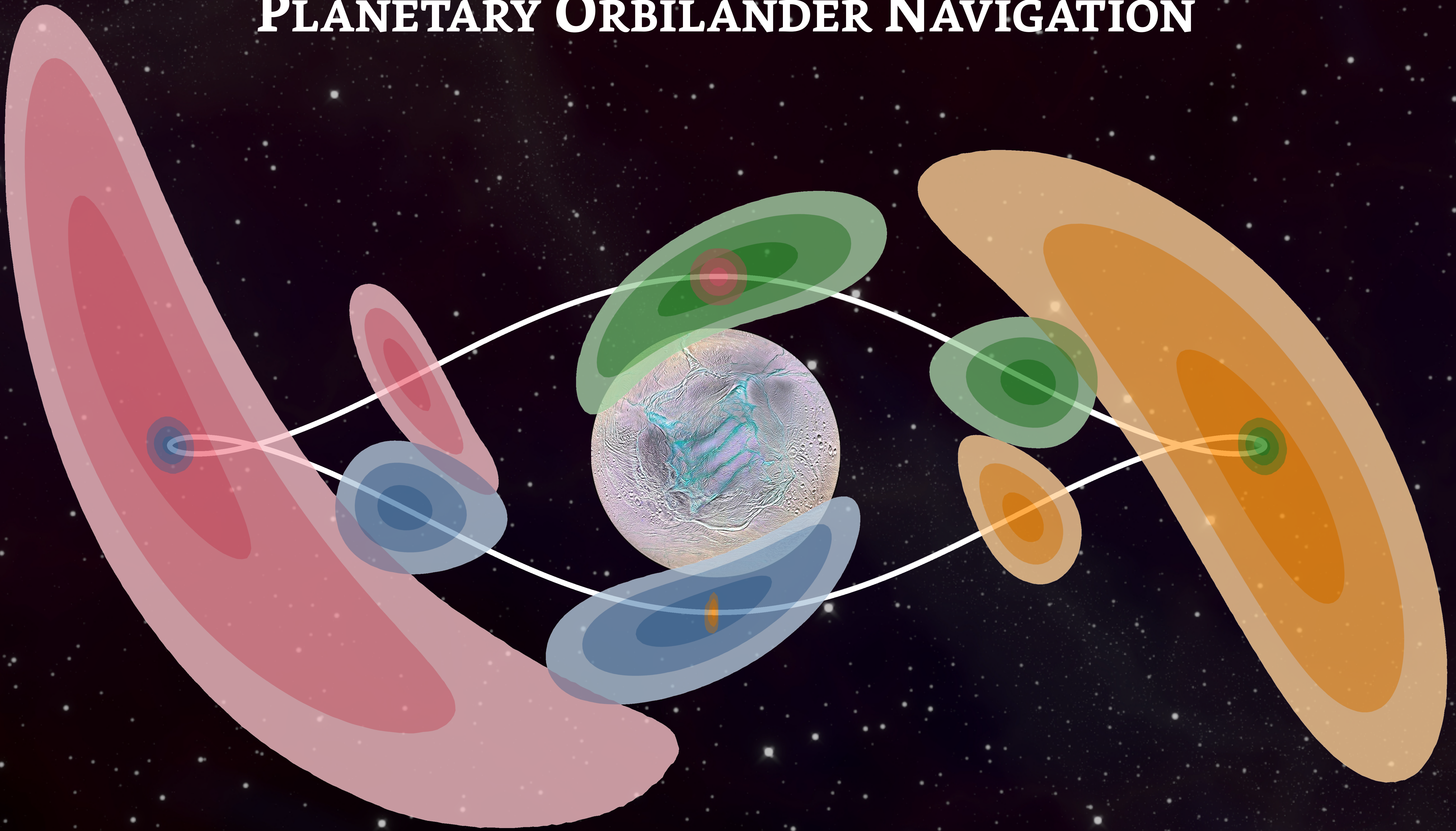
Non-Gaussian Recursive Bayesian Filtering for Outer Planetary Orbilander Navigation



Benjamin L. Hanson
Ph.D. Student, Jacobs School of Engineering
Department of Mechanical and Aerospace Engineering
UC San Diego, La Jolla, CA

Dr. Aaron J. Rosengren
Assistant Professor, Jacobs School of Engineering
Department of Mechanical and Aerospace Engineering
UC San Diego, La Jolla, CA

Dr. Thomas R. Bewley
Professor, Jacobs School of Engineering
Department of Mechanical and Aerospace Engineering
UC San Diego, La Jolla, CA

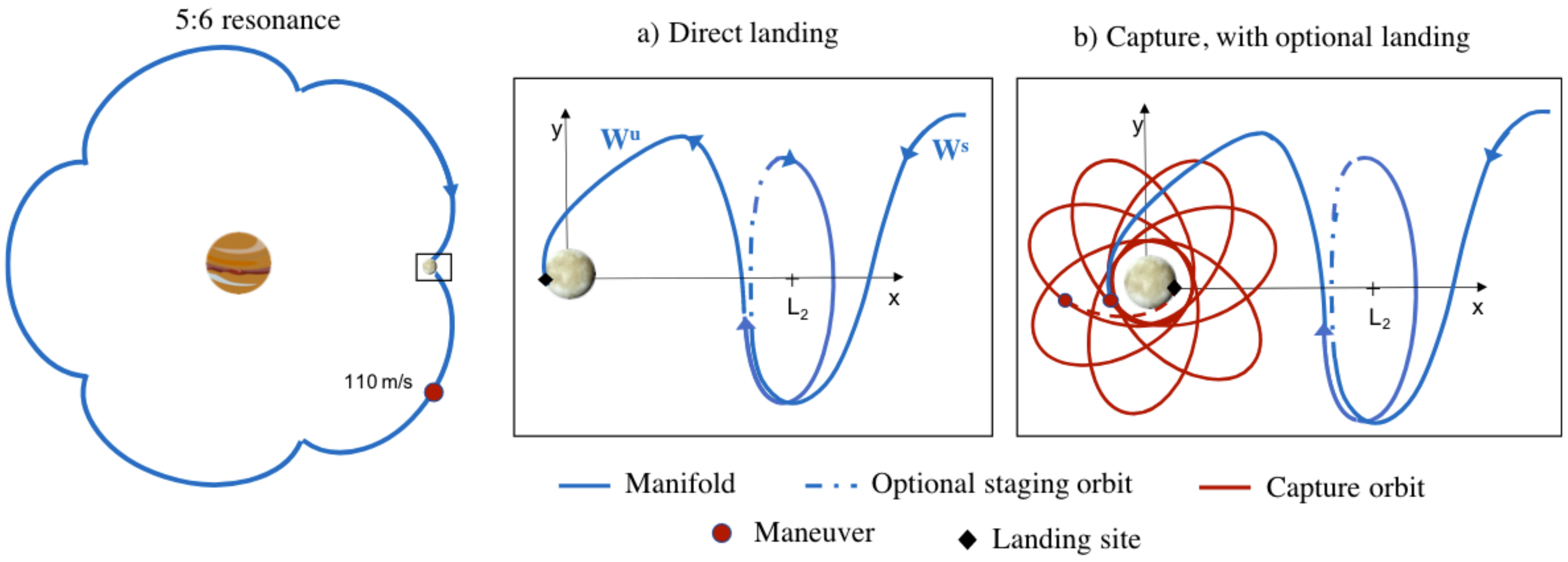
Dr. Todd A. Ely
Principal Navigation Engineer
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA

Case Study: Low-Energy Trajectories for Europa Lander

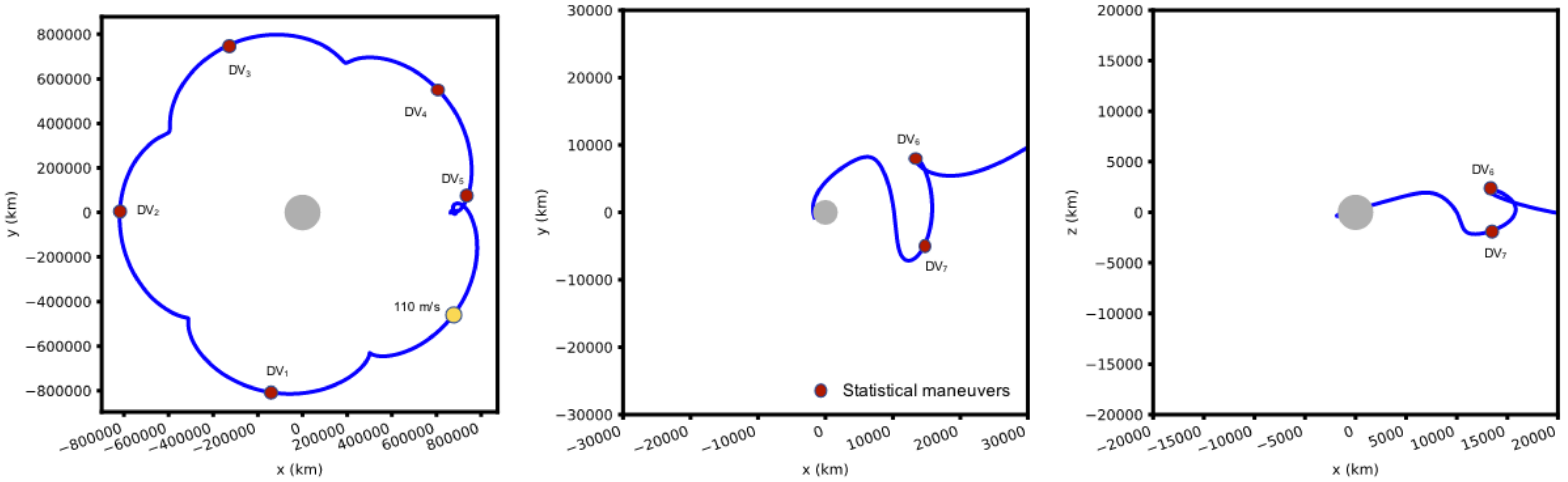
Time validity of the Gaussian assumption of uncertainty

- A theoretically ΔV-free, ballistic capture of a Europa lander realistically requires **statistical maneuvers** to maintain Gaussian error in position and velocity for linearized navigation techniques

Proposed ΔV-free ballistic capture



Actual trajectory with statistical maneuvers ΔV_i



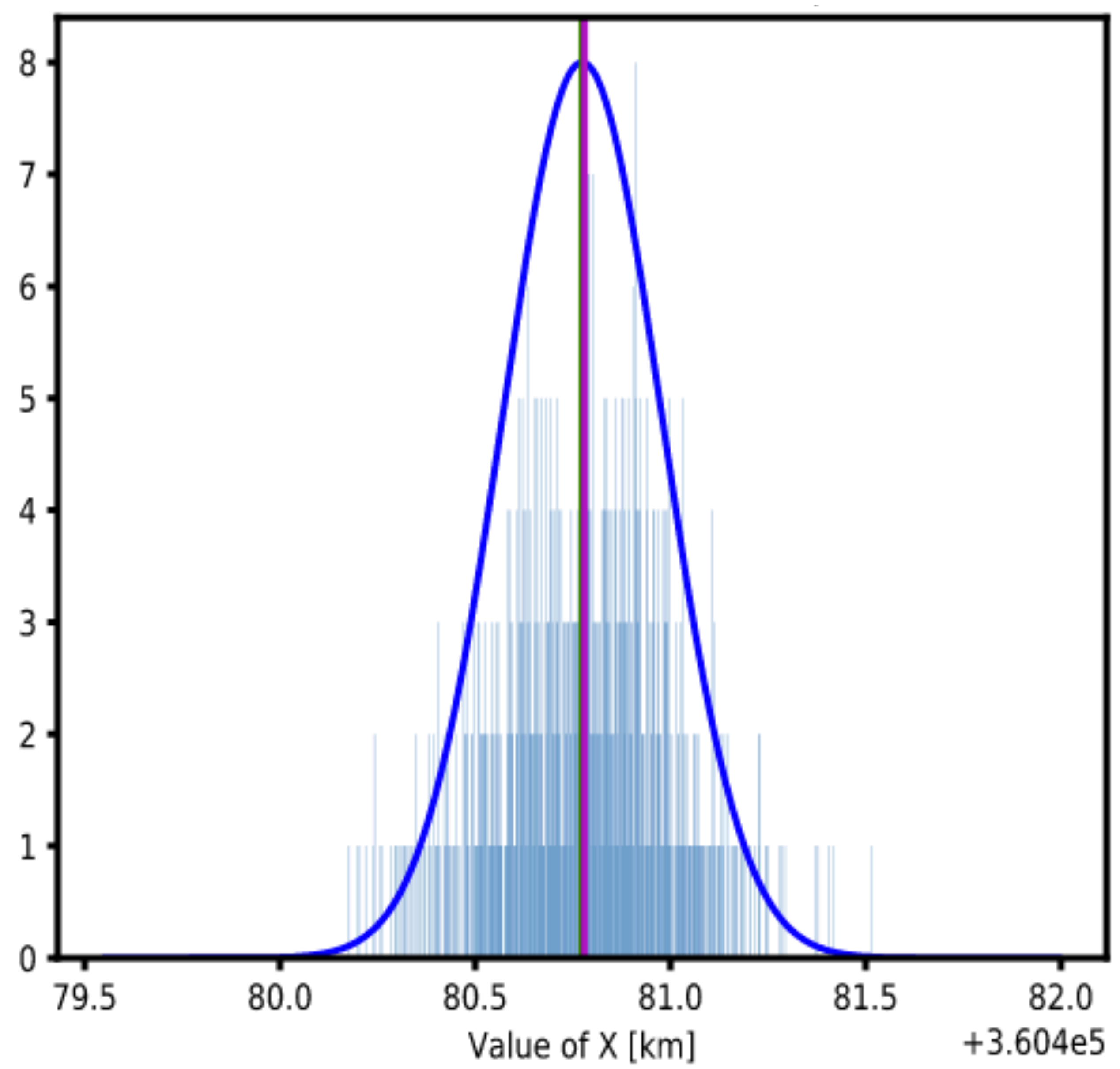
McElrath, Tim, et al. "Navigating low-energy trajectories to land on the surface of Europa." NTRS. (2021)

Case Study: Low-Energy Trajectories for Europa Lander

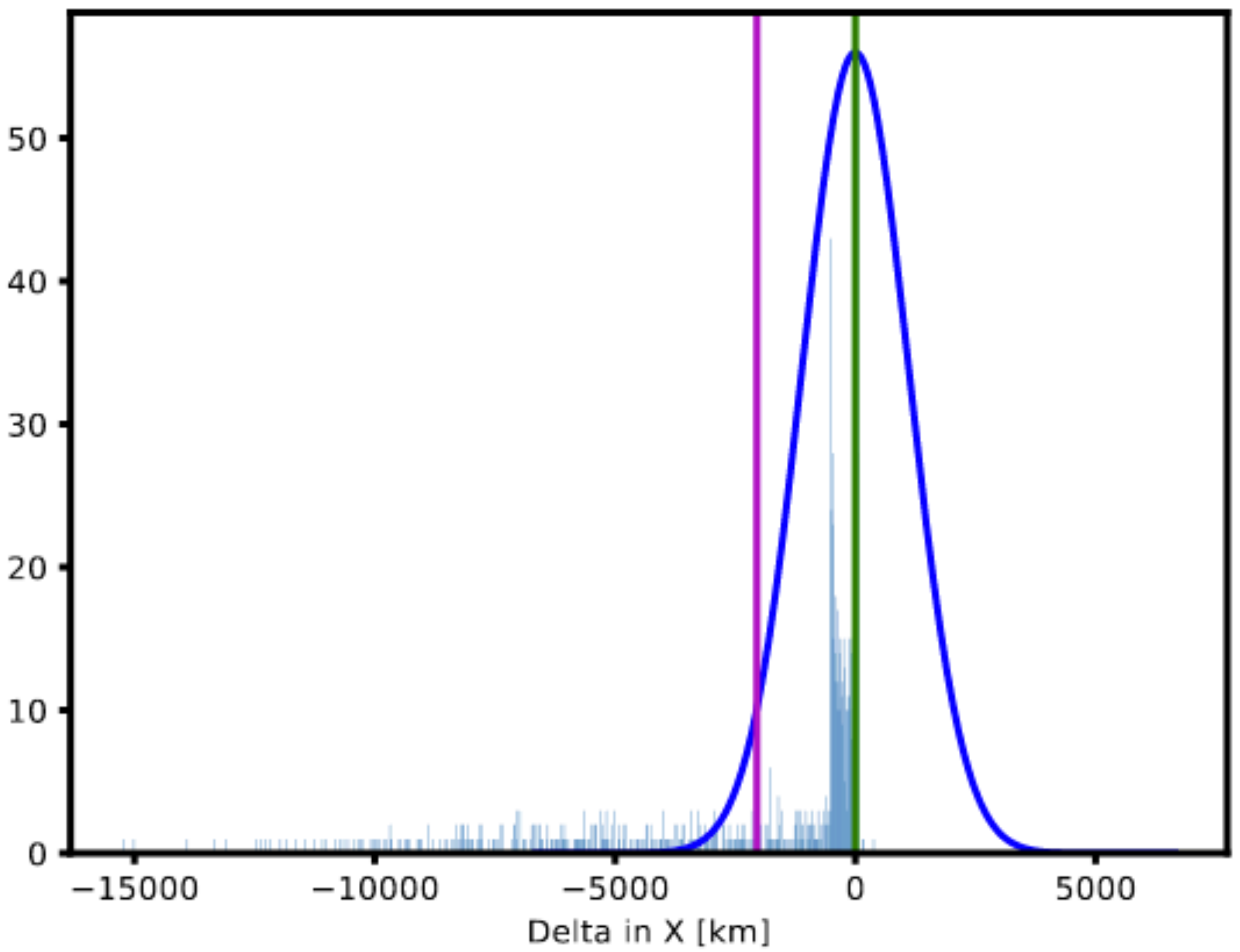
Time validity of the Gaussian assumption of uncertainty

