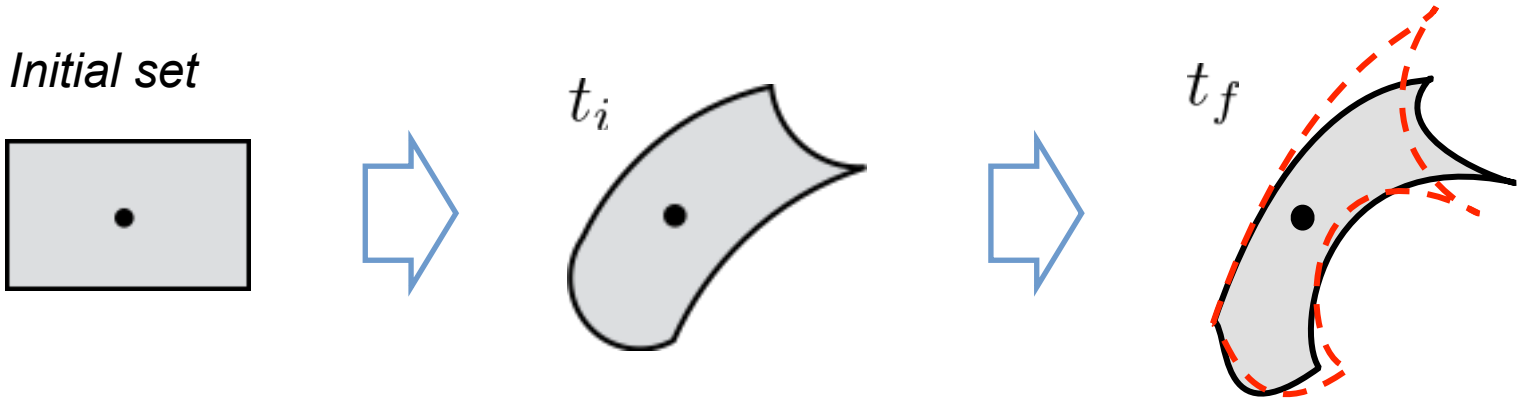


Outline

- Differential Algebra: basic concept
- Uncertainty propagation
  - ODEs flow expansion
  - Linear covariance propagation
  - DA-based Monte Carlo
  - DA-based statistical moments
  - Polynomial bounder
- Applications
  - SST: Impact probability computation
  - High-order filters
- Advanced methods
  - Automatic domain splitting

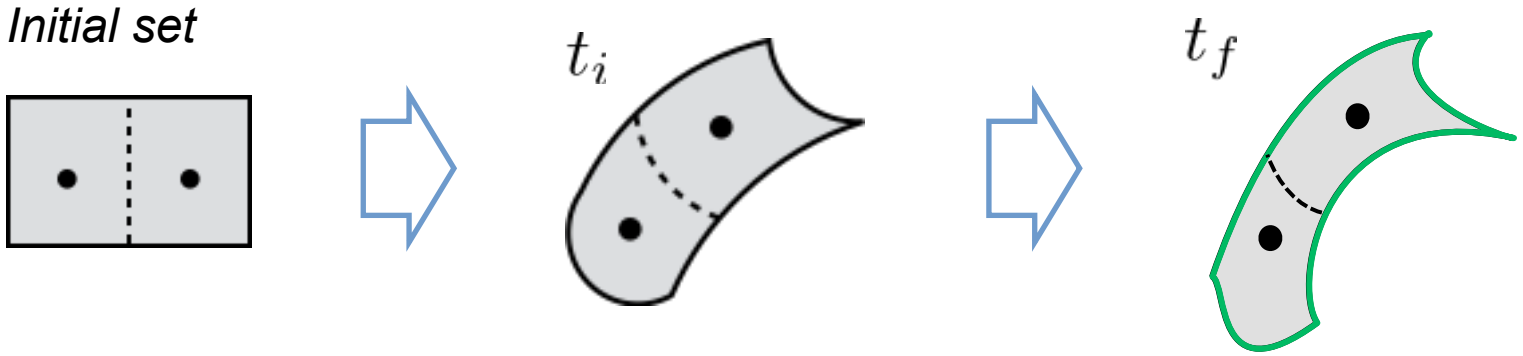
Automatic domain splitting

- In challenging cases, a high-order Taylor expansion is not sufficient to accurately map uncertainties