- A theoretically ΔV-free, ballistic capture of a Europa lander realistically requires **statistical maneuvers** to maintain Gaussian error in position and velocity for linearized navigation techniques

Proposed ΔV-free ballistic capture



Initial Gaussian uncertainty at leveraging maneuver

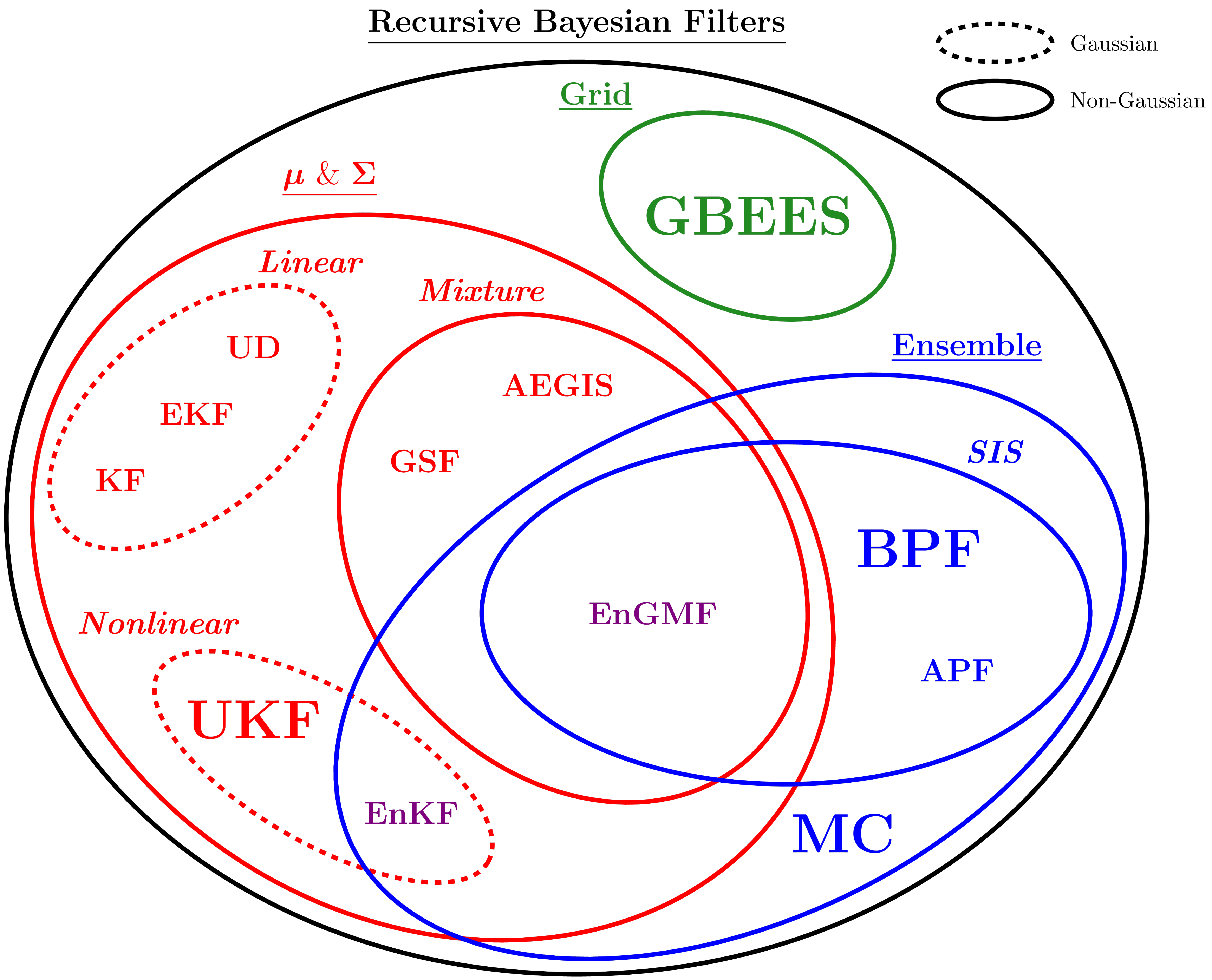


Final non-Gaussian uncertainty at Europa arrival

Key question: What are the temporal limits of Gaussian filters in the Jovian regime (or elsewhere), and when might it be necessary to implement non-Gaussian filters?

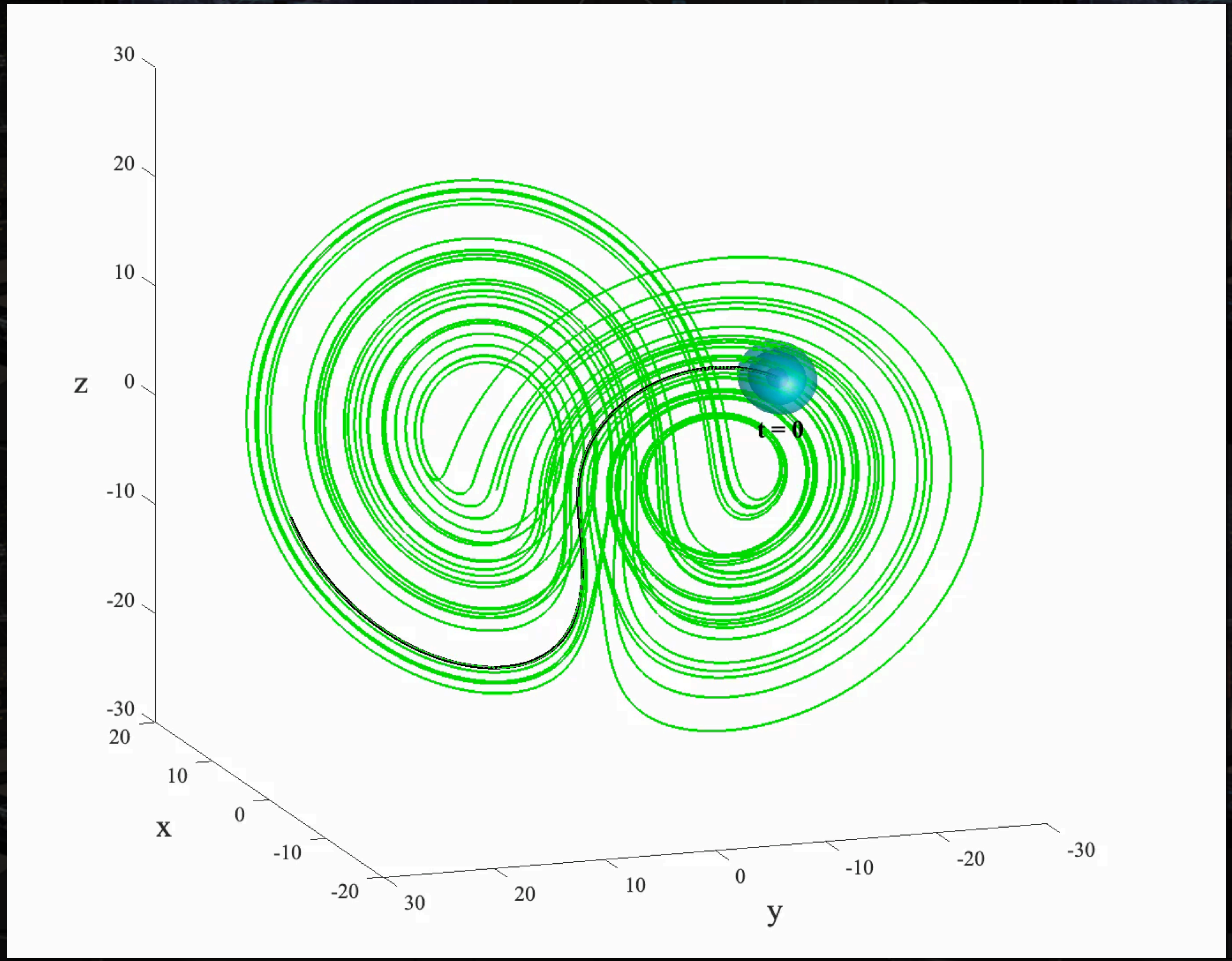
McElrath, Tim, et al. "Navigating low-energy trajectories to land on the surface of Europa." NTRS. (2021)

Current Landscape of Recursive Bayesian Filters



Motivation for New Non-Gaussian Filter

- To address the shortcomings of Gaussian filters, we utilize...



Grid-based Bayesian Estimation Exploiting Sparsity (GBEES)

An efficient Bayesian estimation method for representing and propagating uncertainty

- GBEES is a **2nd-order accurate**, Godunov finite volume method that **treats probability as a fluid**, flowing the PDF through phase space subject to the dynamics of the system
- Can handle deterministic/stochastic systems while **maintaining resolution**

Grid-based Bayesian Estimation Exploiting Sparsity (GBEES)

- GBEES consists of two distinct processes, one performed in **continuous-time**, the other in **discrete-time**:

1. The probability distribution function $p_{\mathbf{x}}(\mathbf{x}', t)$ is continuous-time marched via the **Fokker-Planck Equation**:

$$\frac{\partial p_{\boldsymbol{x}}(\boldsymbol{x}', t)}{\partial t} = -\frac{\partial f_i(\boldsymbol{x}', t) p_{\boldsymbol{x}}(\boldsymbol{x}', t)}{\partial x_i'} + \frac{1}{2}\frac{\partial^2 q_{ij} p_{\boldsymbol{x}}(\boldsymbol{x}', t)}{\partial x_i' \partial x_j'}$$

   - f_i: advection (EOMs) in the i^{th} dimension
   - q_{ij}: $(i, j)^{\text{th}}$ element of the spectral density ($Q = 0$, PDE is hyperbolic)

2. At discrete-time interval t_k, measurement $\boldsymbol{y}_k$ updates $p_{\boldsymbol{x}}(\boldsymbol{x}', t)$ via **Bayes' Theorem**:

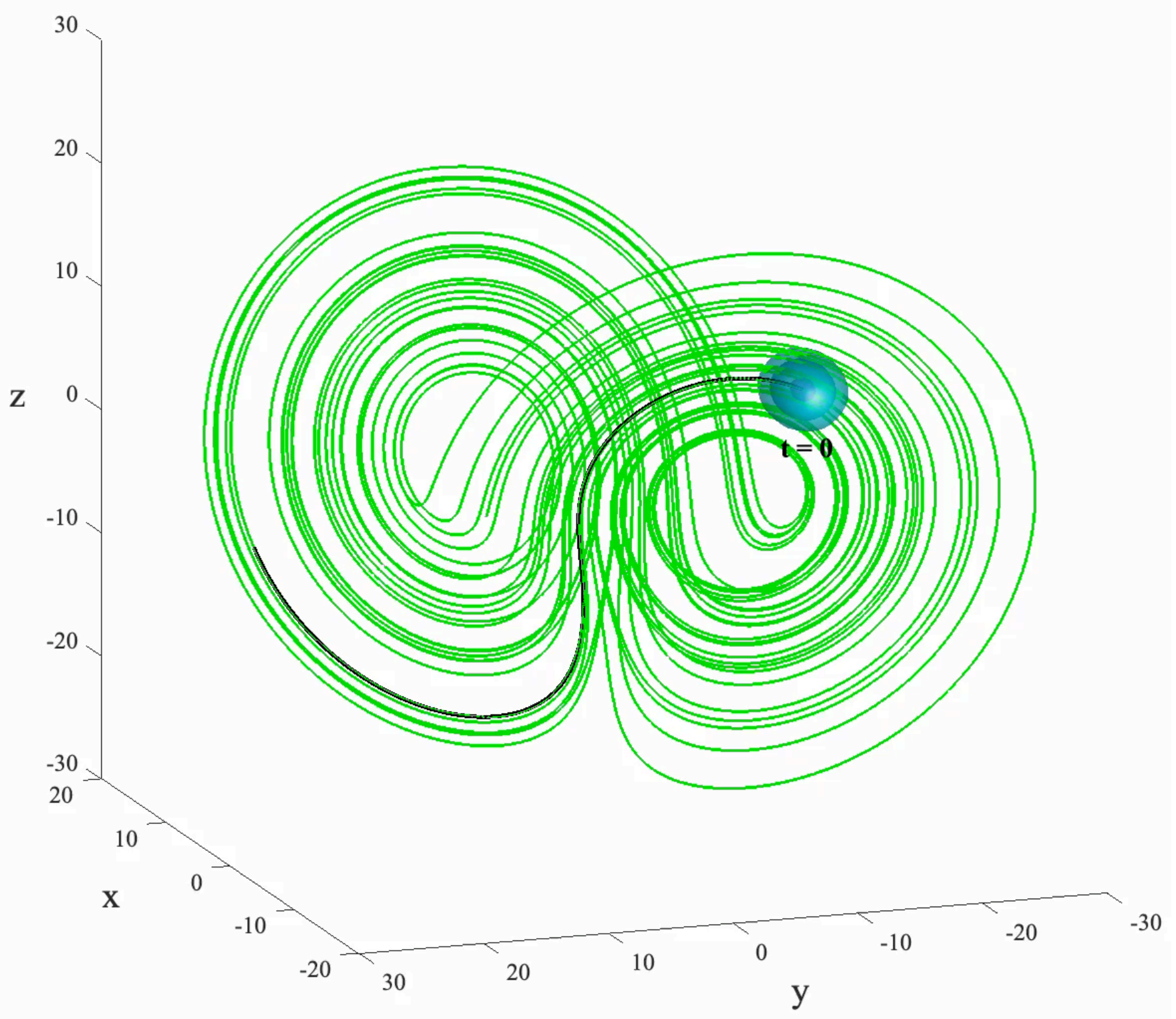
$$p_{\boldsymbol{x}}(\boldsymbol{x}', t_{k+}) = \frac{p_{\boldsymbol{y}}(\boldsymbol{y}_k \,|\, \boldsymbol{x}') p_{\boldsymbol{x}}(\boldsymbol{x}', t_{k-})}{C}$$

   - $p_{\mathbf{x}}(\mathbf{x}', t_{k+})$: a posteriori distribution
   - $p_{\mathbf{y}}(\mathbf{y}_k \,|\, \mathbf{x}')$: measurement distribution
   - $p_{\mathbf{x}}(\mathbf{x}', t_{k-})$: a priori distribution
   - C: normalization constant

Grid-based Bayesian Estimation Exploiting Sparsity (GBEES)

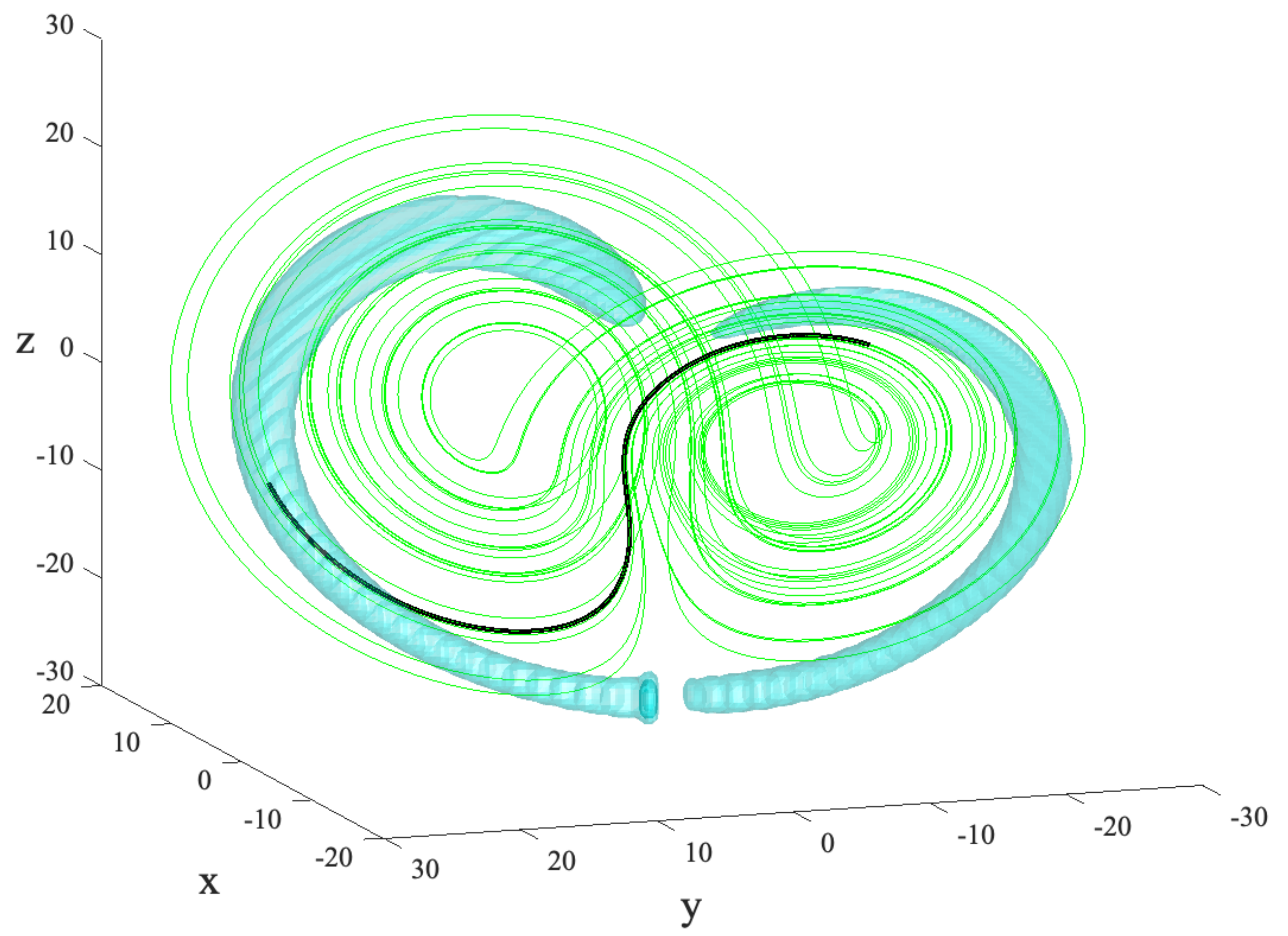
Continuous-time

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \frac{d\boldsymbol{x}}{dt} = \begin{bmatrix} \sigma(y-x) \\ -y-xz \\ -b(z+r)-xy \end{bmatrix}$$



Discrete-time

$$P(A\,|\,B) = \frac{P(B\,|\,A)\,P(A)}{P(B)}$$

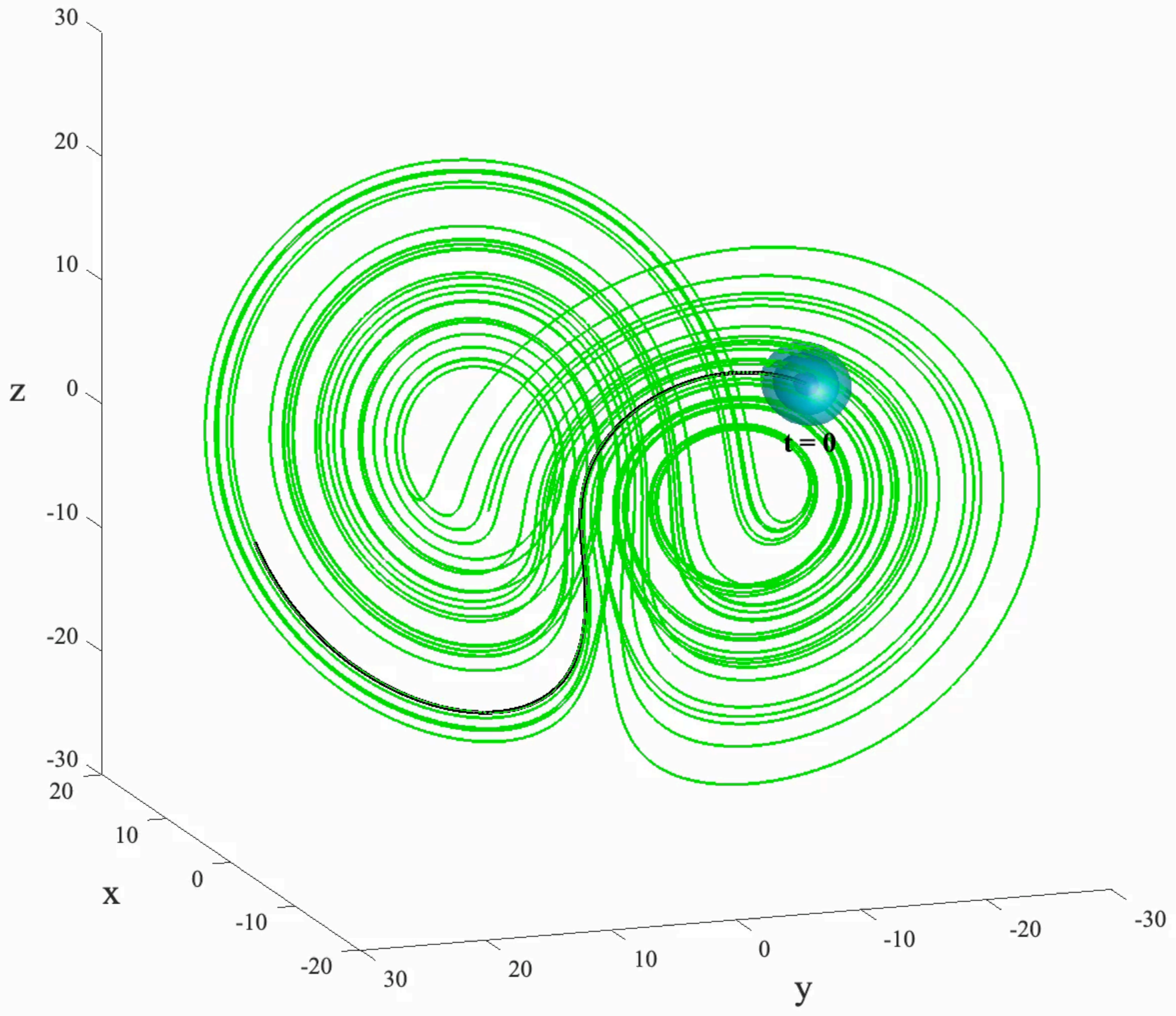


a priori

Bewley, Thomas, et al. "Efficient grid-based bayesian estimation of nonlinear low-dimensional systems with sparse non-gaussian pdfs." Automatica. (2012)

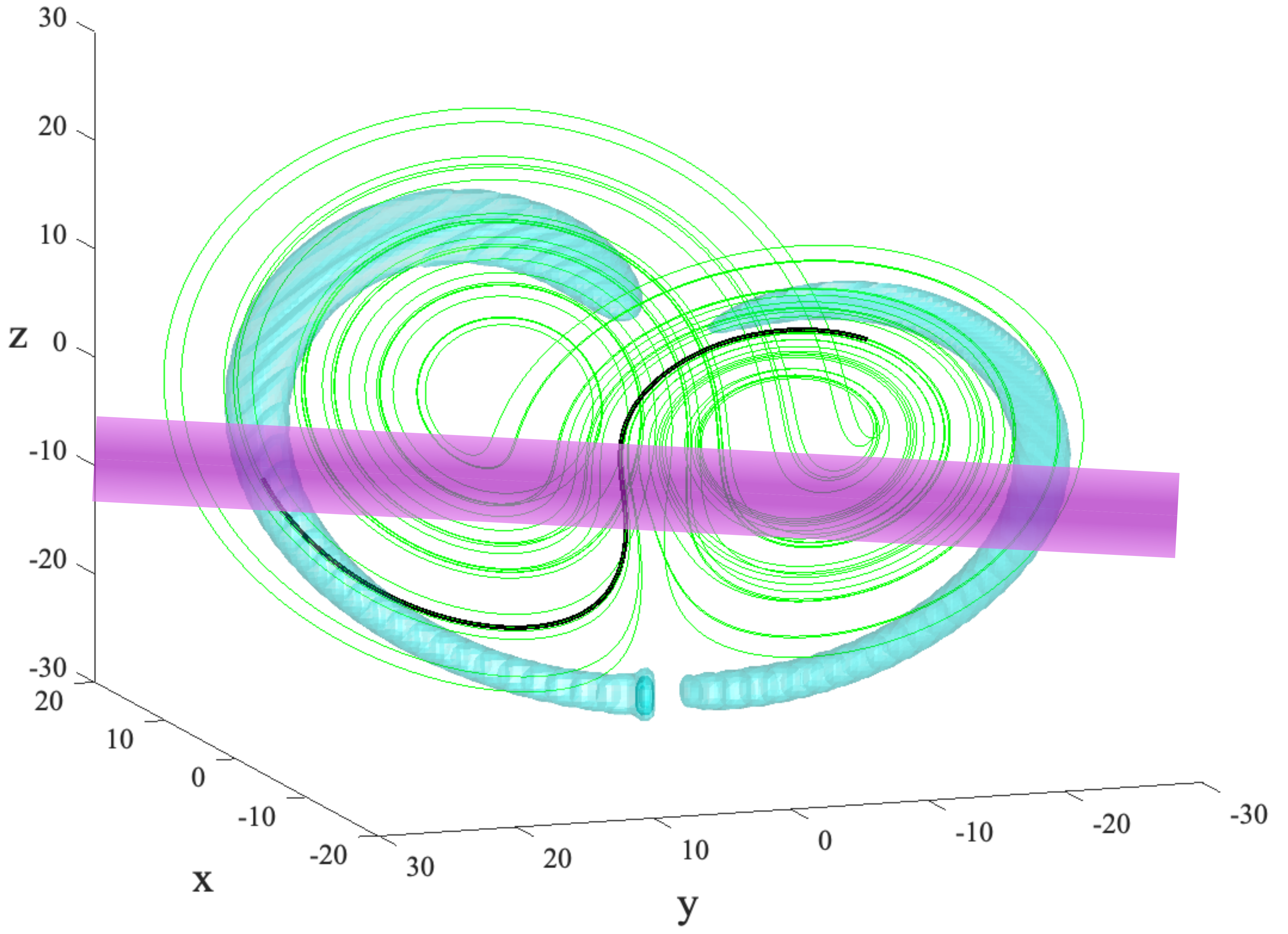
Continuous-time

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \frac{d\boldsymbol{x}}{dt} = \begin{bmatrix} \sigma(y-x) \\ -y-xz \\ -b(z+r)-xy \end{bmatrix}$$



Discrete-time

$$P(A\,|\,B) = \frac{P(B\,|\,A)\,P(A)}{P(B)}$$

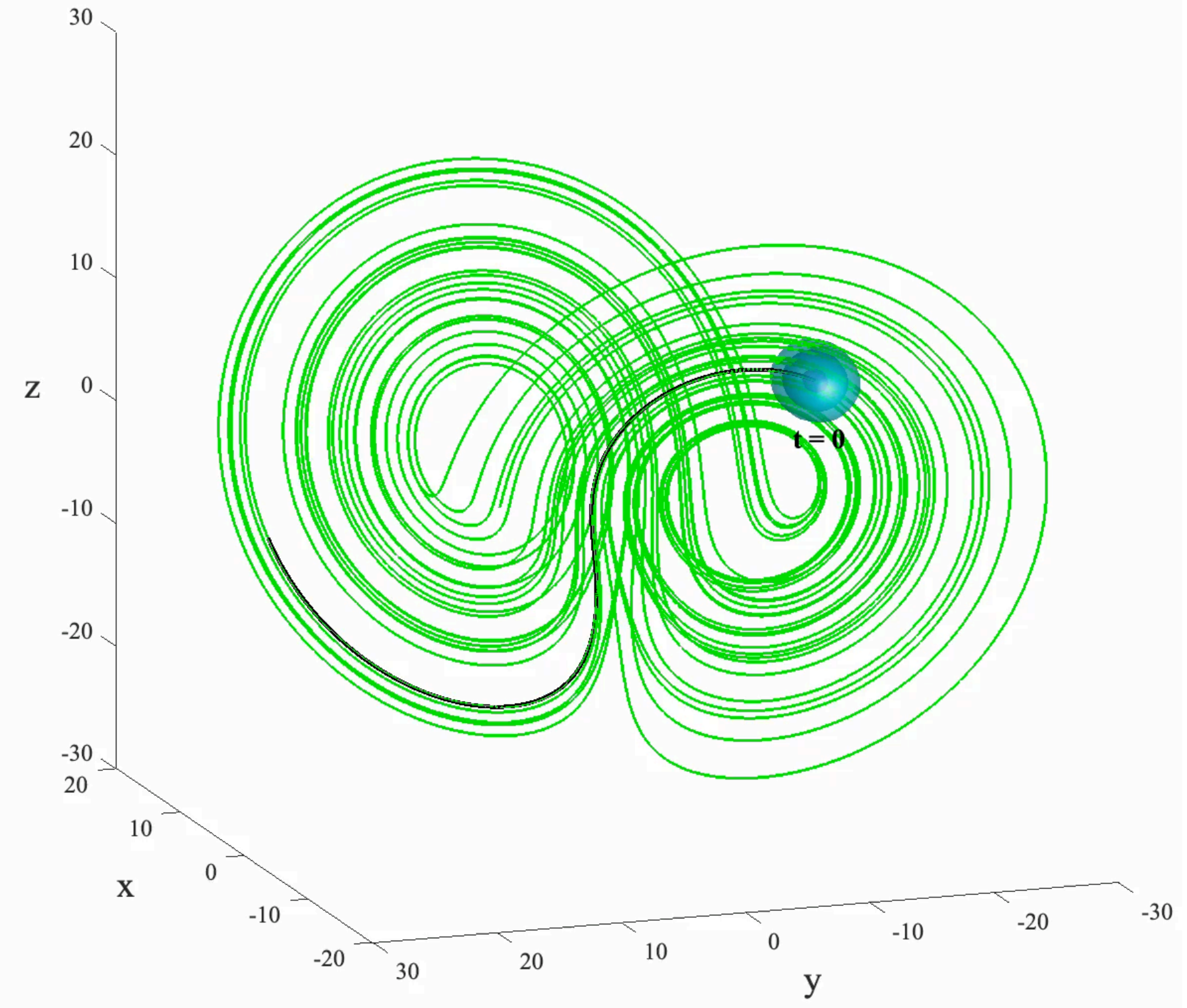


a priori × **likelihood**

Bewley, Thomas, et al. "Efficient grid-based bayesian estimation of nonlinear low-dimensional systems with sparse non-gaussian pdfs." Automatica. (2012)