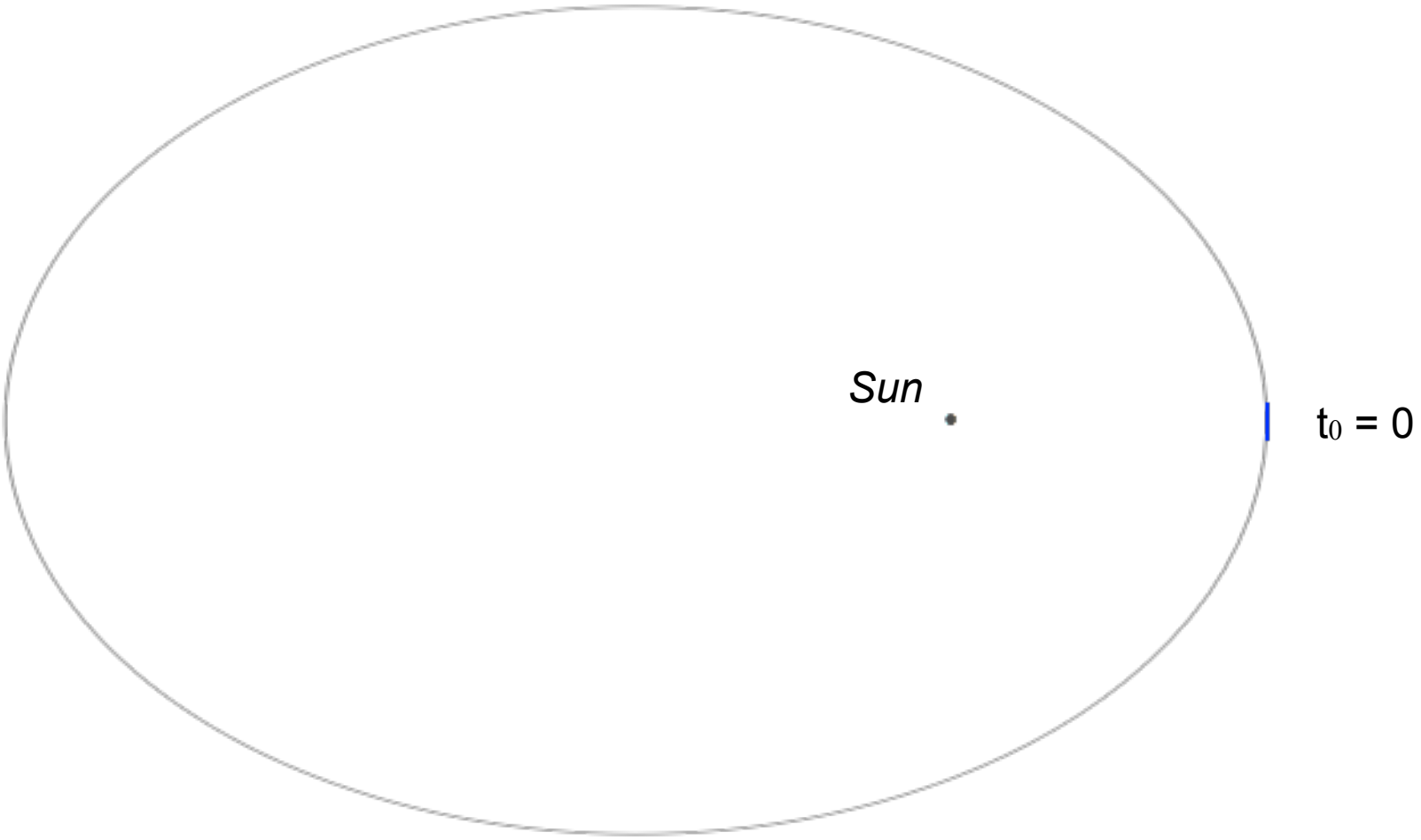


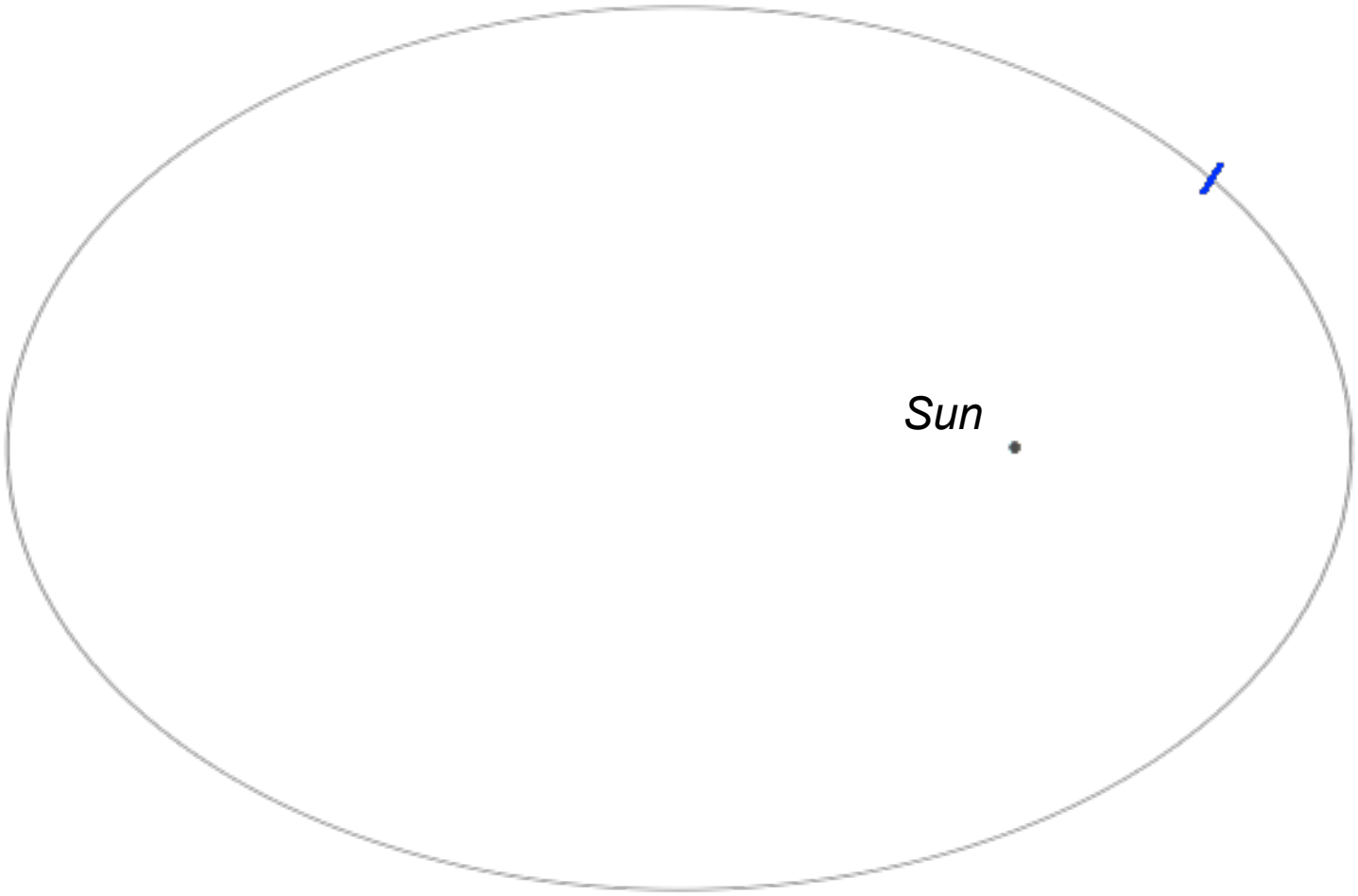
Automatic domain splitting

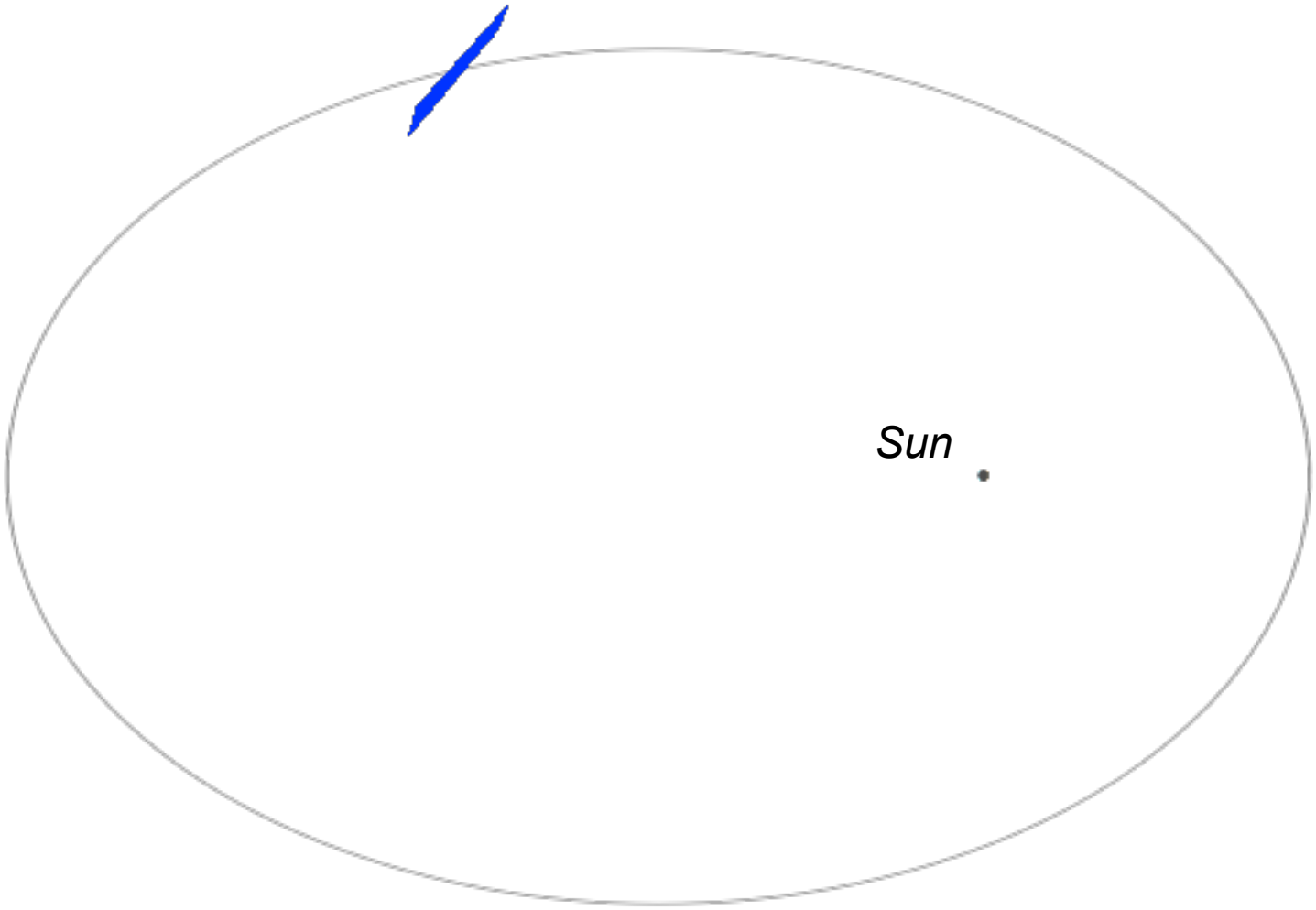
- In challenging cases, a high-order Taylor expansion is not sufficient to accurately map uncertainties