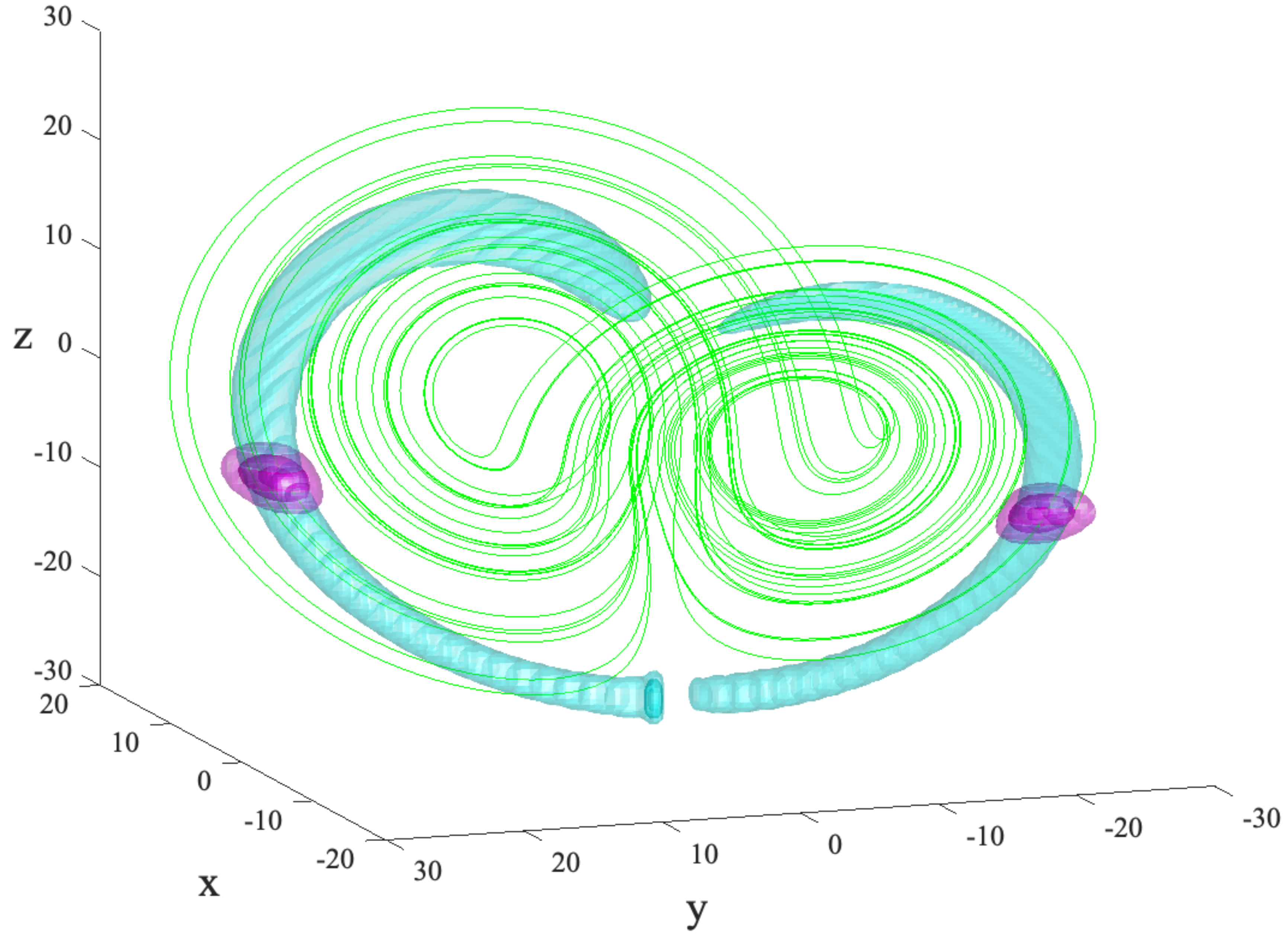
Continuous-time

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \frac{d\boldsymbol{x}}{dt} = \begin{bmatrix} \sigma(y-x) \\ -y - xz \\ -b(z+r) - xy \end{bmatrix}$$



Discrete-time

$$P(A\,|\,B) = \frac{P(B\,|\,A)\,P(A)}{P(B)}$$

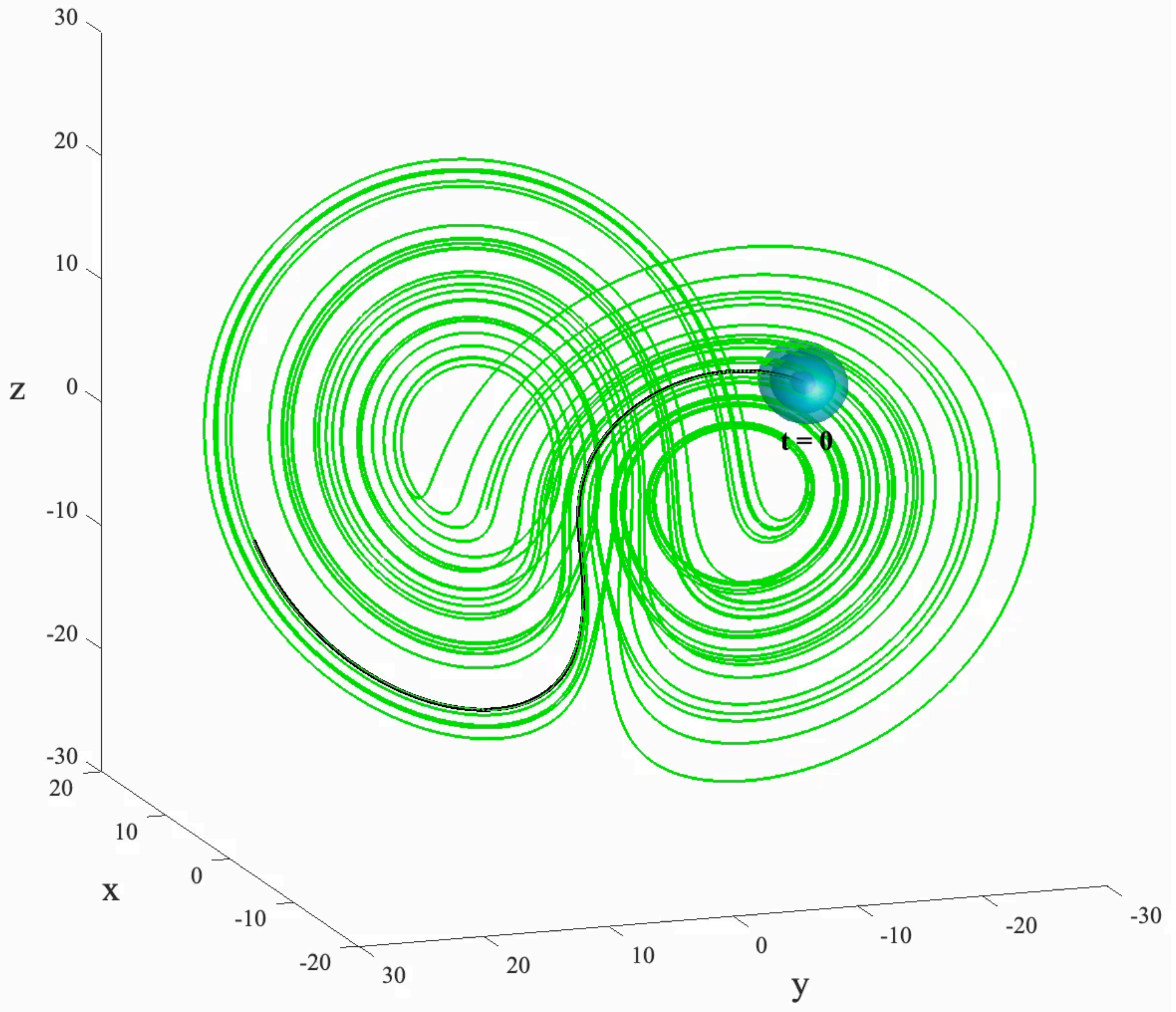


a priori × **likelihood** = ***a posteriori***

Bewley, Thomas, et al. "Efficient grid-based bayesian estimation of nonlinear low-dimensional systems with sparse non-gaussian pdfs." Automatica. (2012)

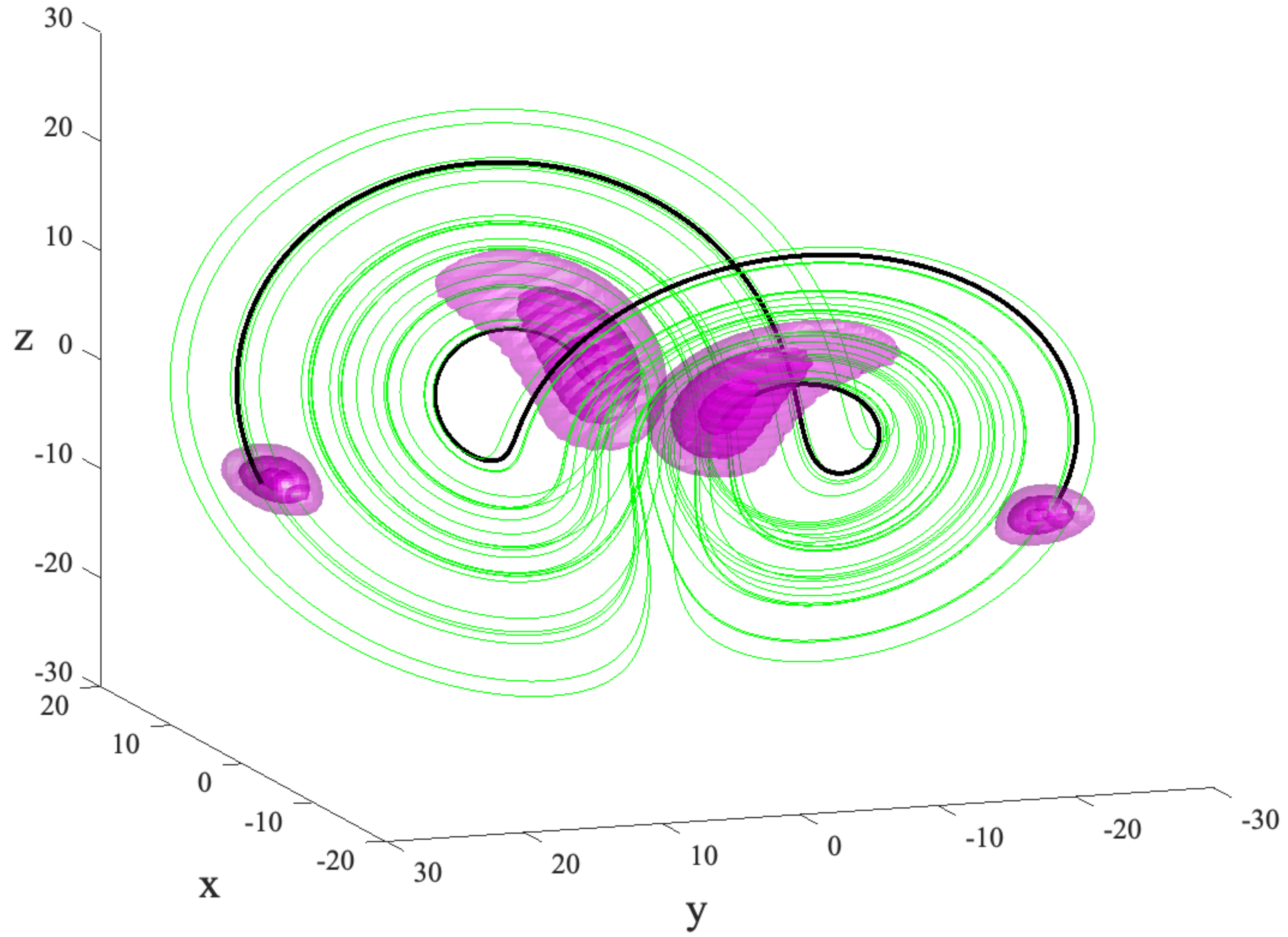
Continuous-time

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \frac{d\boldsymbol{x}}{dt} = \begin{bmatrix} \sigma(y-x) \\ -y-xz \\ -b(z+r)-xy \end{bmatrix}$$



Discrete-time

$$P(A\,|\,B) = \frac{P(B\,|\,A)\,P(A)}{P(B)}$$



a priori × **likelihood** = ***a posteriori***

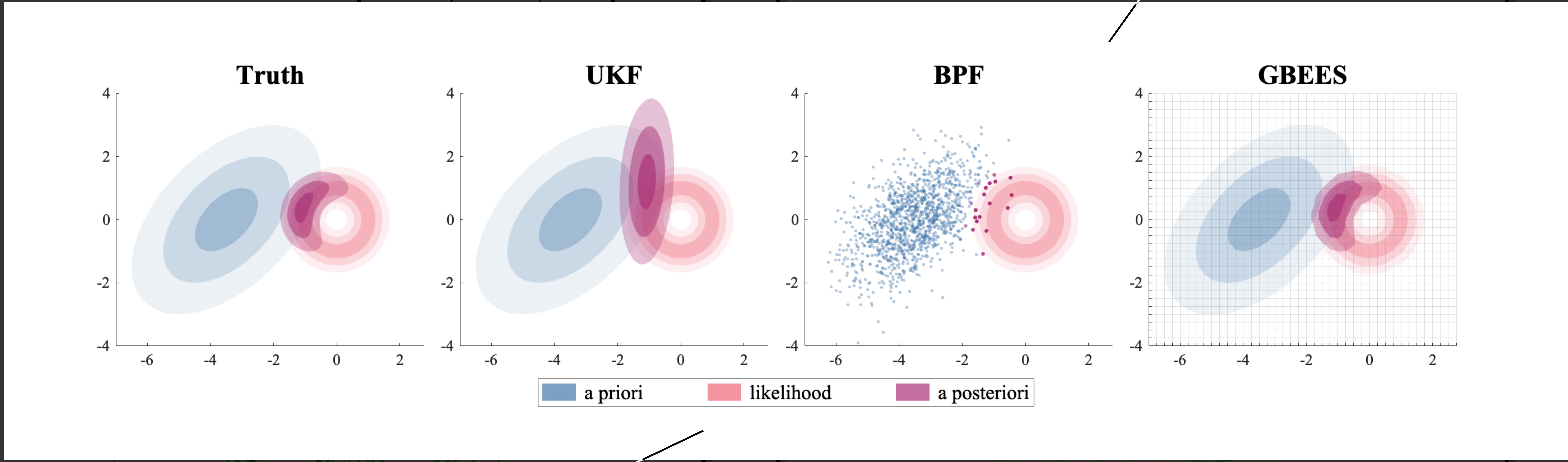
Bewley, Thomas, et al. "Efficient grid-based bayesian estimation of nonlinear low-dimensional systems with sparse non-gaussian pdfs." Automatica. (2012)

Grid-based **Bayesian Estimation** Exploiting Sparsity (GBEES)

$n_{\text{particles}} = n_{\text{grid cells}}$

Continuous-time

Discrete-time



$$x = \begin{bmatrix} x \\ y \end{bmatrix}, \quad h(x) = |x|$$

a priori × **likelihood** = ***a posteriori***

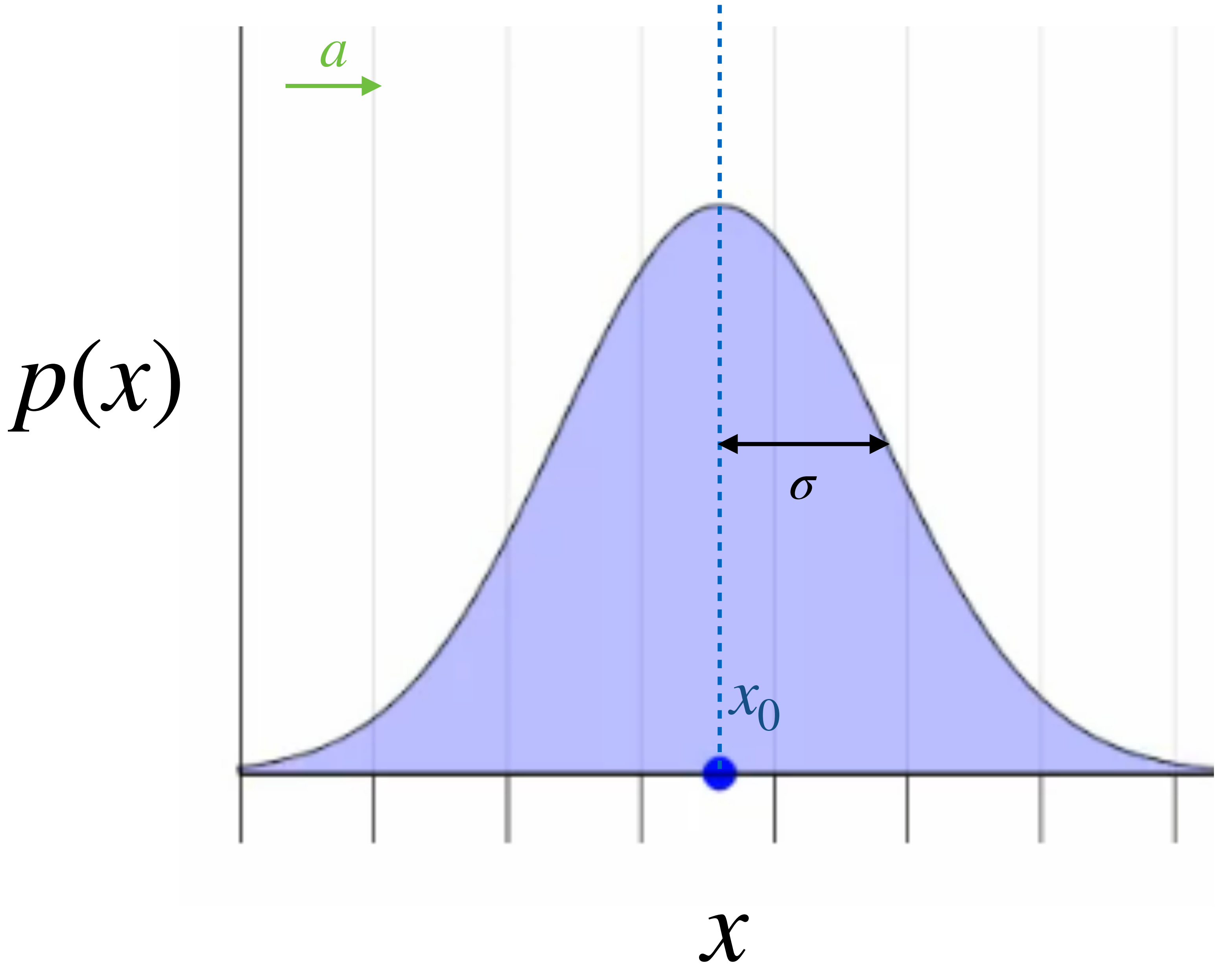
Bewley, Thomas, et al. "Efficient grid-based bayesian estimation of nonlinear low-dimensional systems with sparse non-gaussian pdfs." Automatica. (2012)

Grid-based Bayesian Estimation Exploiting Sparsity (GBEES)

- Consider a 1-dimensional, linear test example:

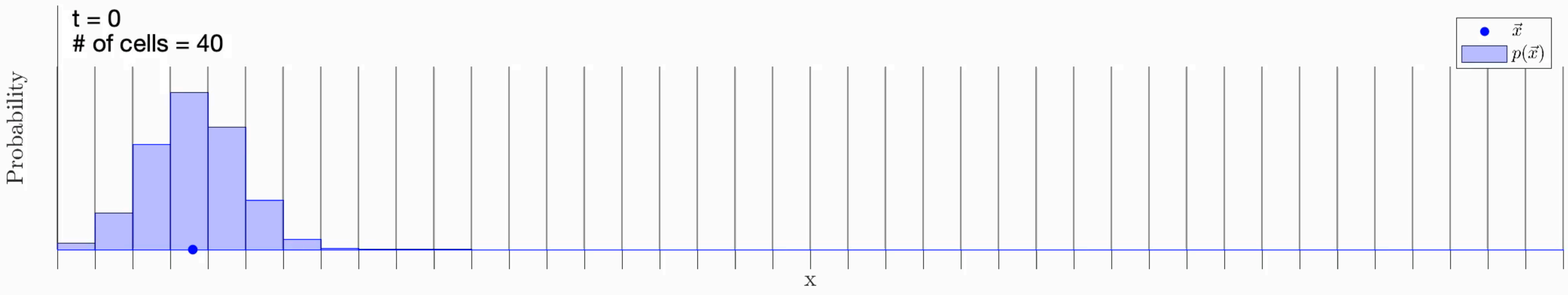
$$\boldsymbol{x} = [x], \quad \frac{d\boldsymbol{x}}{dt} = [a], \quad a > 0$$

- Initial observation of $x(t)$ results in a Gaussian PDF $p(x)$ centered about x_0 with standard deviation σ

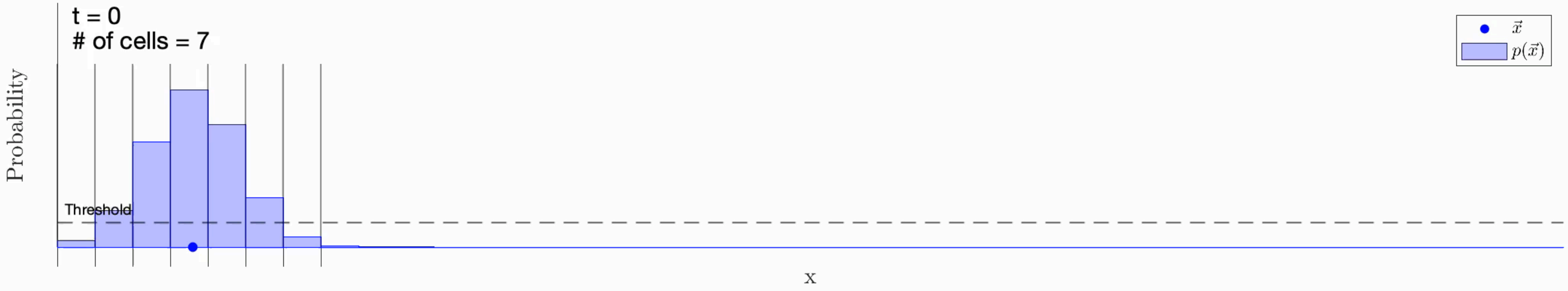


How does $p(x)$, governed by $d\boldsymbol{x}/dt$, change with respect to t?

Ignoring sparsity



Exploiting sparsity

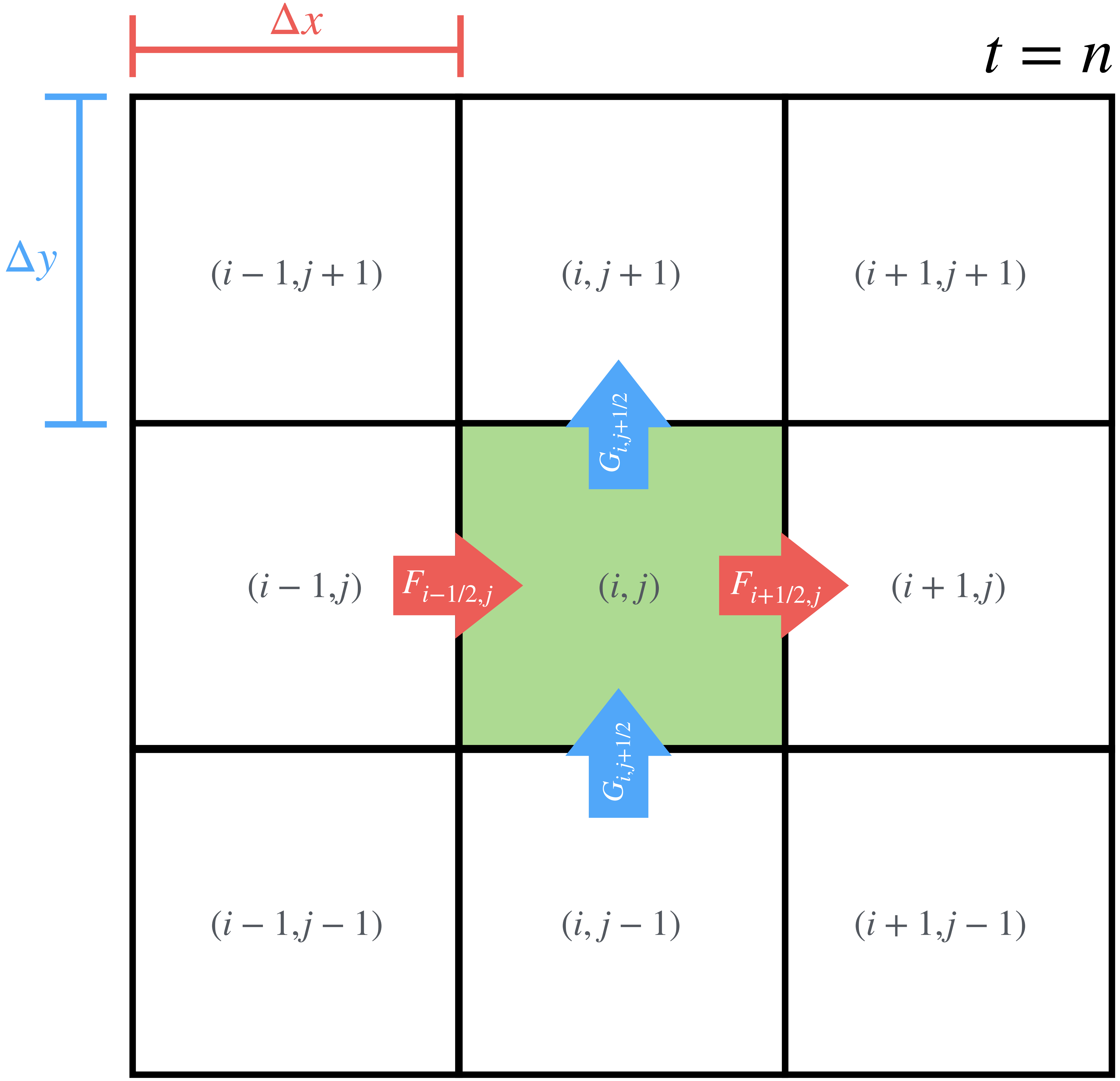


Not GBEES, just a visual aid

General Formulation

Godunov Upwind Scheme - Fully Discretized, 2nd-order, Taylor Approximation

- A Godunov-type finite volume method implemented on a uniform Cartesian 2D mesh



$$\frac{p_{i,j}^{n+1} - p_{i,j}^{n}}{\Delta t} = -\frac{F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}}{\Delta x} - \frac{G_{i,j+1/2}^{n} - G_{i,j-1/2}^{n}}{\Delta y}$$

- $p_{i,j}^{n+1}$: probability at time step $n+1$ at cell (i, j)
- $p_{i,j}^{n}$: probability at time step n at cell (i, j)
- Δt: size of time step
- $F_{i-1/2,j}^{n}$: flux a half grid length back in the x-direction
- $F_{i+1/2,j}^{n}$: flux a half grid length forward in the x-direction
- $G_{i-1/2,j}^{n}$: flux a half grid length back in the y-direction
- $G_{i+1/2,j}^{n}$: flux a half grid length forward in the y-direction
- Δx: x-grid width
- Δy: y-grid width

Instead of flux being a function of volume and advection, flux is a function of probability and the equations of motion!

Goal: Find practical trajectories where orbital uncertainty becomes non-Gaussian and apply RBFs

Astrodynamic Applications: Three-Body Problem

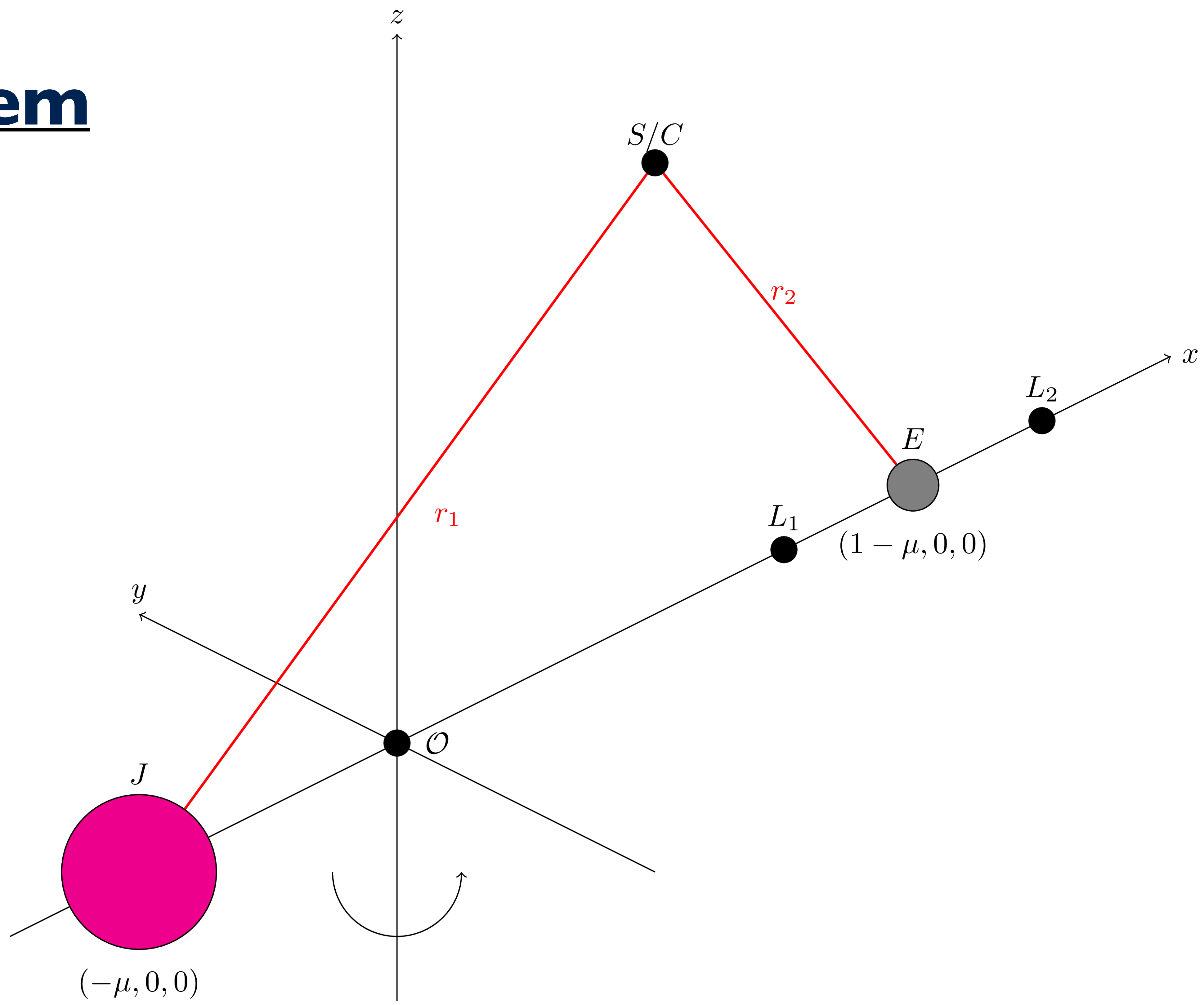
Circular Restricted Three-Body Problem (CR3BP)

- We look to apply the developed framework to orbital uncertainty propagation

Circular Restricted Three-Body Problem

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \quad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 2\dot{y} + \Omega_x \\ -2\dot{x} + \Omega_y \\ \Omega_z \end{bmatrix}$$

where $\Omega(x, y) = \dfrac{x^2 + y^2}{2} + \dfrac{1-\mu}{r_1} + \dfrac{\mu}{r_2} + \dfrac{\mu(1-\mu)}{2}$



- We use initial conditions generated from the JPL Three-Body Periodic Orbit Catalog

Park, Ryan. "Jet Propulsion Laboratory Three-Body Periodic Orbit Catalog." (2024)