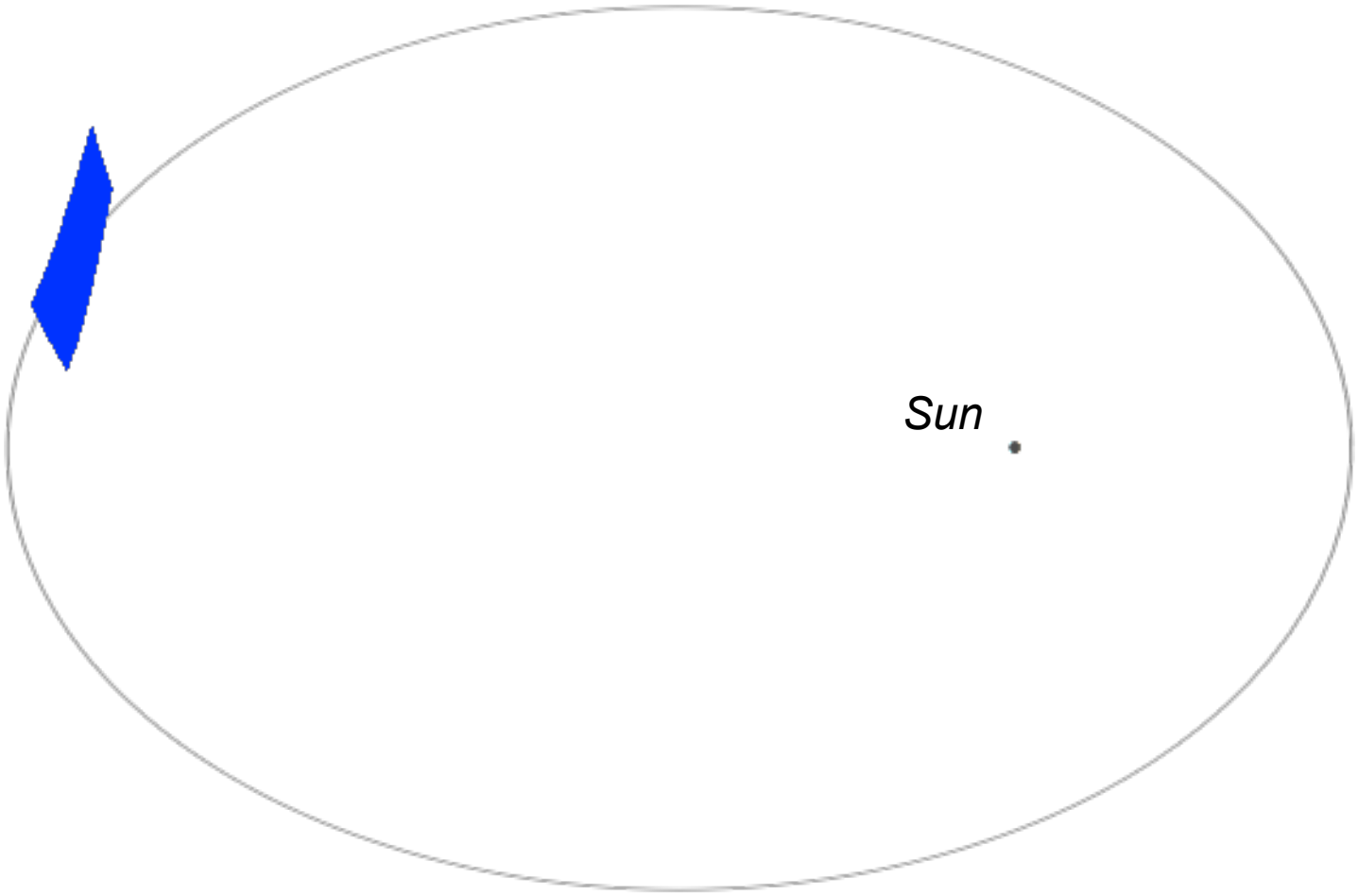


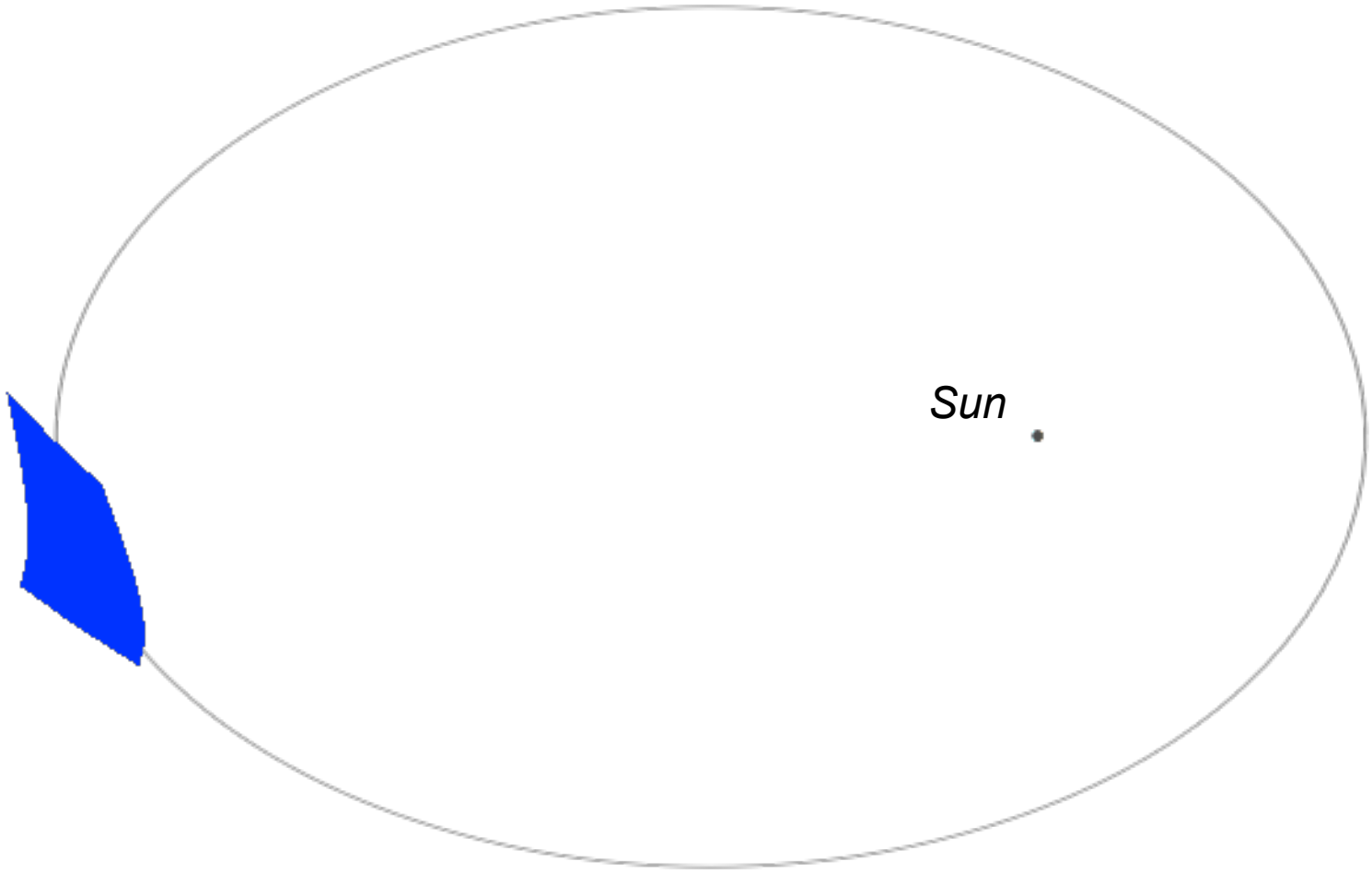
- An algorithm can be implemented to automatically split the initial domain to guarantee the desired accuracy
- Very useful for: large uncertainty sets, long-term propagations, highly nonlinear dynamics

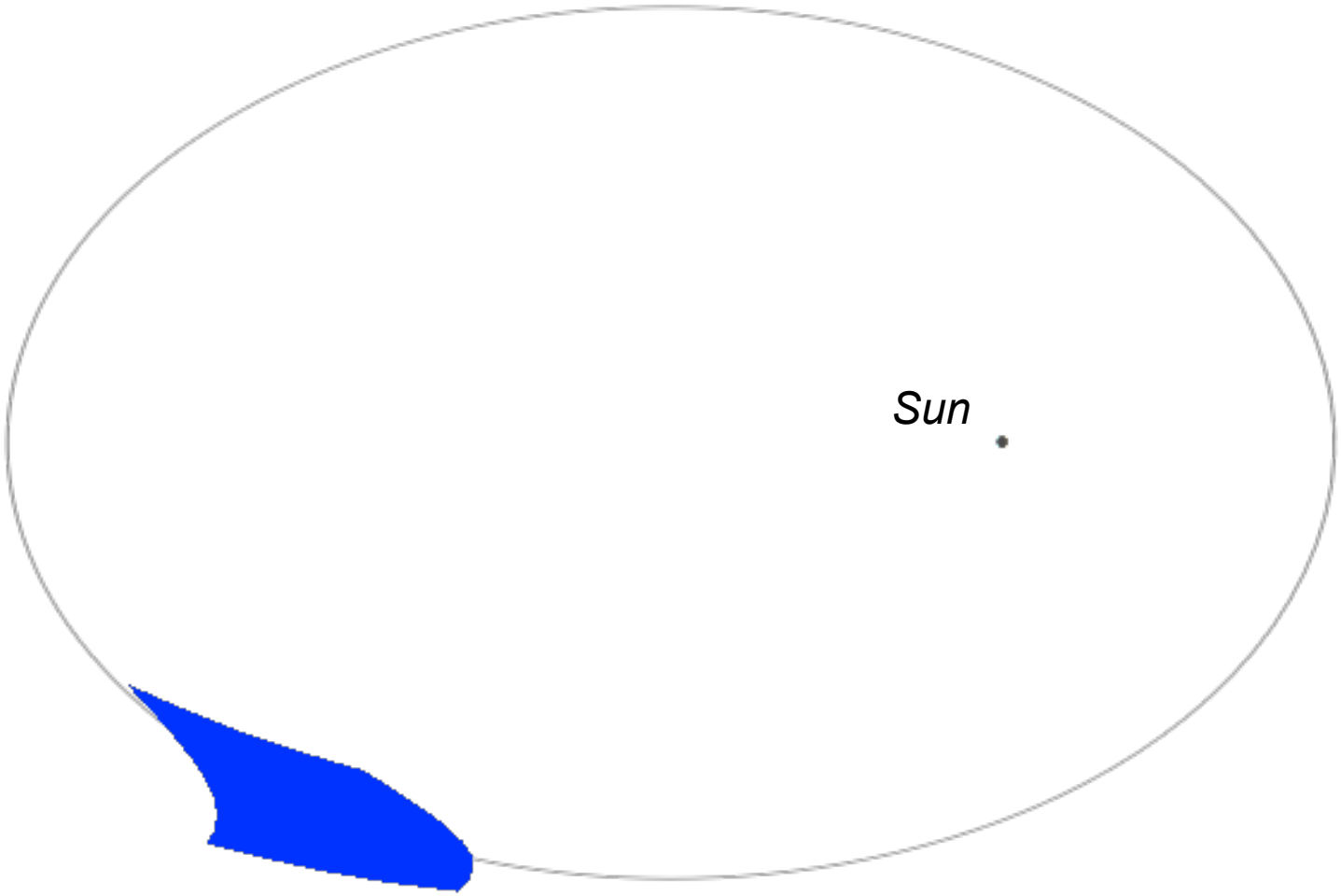

Sun
$t_0 = 0$

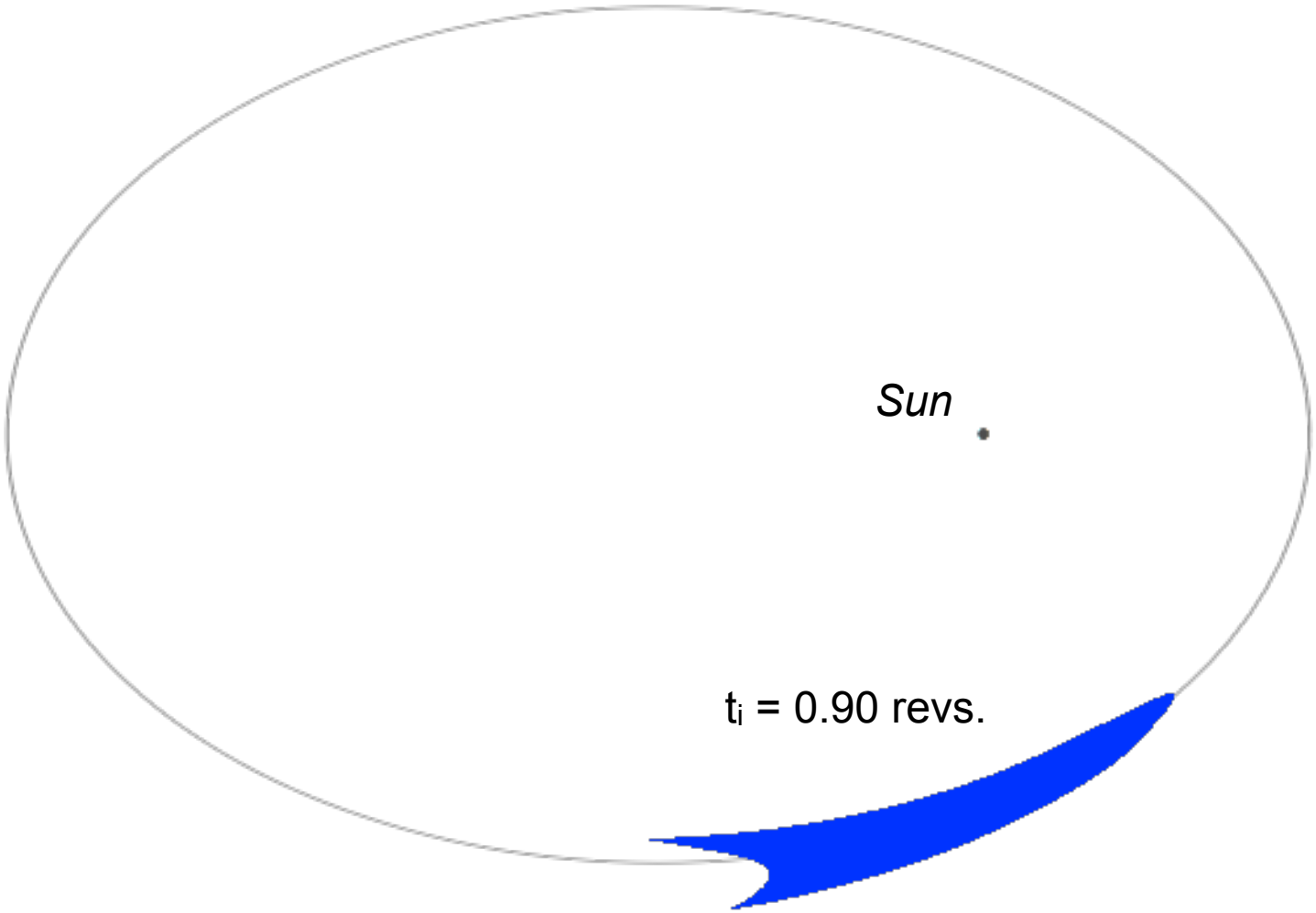

Sun

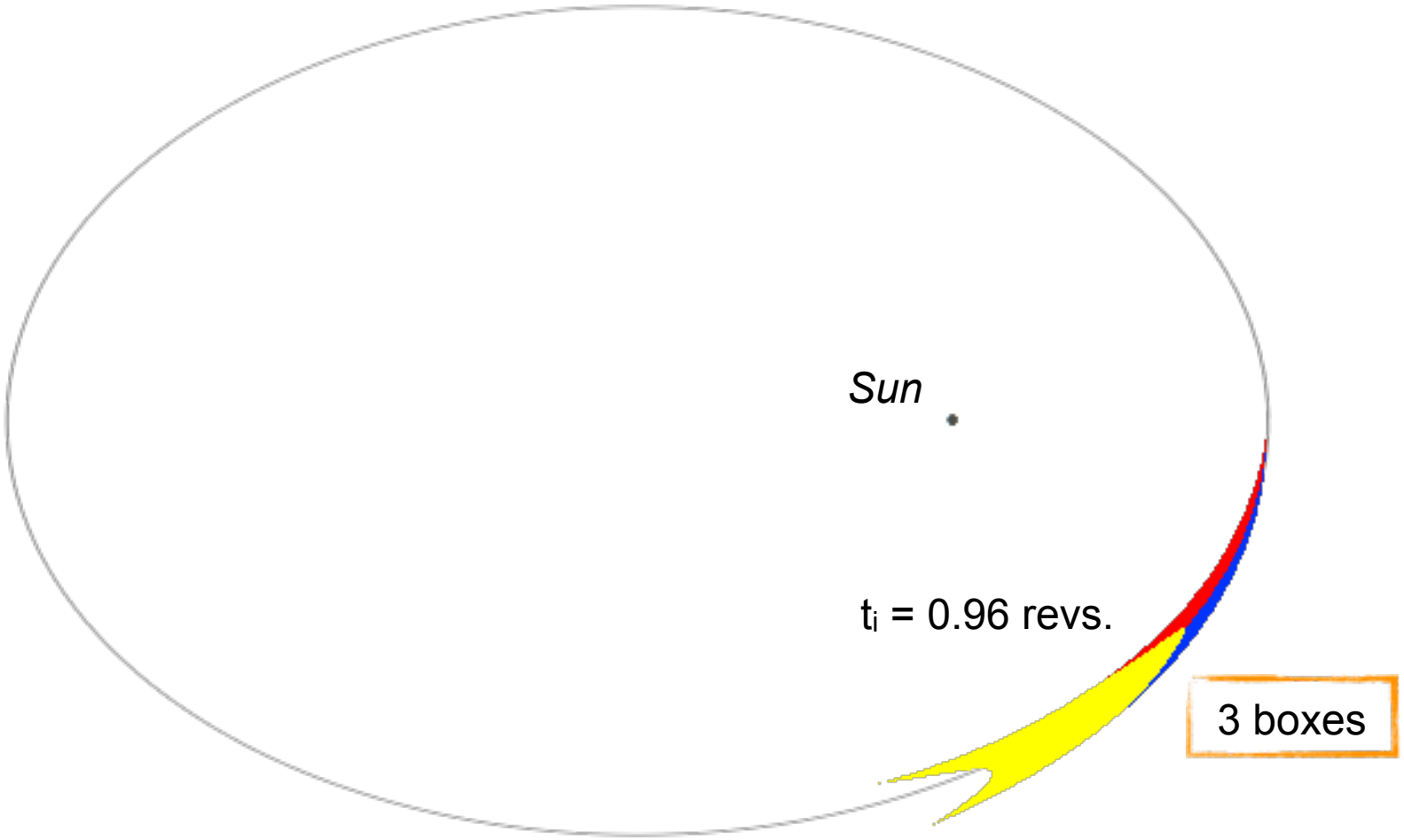

Sun

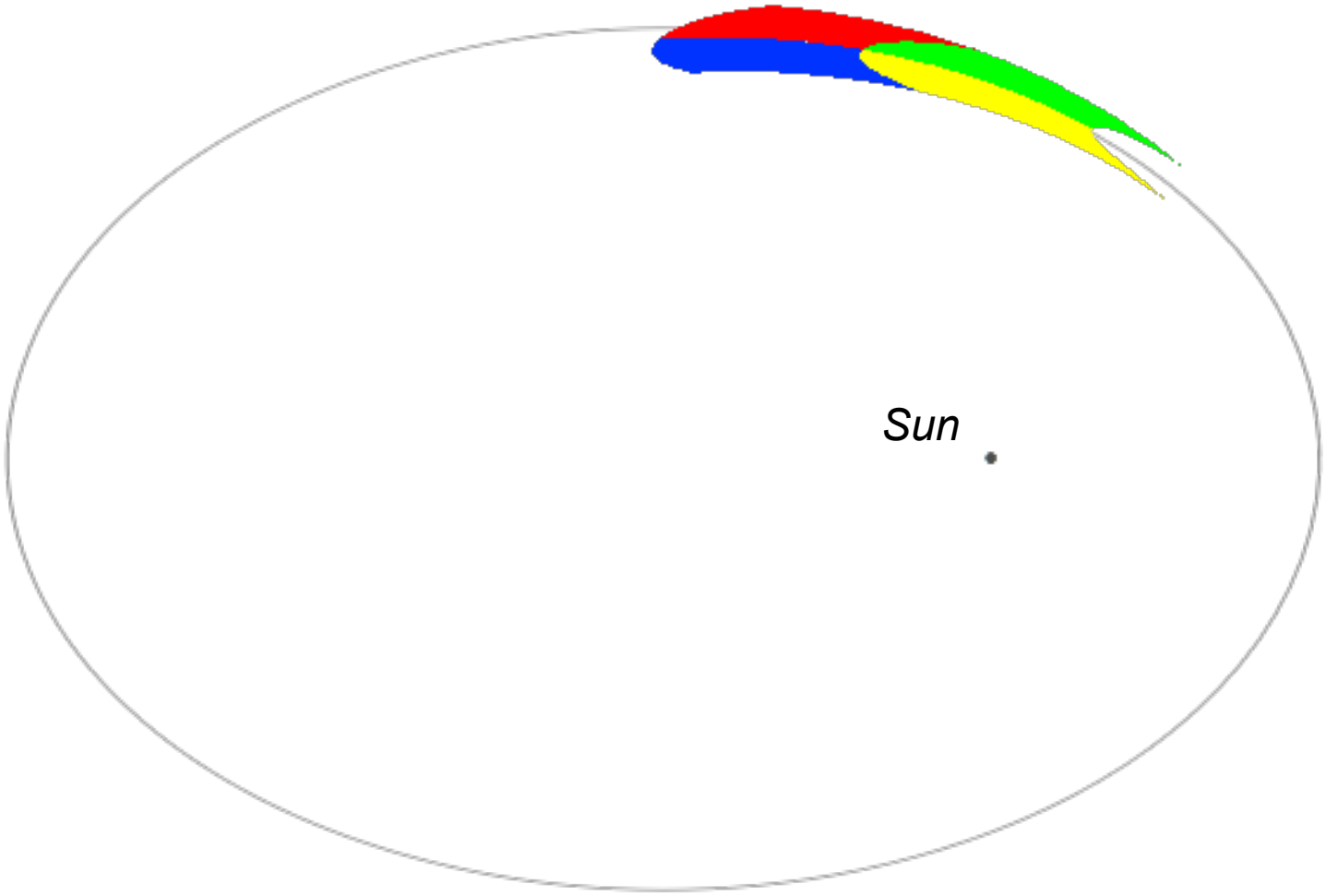