Astrodynamic Applications: Three-Body Problem

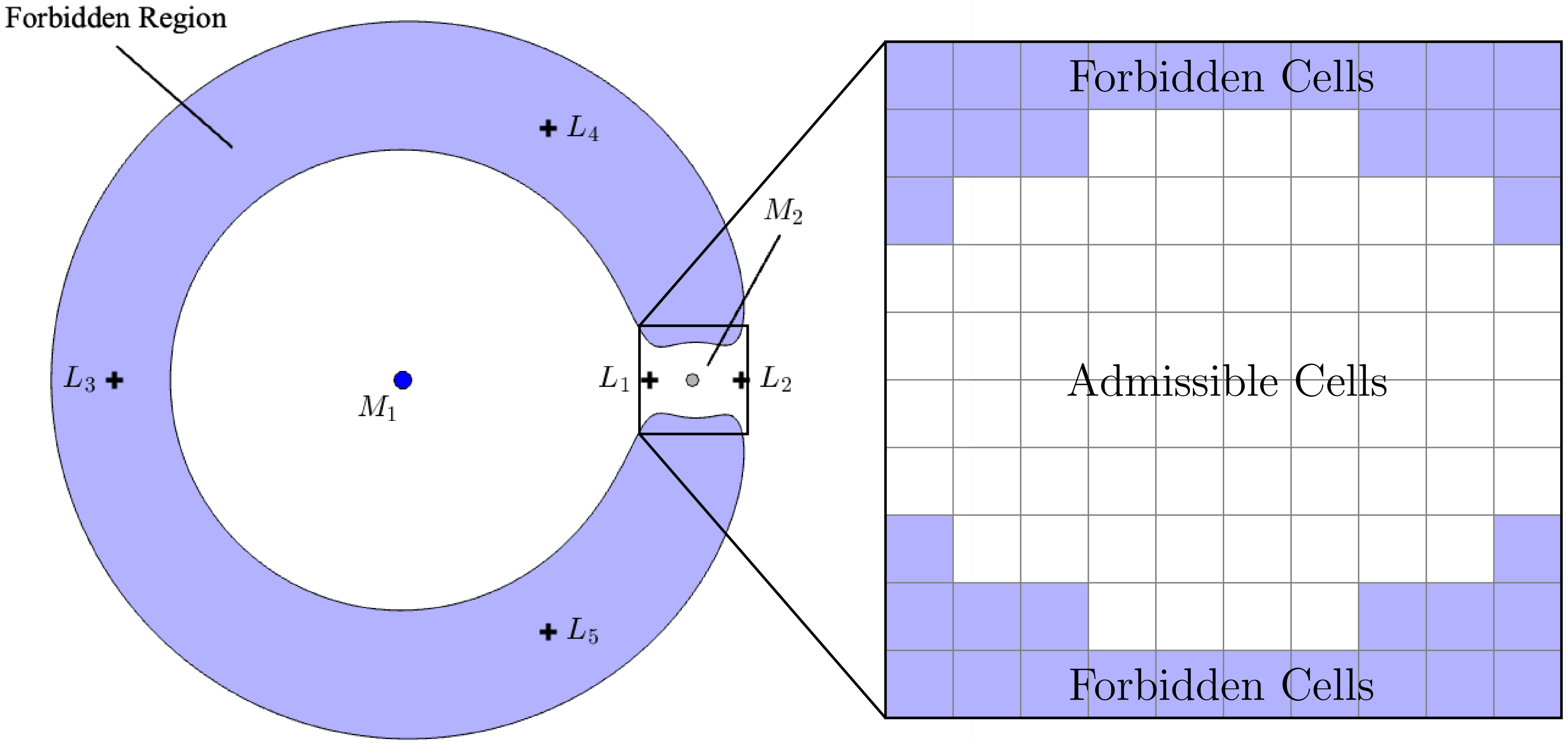
GBEES Jacobi Bounding

- One integral of motion exists for the CR3BP

$$C(x, y, z, \dot{x}, \dot{y}, \dot{z}) = x^2 + y^2 + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} + \mu(1-\mu) - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$$

- GBEES is a 2nd-order accurate numerical scheme, so C is not necessarily conserved
- Instead, we hardcode this requirement into the grid creation

  - ✳ When $\dot{x} = \dot{y} = \dot{z} = 0$, C is the zero-velocity curves
  - ✳ Phase space is discretized as would be done for GBEES



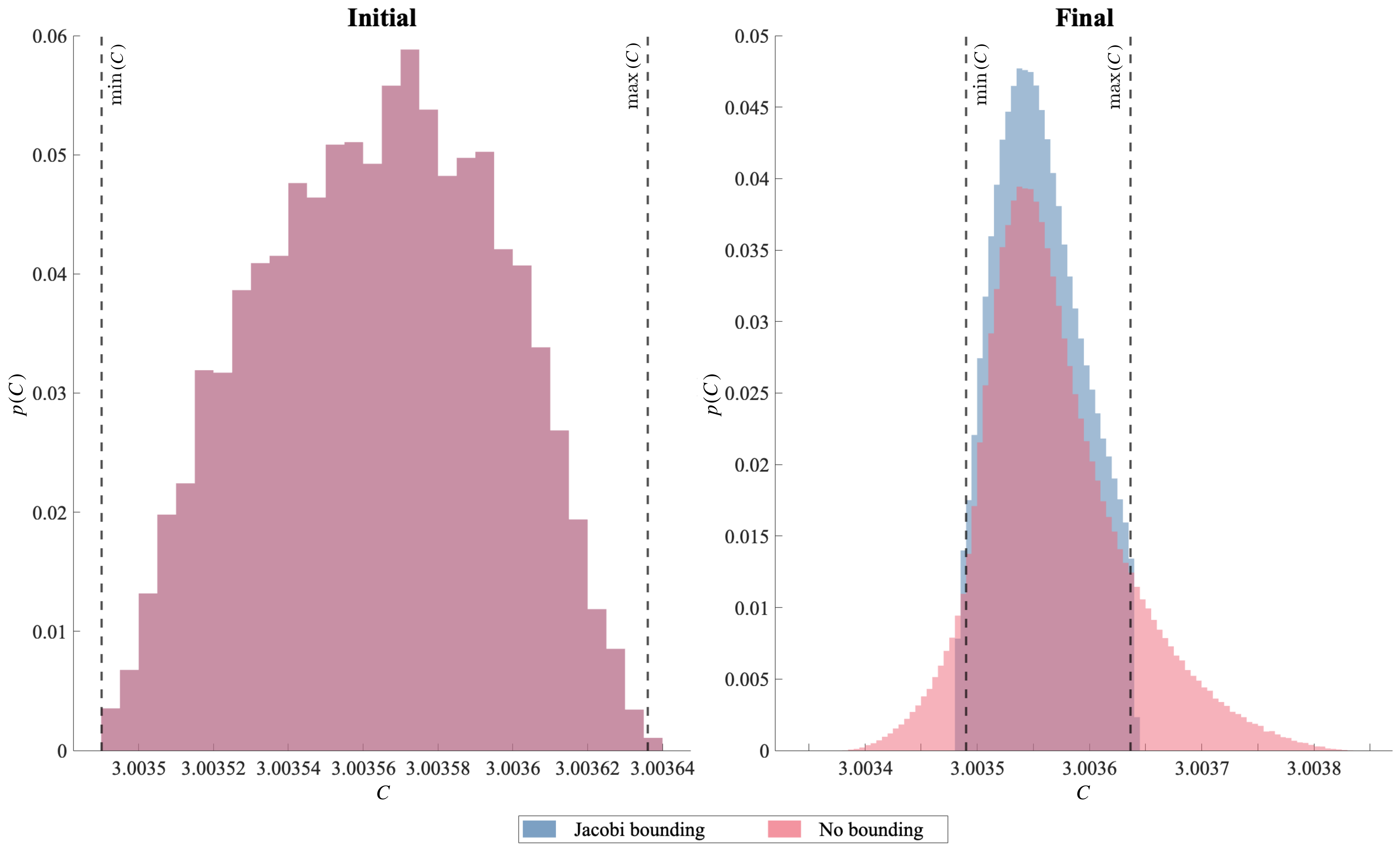
GBEES Jacobi Bounding

- One integral of motion exists for the CR3BP

$$C(x,y,z,\dot{x},\dot{y},\dot{z}) = x^2 + y^2 + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} + \mu(1-\mu) - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$$

- GBEES is 2nd-order accurate, so C is not necessarily conserved numerically
- Instead, we hardcode this requirement into the grid creation

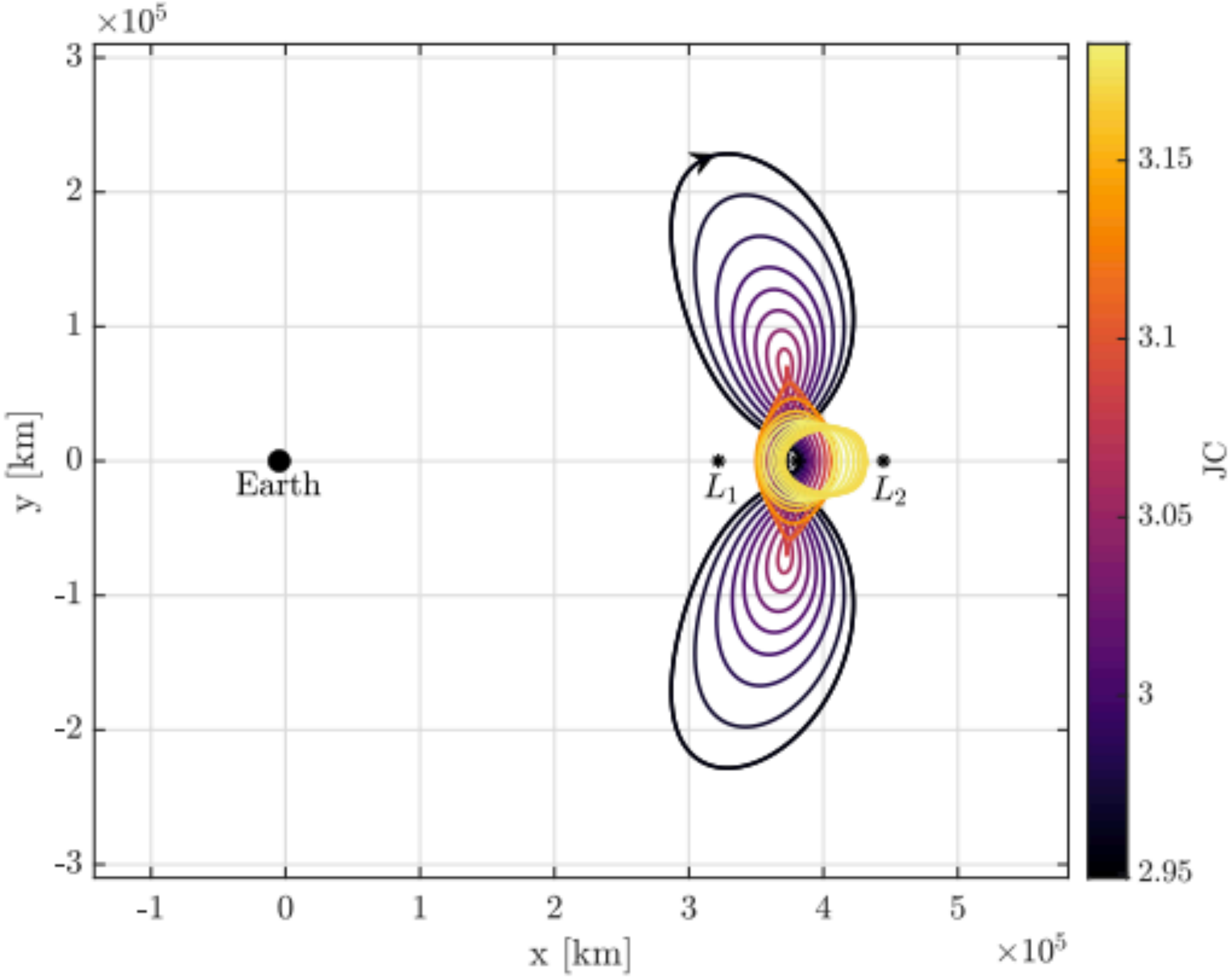
  * For the initial PDF, there exists a $\min(C)$ and a $\max(C)$
  * We ensure that all grid cells that are created in the propagation period are between these bounds

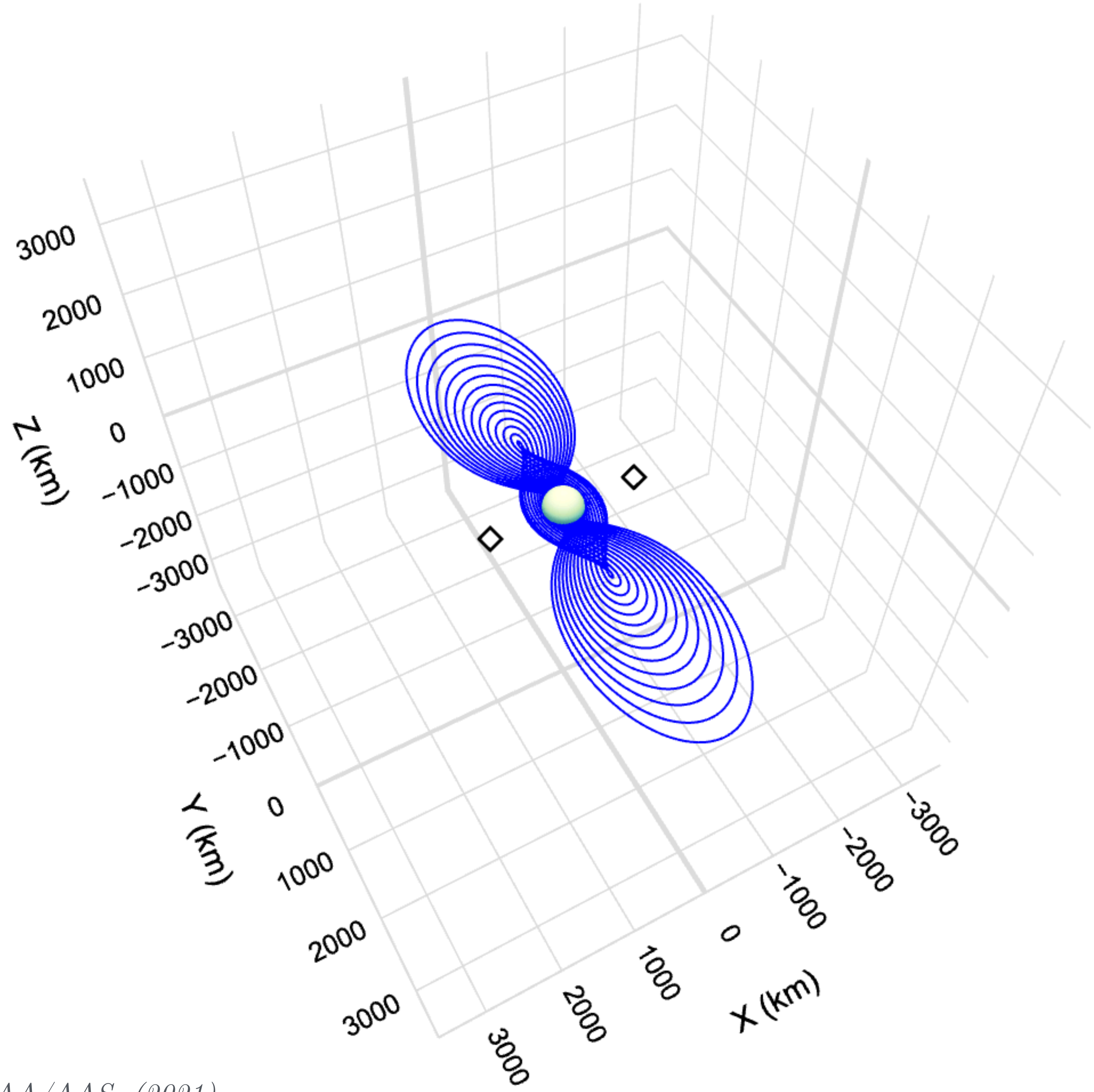


Circular Restricted Three-Body Problem (CR3BP)

Distant Prograde Orbits

- A family of planar, P_2-centered, stable/unstable periodic orbits that emerge from the dynamics of the CR3BP are Distant Prograde Orbits (DPOs)





Gupta, Maaninee et al. "Earth-Moon multi-body orbits to facilitate cislunar surveillance activities." AIAA/AAS. (2021)

Circular Restricted Three-Body Problem (CR3BP)

Distant Prograde Orbits

- A family of planar, P_2-centered, stable/unstable period orbits that emerge from the dynamics of the CR3BP are Distant Prograde Orbits (DPOs)