Sun

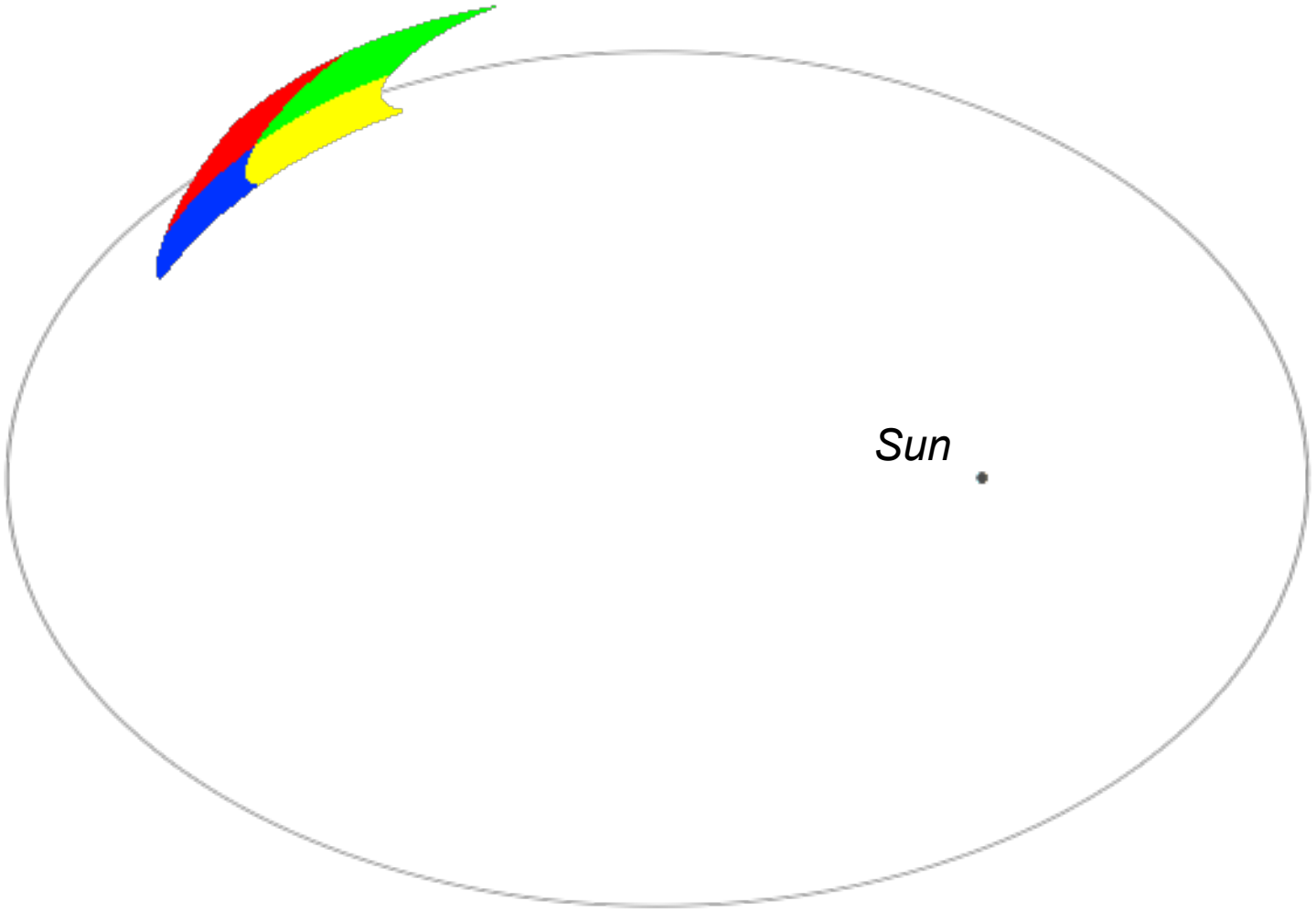

Sun

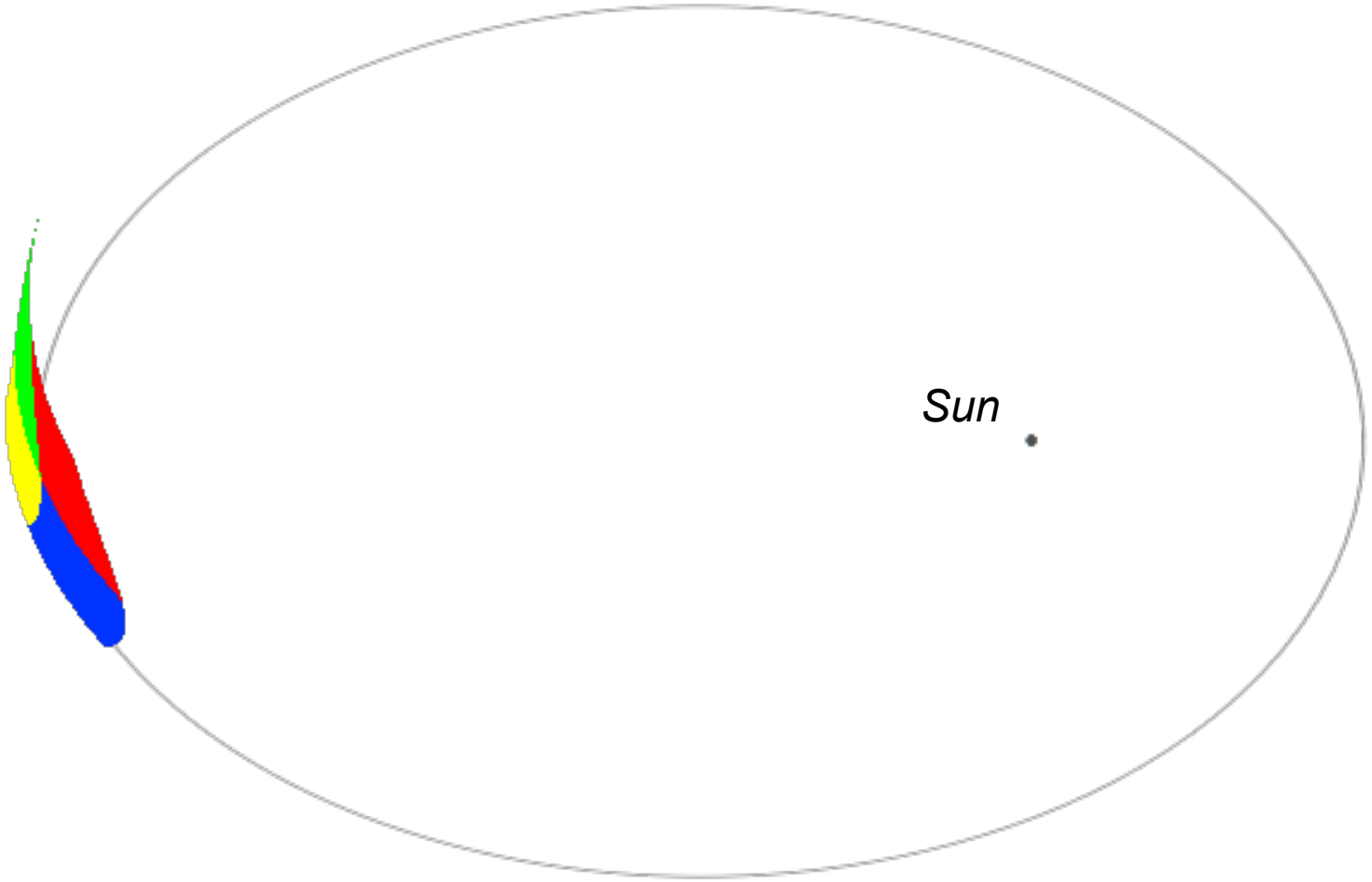

Sun

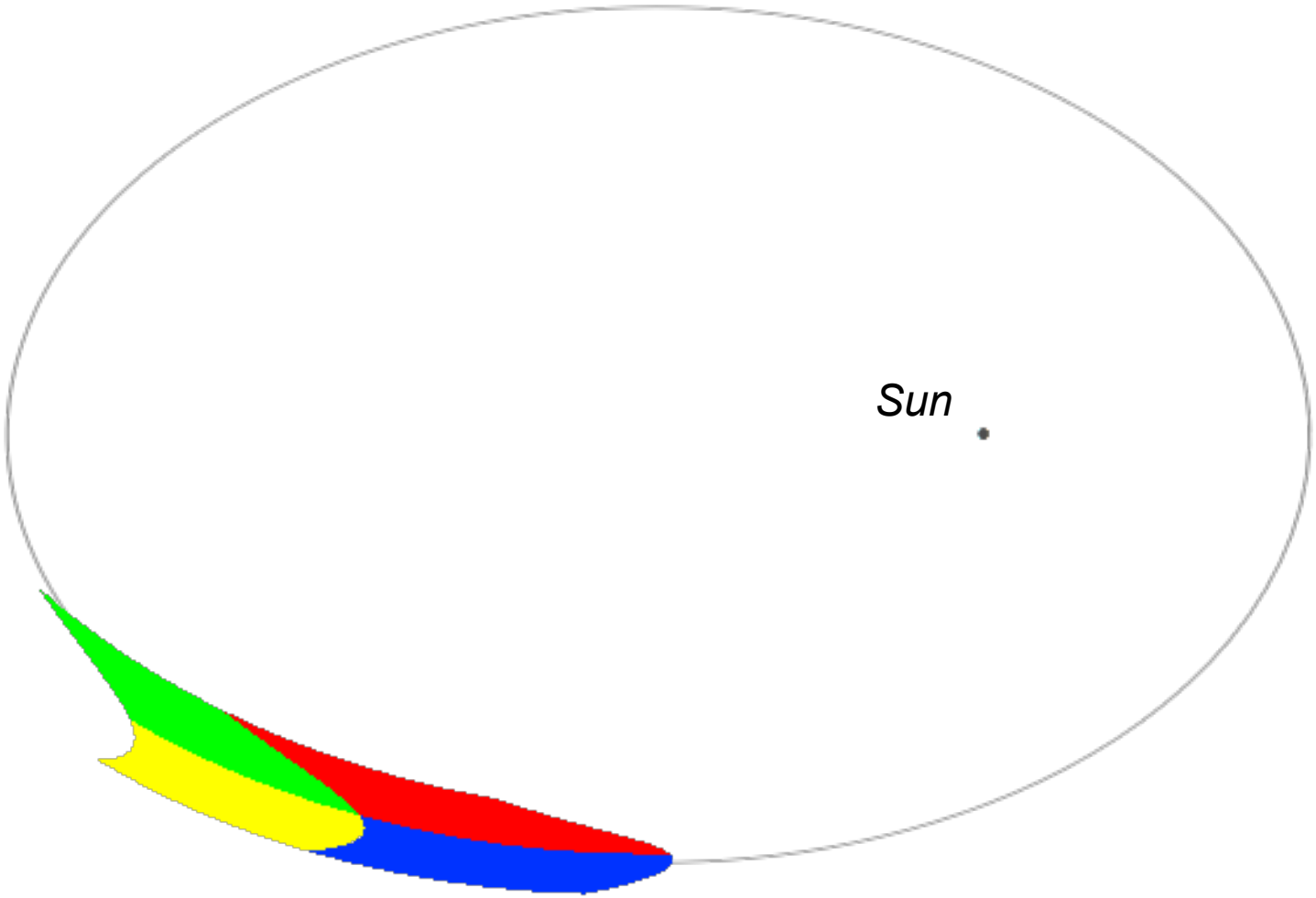

Sun
t_i = 0.90 revs.

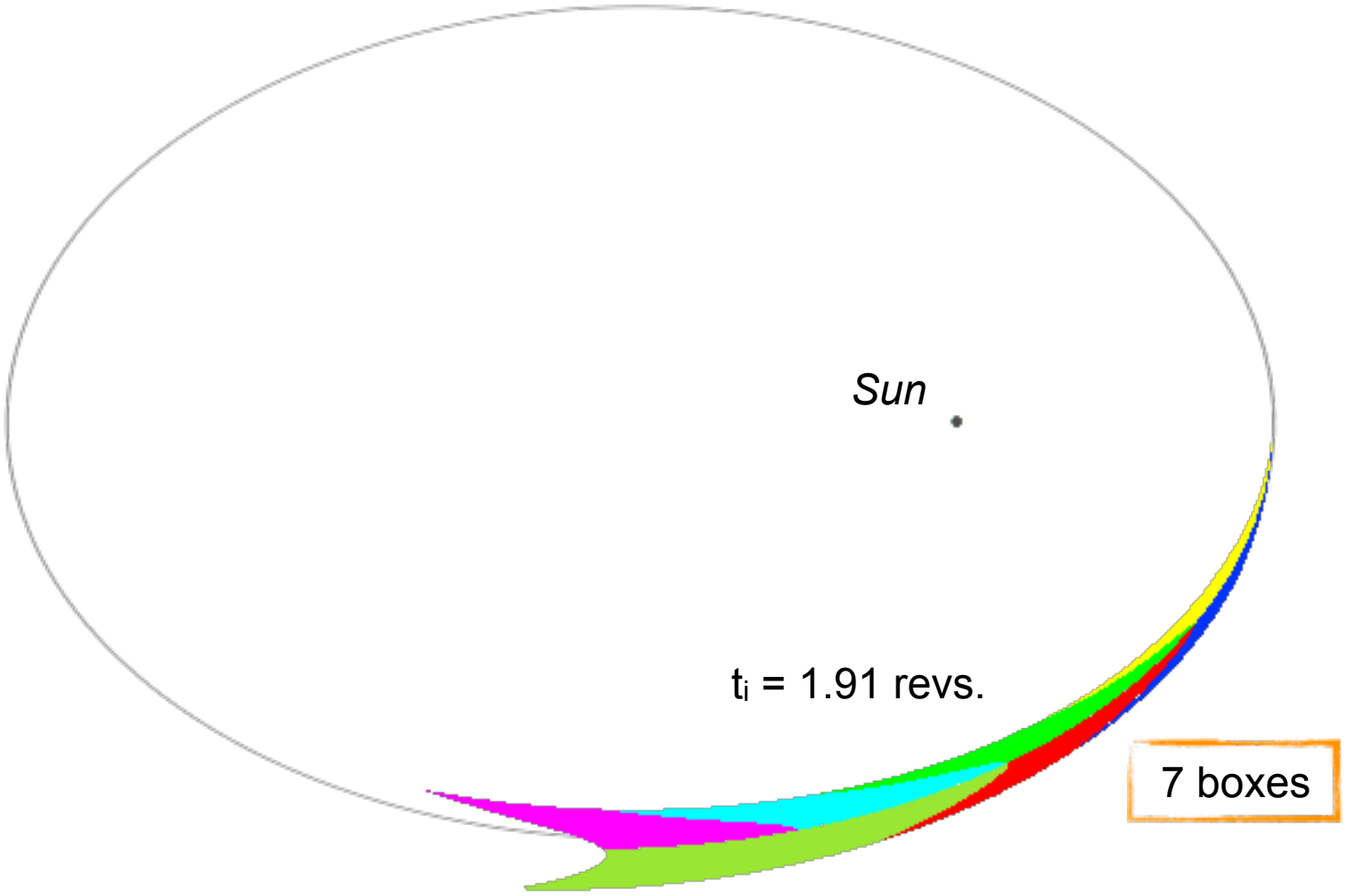

Sun
t_i = 0.96 revs.
3 boxes

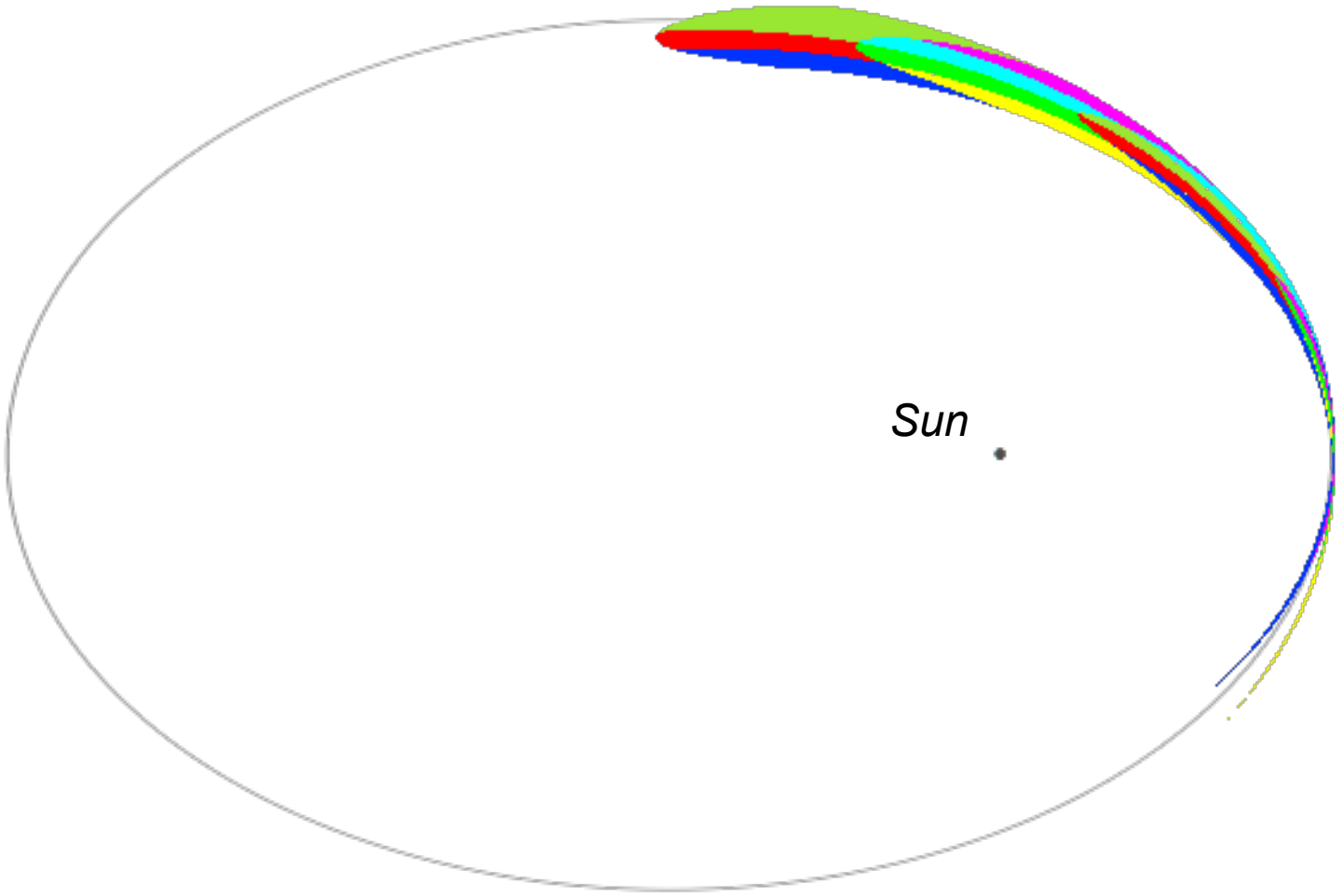

Sun

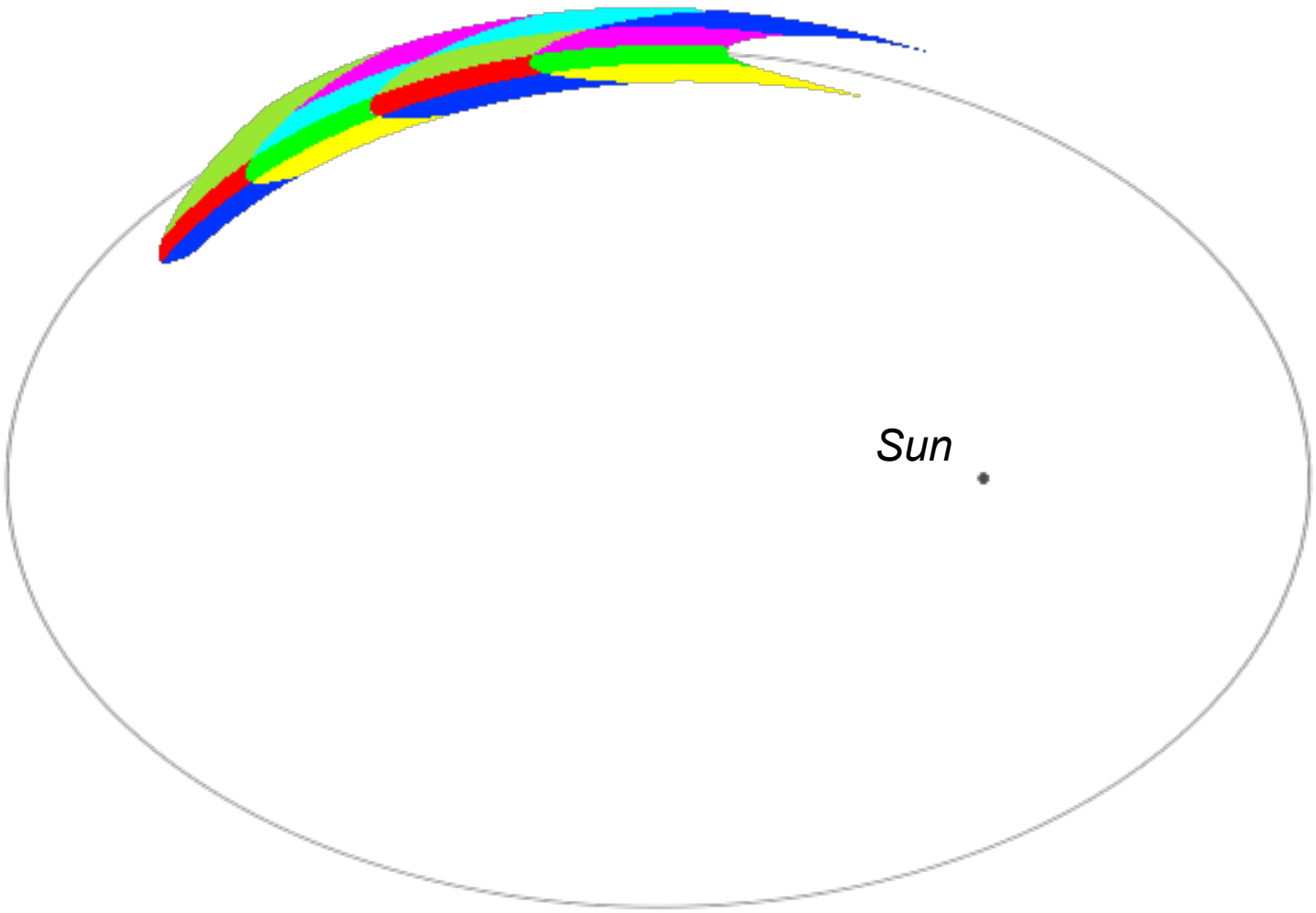

Sun

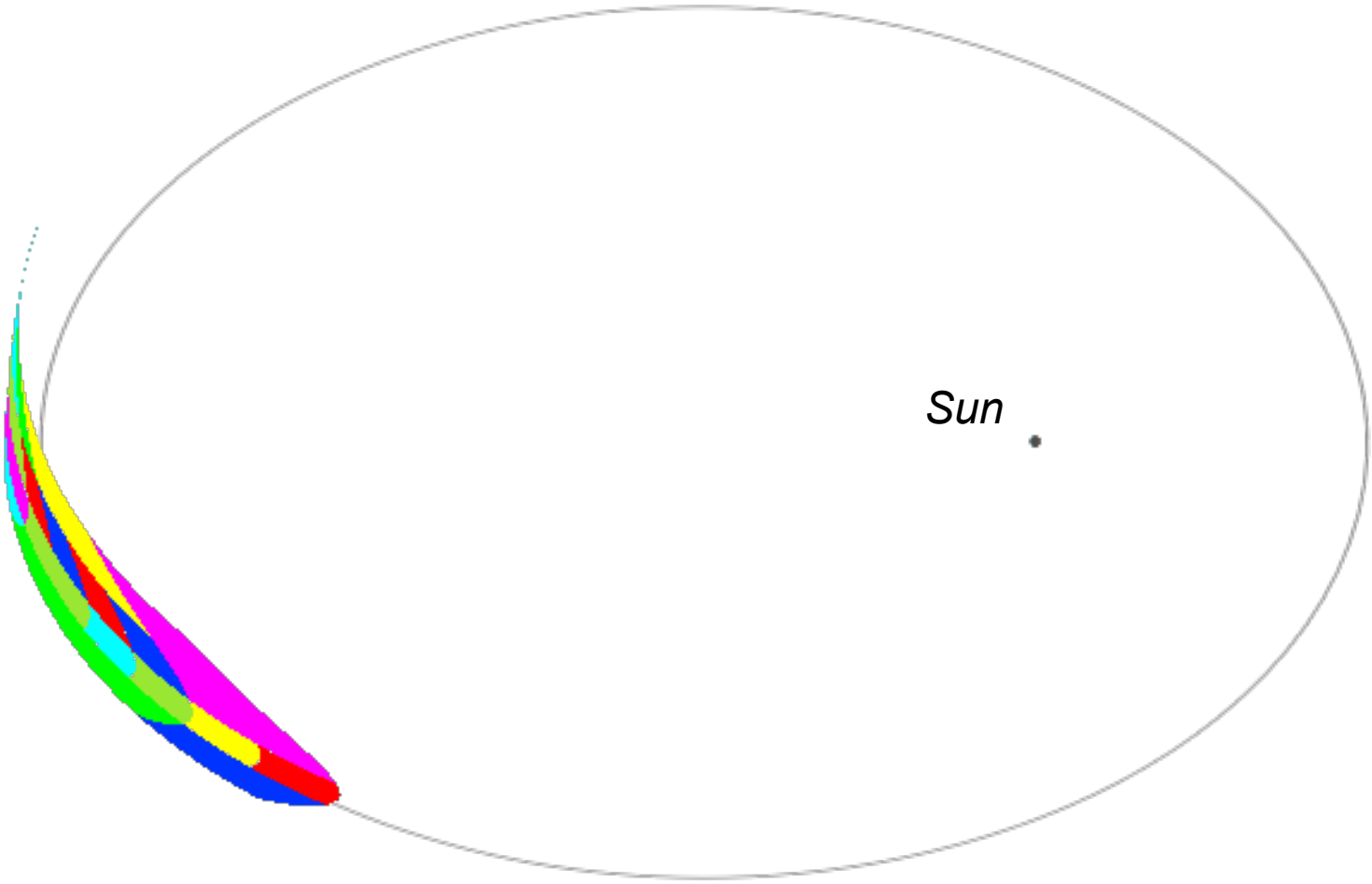

Sun

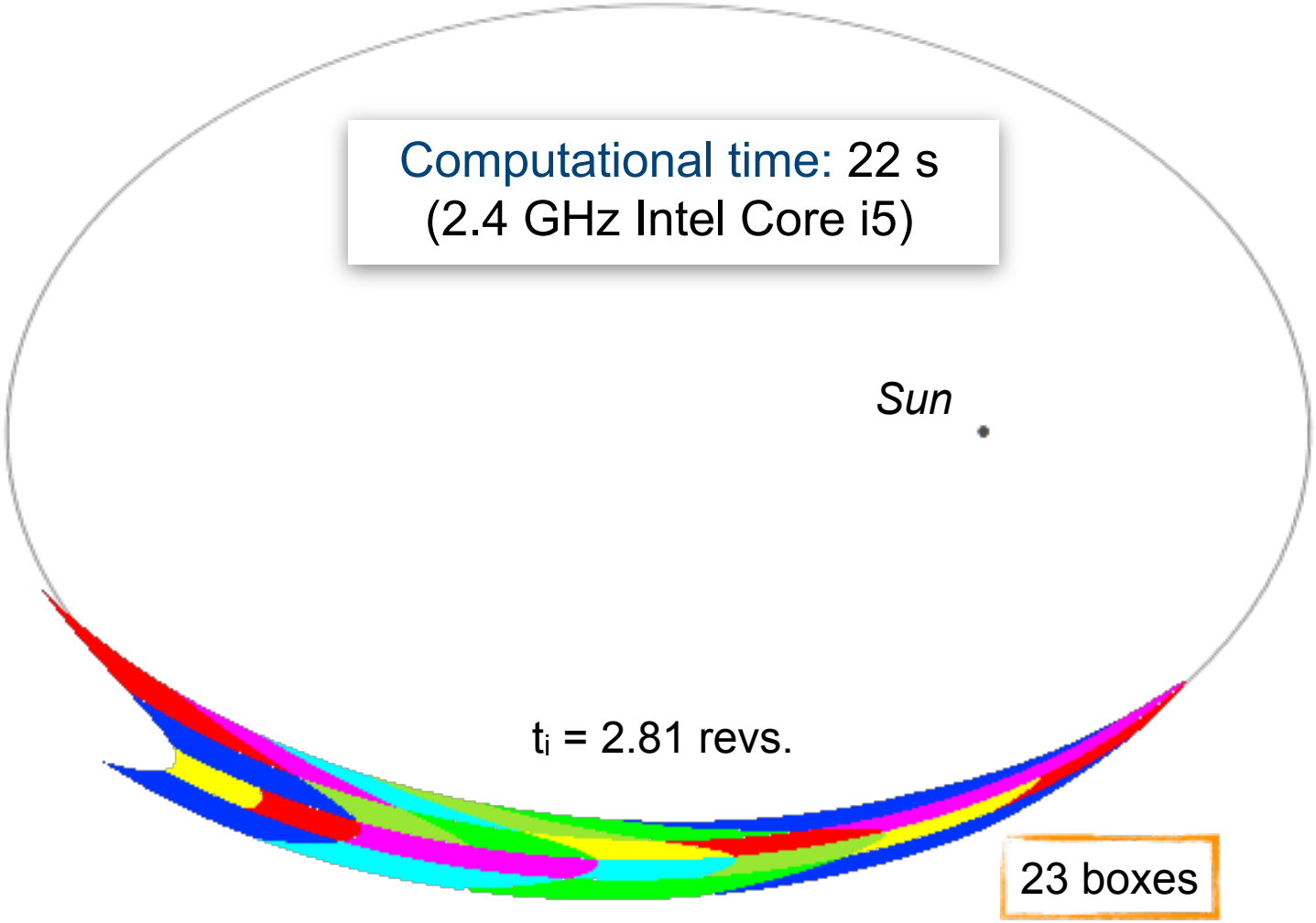

Sun


Sun
t_i = 1.91 revs.
7 boxes


Sun

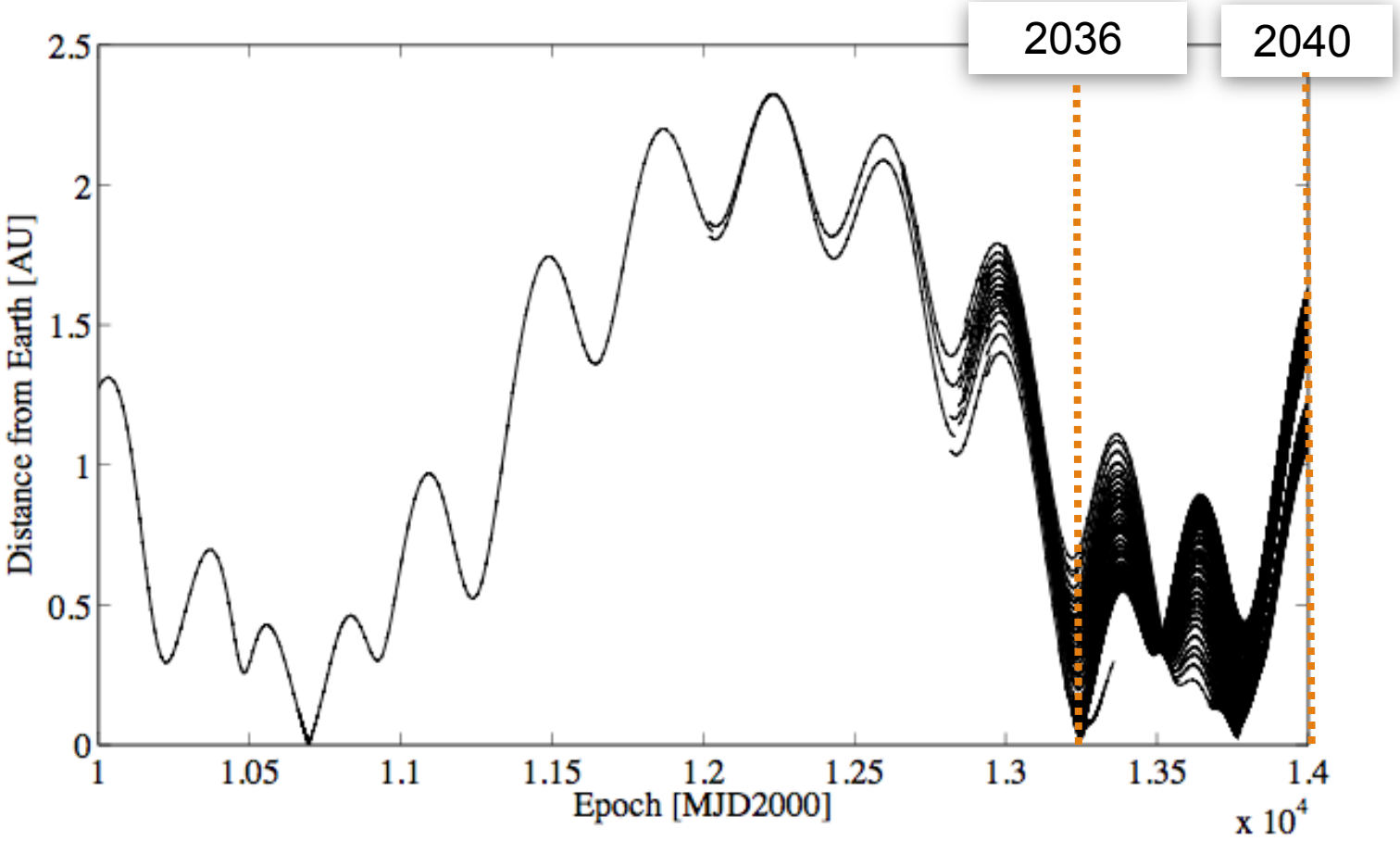

Sun


Sun


Computational time: 22 s
(2.4 GHz Intel Core i5)
Sun
t_i = 2.81 revs.
23 boxes

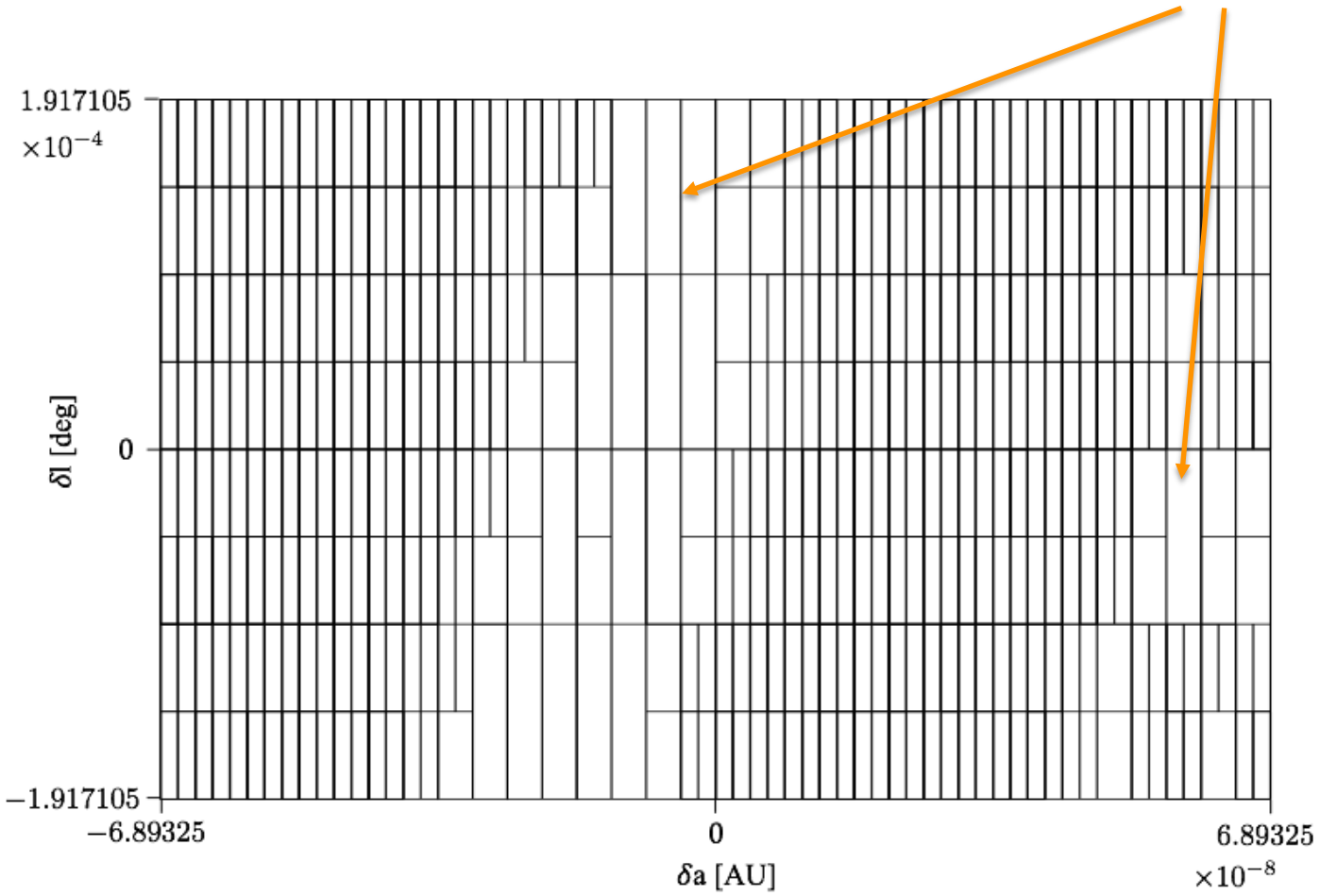
Automatic domain splitting

- Long term propagation of Apophis motion (NEODys in Sept. 2009)



Automatic domain splitting

- Long term propagation of Apophis motion



Conclusions

- Differential Algebra can
  - improve efficiency of standard nonlinear methods
  - improve accuracy of standard linear methods
- Note: method based on Taylor approximations ⇒ Size of uncertainty set and order shall guarantee sufficient accuracy
- Polynomials can be manipulated to impose constraints on both initial and final state (e.g. see SST application)
- 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

References

2015

- High order transfer maps for perturbed Keplerian motion. Celestial Mechanics and Dynamical Astronomy
- Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting. Celestial Mechanics and Dynamical Astronomy