<u>Stability is a function of C</u>

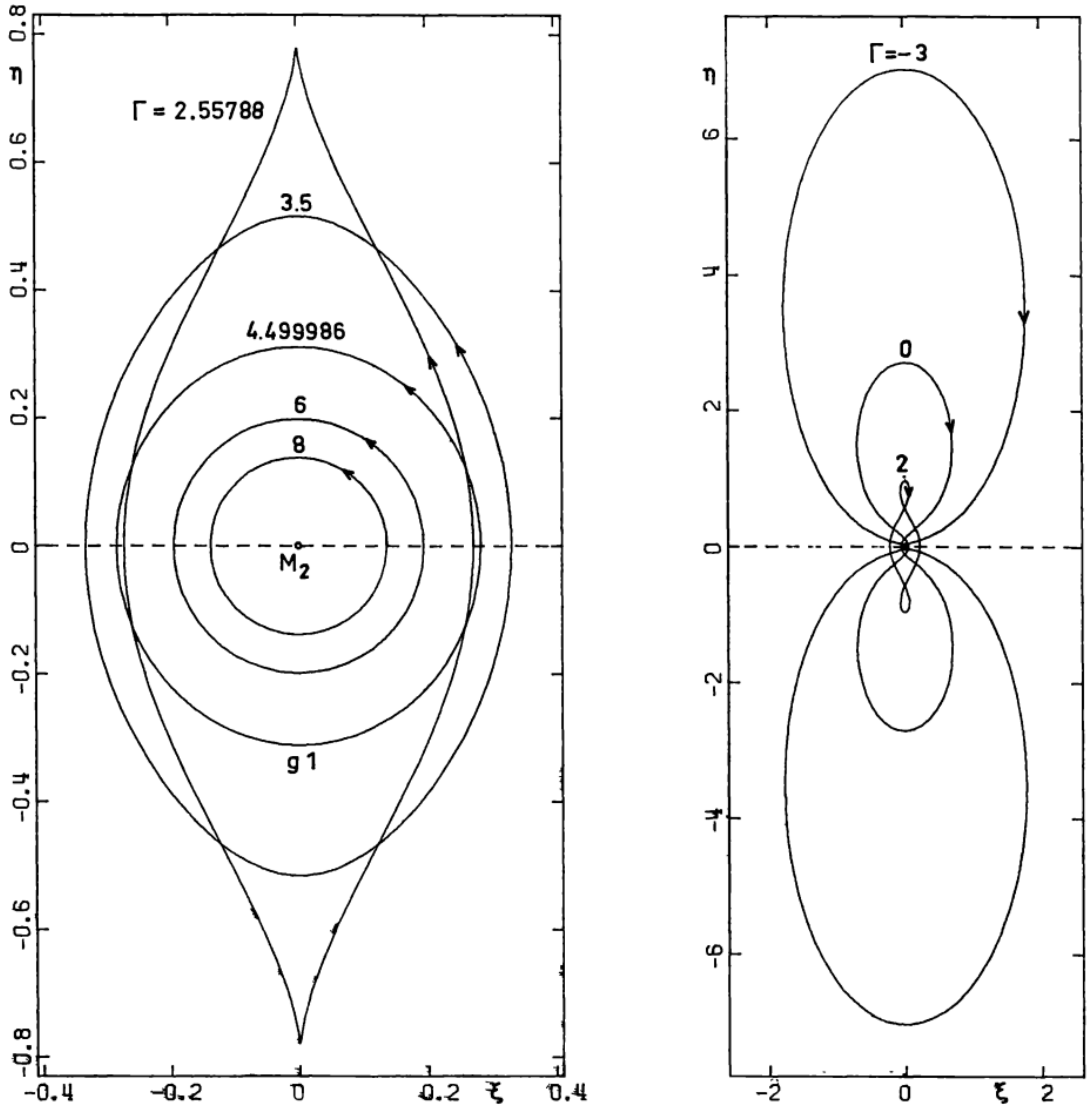


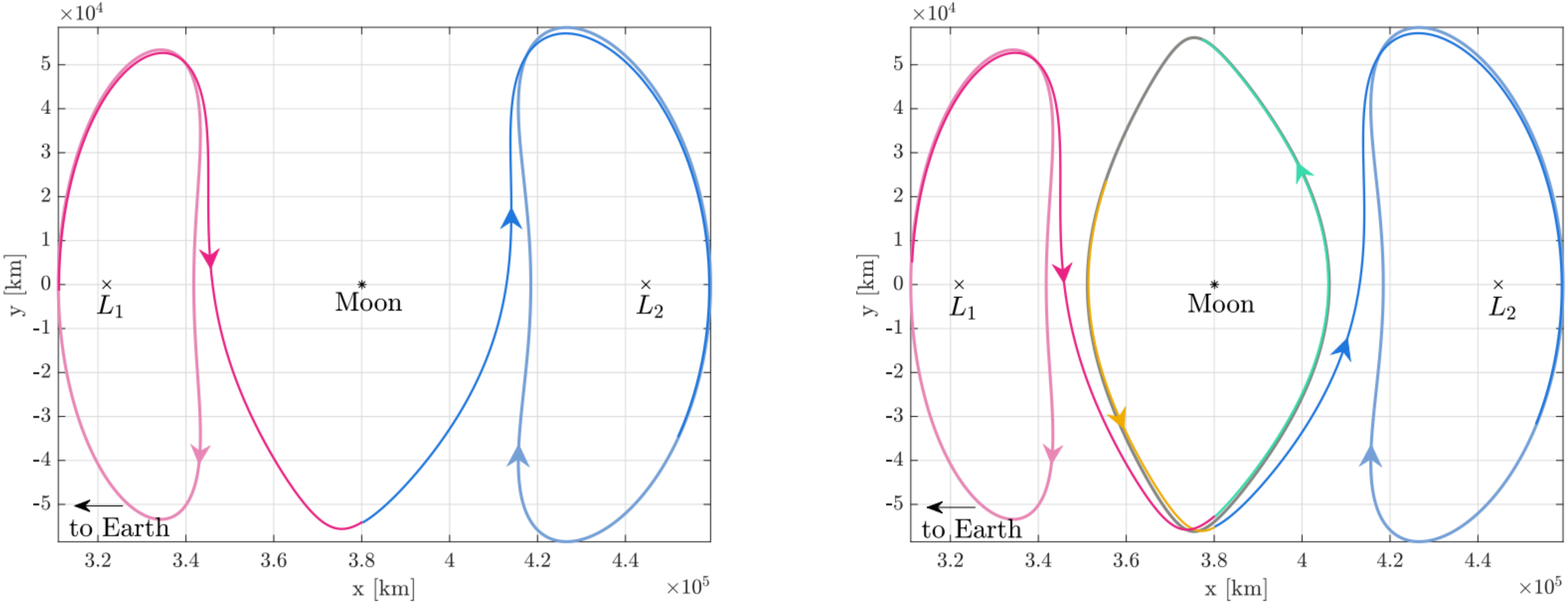
Fig. 4. Family g of periodic orbits. Note that the two parts of the figure have different scales. Orbits with $\Gamma > 4.499986$ are stable, the others are unstable

Henon, Michel. "Numerical exploration of the restricted problem, V." AAP. (1969)

Circular Restricted Three-Body Problem (CR3BP)

Distant Prograde Orbits

- A family of planar, P_2-centered, stable/unstable period orbits that emerge from the dynamics of the CR3BP are Distant Prograde Orbits (DPOs)



DPOs serve as heteroclinic link between L_1 and L_2 Lyapunov orbits

Gupta, Maaninee et al. "Earth-Moon multi-body orbits to facilitate cislunar surveillance activities." AIAA/AAS. (2021)

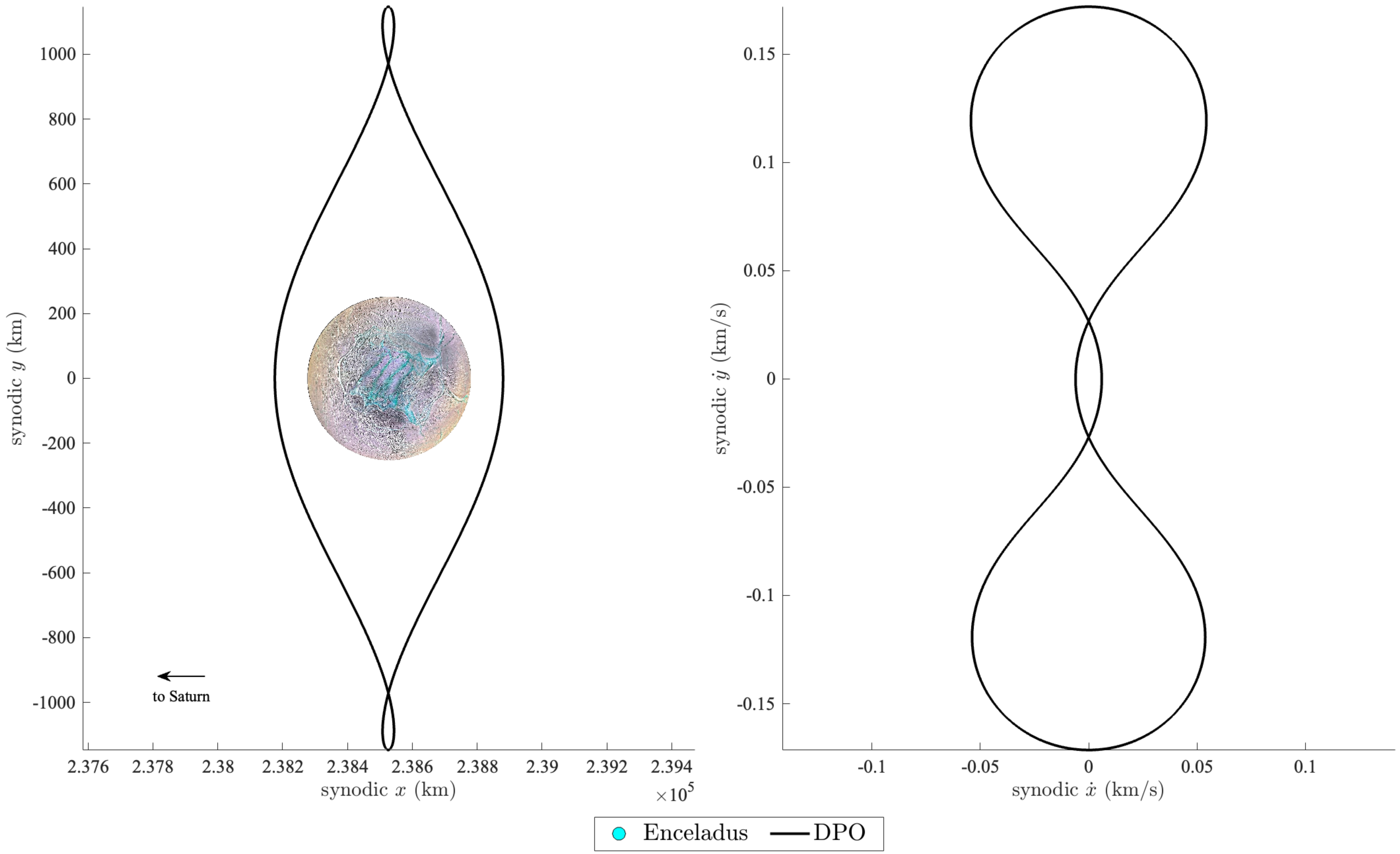
Astrodynamic Applications: Three-Body Problem

DPO uncertainty propagation conditions

- We choose an unstable DPO in the Saturn-Enceladus system for testing GBEES and other selected RBFs

$$\boldsymbol{x}_0 = \begin{bmatrix} 1.001471 & (\text{LU}) \\ -1.751810E-5 & (\text{LU}) \\ 0.0 & (\text{LU}) \\ 7.198783E-5 & (\text{LU/TU}) \\ 1.363392E-2 & (\text{LU/TU}) \\ 0.0 & (\text{LU/TU}) \end{bmatrix}$$

μ	1.9011E-7
LU (km)	238529
TU (s)	18913
T (hr)	19.5811
C	3.000078
SI	3.0187E+2



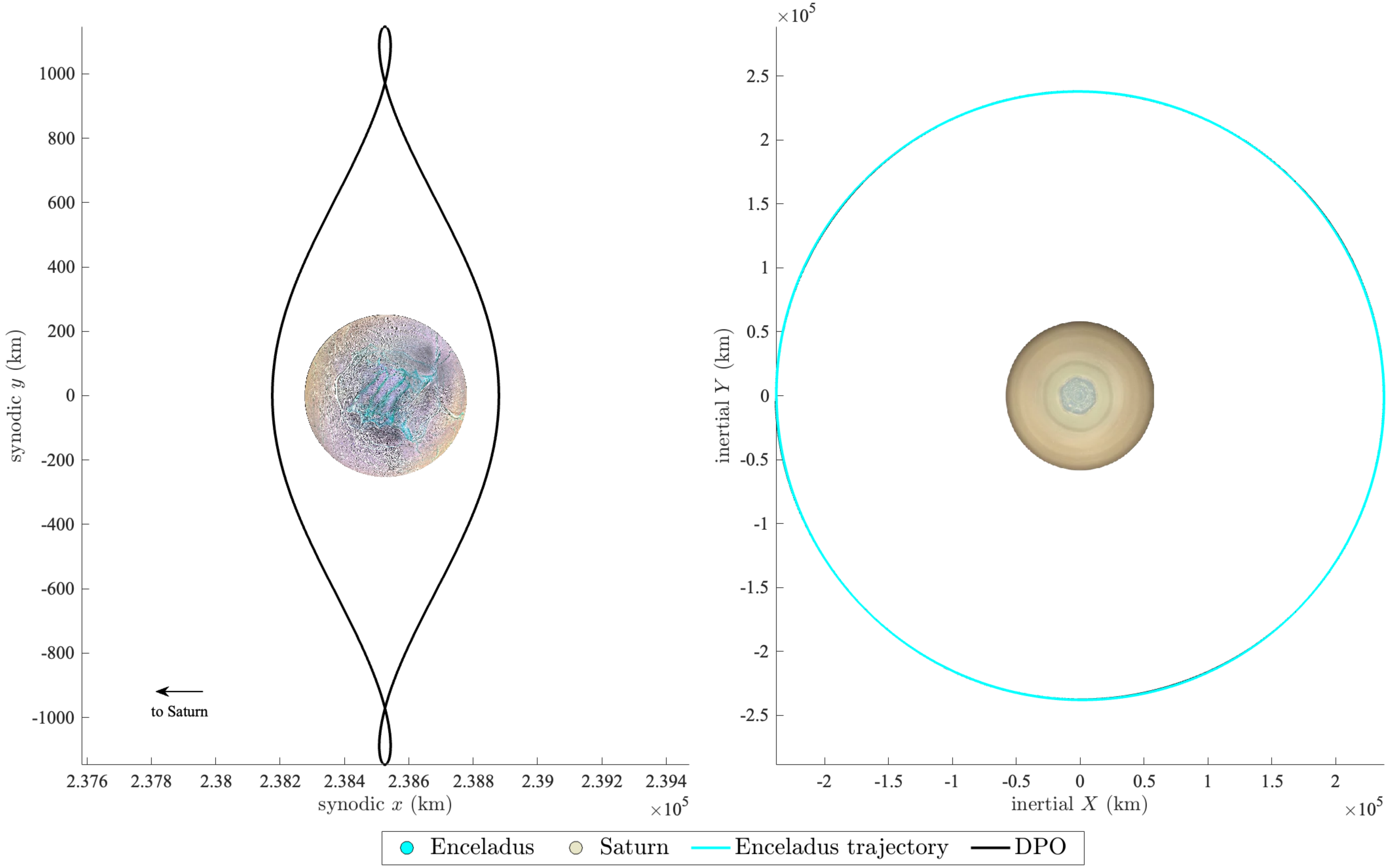
Astrodynamic Applications: Three-Body Problem

DPO uncertainty propagation conditions

- We choose an unstable DPO in the Saturn-Enceladus system for testing GBEES and other RBFs of note