2014

- A high order method for orbital conjunctions analysis: Monte Carlo collision probability computation. Advances in Space Research
- Efficient evaluation of vapour–liquid equilibria from multi-parameter thermodynamic models using differential algebra. Journal of Computational and Applied Mathematics
- A high order method for orbital conjunctions analysis: sensitivity to initial uncertainties. Advances in Space Research
- High order optimal feedback control of space trajectories with bounded control. Acta Astronautica
- Nonlinear filtering methods for spacecraft navigation based on differential algebra. Acta Astronautica
- High order optimal control of space trajectories with uncertain boundary conditions. Acta Astronautica

2013

- Nonlinear mapping of uncertainties in celestial mechanics. Journal of Guidance, Control, and Dynamics
- Rigorous computation of orbital conjunctions. Advances in Space Research

References

2012

- High-order expansion of the solution of preliminary orbit determination problem. Celestial Mechanics and Dynamical Astronomy
- Rigorous global optimization of impulsive planet-to-planet transfers in the patched-conics approximation. Engineering Optimization

2010

- Asteroid close encounters characterization using differential algebra: the case of Apophis. Celestial Mechanics and Dynamical Astronomy
- Computing the critical points of the distance function between two Keplerian orbits via rigorous global optimization. Celestial Mechanics and Dynamical Astronomy
- Gravity assist space pruning based on differential algebra. Celestial Mechanics and Dynamical Astronomy

2008

- Application of high order expansions of two-point boundary value problems to astrodynamics. Celestial Mechanics and Dynamical Astronomy
- High-order robust guidance of interplanetary trajectories based on differential algebra. Journal of Aerospace Engineering, Sciences and Applications

Uncertainty Propagation Using Differential Algebra

Armellin Roberto

with contributions by:

Pierluigi Di Lizia, Alexander Wittig, Alessandro Morselli, Monica Valli,

Workshop on
Nonlinear Uncertainty Propagation using Differential Algebra
22nd September 2015

DINAMICA srl
Registered Office:
Piazza della Repubblica, 10 - 20121 - Milano (Italy)
Operational Headquarters:
Via Morghen, 13 - 20156 - Milano (Italy)
Phone +39 02 8342 2930
Fax +39 02 3206 6679
e-mail: dinamica@dinamicatech.com
website: www.dinamicatech.com