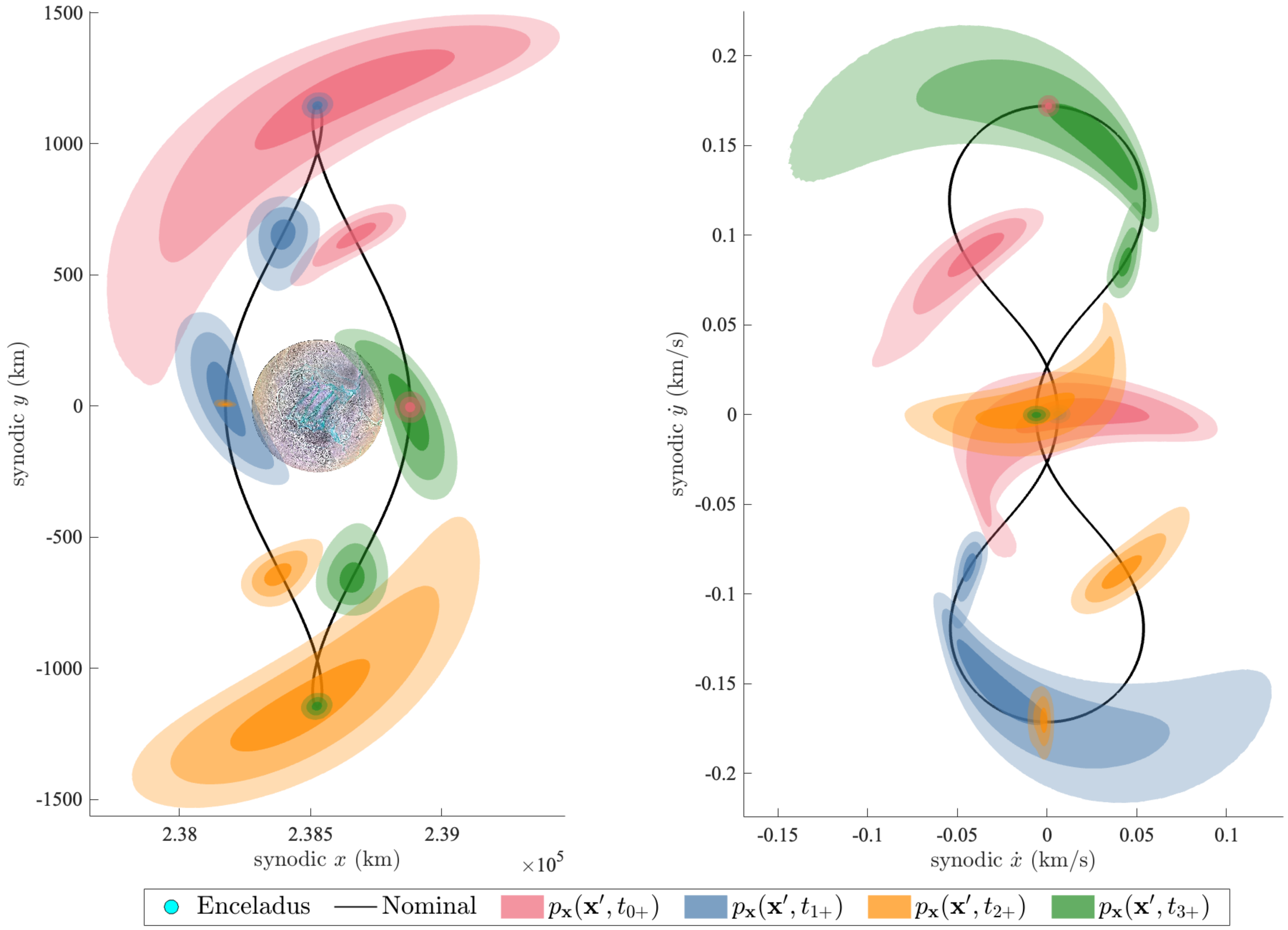
$$\boldsymbol{y} = \begin{bmatrix} \rho \\ \theta \\ \dot{\rho} \end{bmatrix} = \boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \sqrt{(x-1+\mu)^2 + y^2} \\ \tan^{-1}\left(\frac{y}{x-1+\mu}\right) \\ \frac{(x-1+\mu)\dot{x} + y\dot{y}}{\rho} \end{bmatrix}$$

σ_{ρ_0} (km)	20.0
σ_{θ_0} (rad)	1.74533E-2
$\sigma_{\dot{\rho}_0}$ (km/s)	2.0E-3
Δt_y (hr)	4.895



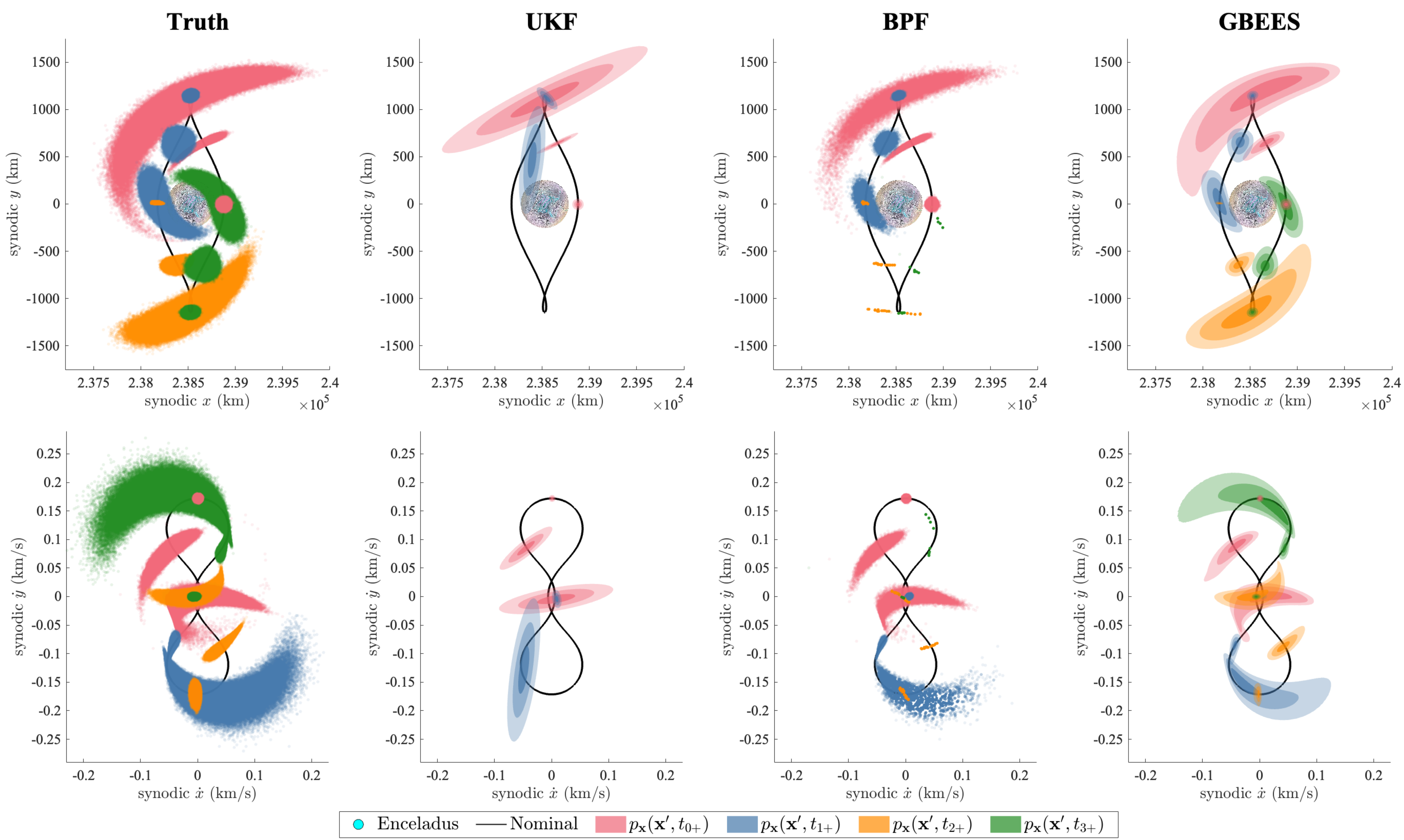
Saturn-Enceladus Distant Prograde Orbit Propagation

GBEES compared with other RBFs



Notes

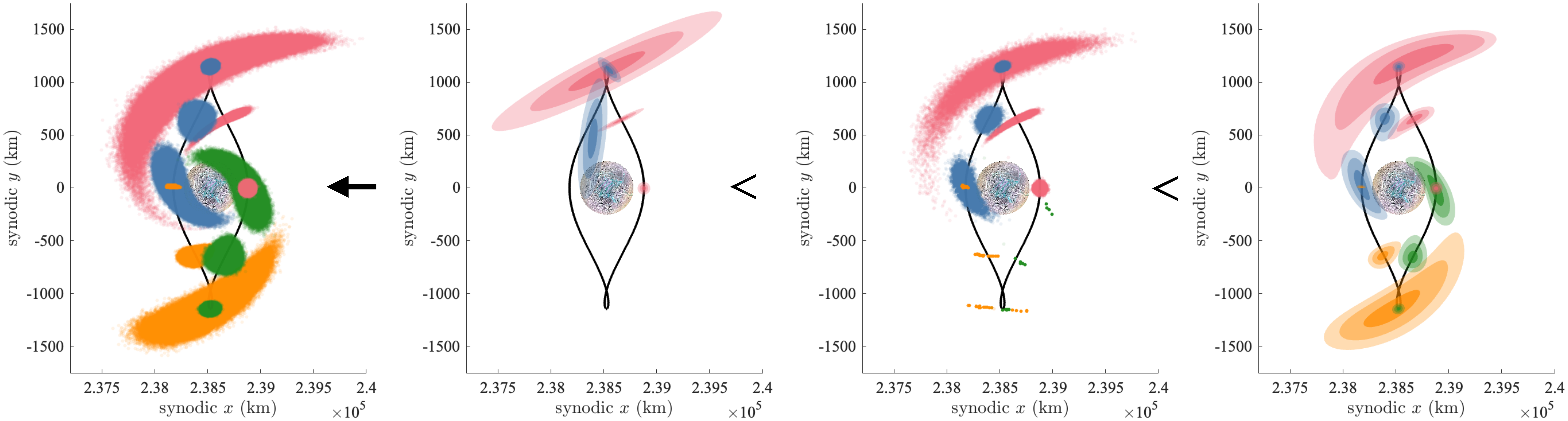
- Coordinates are in the synodic frame
- The true PDFs propagated by GBEES are 4D — these PDFS are the 4D ones integrated over velocity/position for visualization of the 2D position/velocity PDFs
- A change in color indicates a measurement update, with four occurring over this propagation period


Truth
UKF
BPF
GBEES
synodic y (km)
synodic x (km)
×10^5
synodic ẏ (km/s)
synodic ẋ (km/s)
Enceladus
Nominal
$p_{\mathbf{x}}(\mathbf{x}', t_{0+})$
$p_{\mathbf{x}}(\mathbf{x}', t_{1+})$
$p_{\mathbf{x}}(\mathbf{x}', t_{2+})$
$p_{\mathbf{x}}(\mathbf{x}', t_{3+})$

Saturn-Enceladus Distant Prograde Orbit Propagation

Non-Gaussian metric of comparison

- How do we compare the accuracy of these highly non-Gaussian distributions?



- A non-normal measure of the dissimilarity of distributions — **the Bhattacharyya Coefficient**

$$BC(P,Q) = \sum_{x \in \chi} \sqrt{P(\boldsymbol{x})\, Q(\boldsymbol{x})} \qquad \text{where} \qquad 0 \le BC(P,Q) \le 1$$

- $BC(P,Q) = 1$ indicates perfect overlap while $BC(P,Q) = 0$ indicates no overlap

Fukunaga, Keinosuke, Introduction to statistical pattern recognition. Elsevier. (2013)

Saturn-Enceladus Distant Prograde Orbit Propagation

Non-Gaussian metric of comparison

Algorithm 1 Calculate Bhattacharyya Coefficient of (P, Q)

Inputs: $P, Q : \mathbb{R}^d \to [0, 1],\ \ n > 1$

```
bounds ← [∞ · 1_{d×1}, −∞ · 1_{d×1}]
grid ← 0_{d×n}                                                          ▷ finding bounding region
for i = 1 to d do
    bounds[i, 1] ← min {b[i, 1], min{x[i] ∈ support(P ∪ Q)}}
    bounds[i, 2] ← max {b[i, 2], max{x[i] ∈ support(P ∪ Q)}}
    grid[i] ← {linear set from bounds[i, 1] to bounds[i, 2] of size n}
end for

χ ← 0_{n^d×d}
P*, Q* ← 0_{n^d×1}
count ← 1                                                               ▷ discretizing P, Q over χ
for x_1 = 1 to n do
    ⋮
    for x_d = 1 to n do
        χ[count] ← {grid[1, x_1], . . . , grid[d, x_d]}
        P*[count] ← P(χ[count])
        Q*[count] ← Q(χ[count])
        count ← count + 1
    end for
    ⋮
end for
P* ← P*/sum(P*)
Q* ← Q*/sum(Q*)

BC ← 0                                                                  ▷ calculating BC(P*, Q*)
for i = 1 to n^d do
    BC ← BC + √(P*[i]Q*[i])
end for
```

Saturn-Enceladus Distant Prograde Orbit Propagation

Non-Gaussian metric of comparison

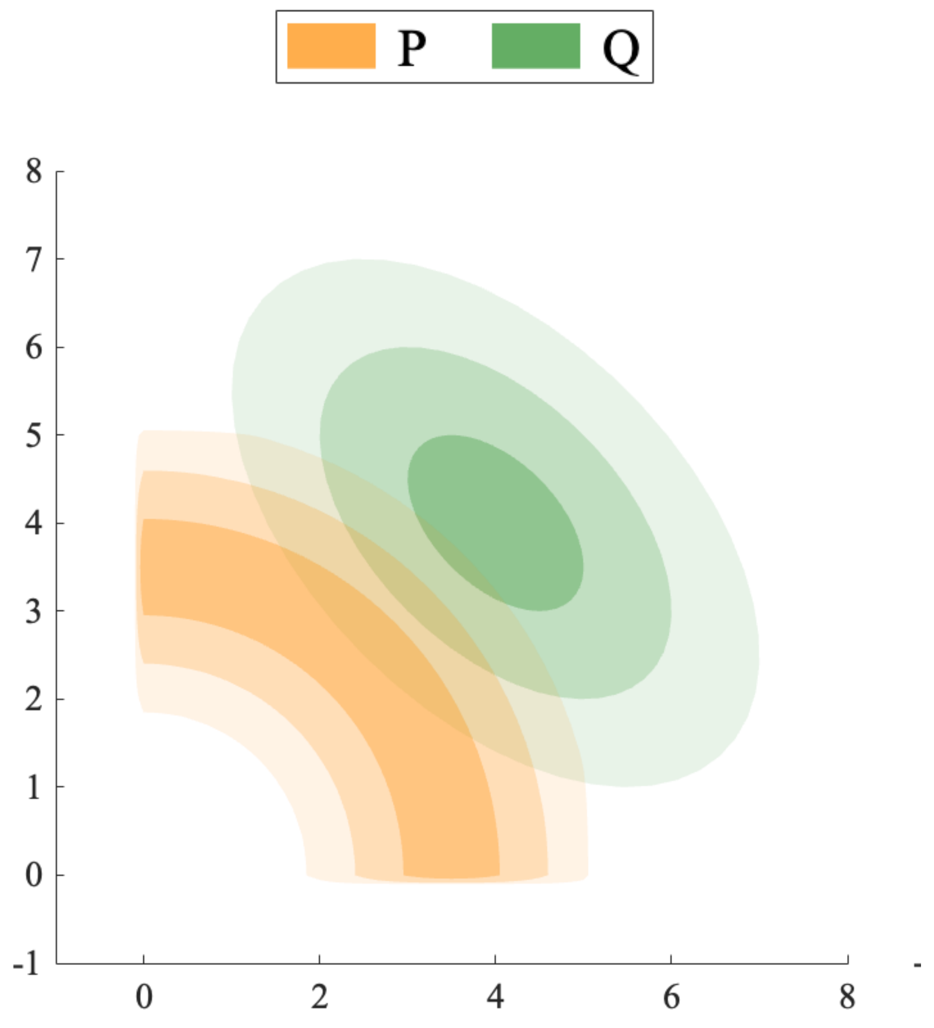
Algorithm 1 Calculate Bhattacharyya Coefficient of (P, Q)

Inputs: $P, Q : \mathbb{R}^d \to [0,1],\ n > 1$

bounds $\leftarrow [\infty \cdot \mathbf{1}_{d\times 1}, -\infty \cdot \mathbf{1}_{d\times 1}]$
grid $\leftarrow \mathbf{0}_{d\times n}$ ▷ **finding bounding region**
for $i = 1$ to d **do**
  bounds$[i,1] \leftarrow \min\{b[i,1], \min\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  bounds$[i,2] \leftarrow \max\{b[i,2], \max\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  grid$[i] \leftarrow$ {linear set from bounds$[i,1]$ to bounds$[i,2]$ of size n}
end for

$\chi \leftarrow \mathbf{0}_{n^d \times d}$
$P^*, Q^* \leftarrow \mathbf{0}_{n^d \times 1}$
count $\leftarrow 1$ ▷ **discretizing** P, Q **over** χ
for $x_1 = 1$ to n **do**
  ⋮
  for $x_d = 1$ to n **do**
    χ[count] $\leftarrow$ {grid$[1, x_1], \ldots,$ grid$[d, x_d]$}
    P^*[count] $\leftarrow P(\chi$[count]$)$
    Q^*[count] $\leftarrow Q(\chi$[count]$)$
    count $\leftarrow$ count $+ 1$
  end for
  ⋮
end for
$P^* \leftarrow P^*/\text{sum}(P^*)$
$Q^* \leftarrow Q^*/\text{sum}(Q^*)$

$BC \leftarrow 0$ ▷ **calculating** $BC(P^*, Q^*)$
for $i = 1$ to n^d **do**
  $BC \leftarrow BC + \sqrt{P^*[i]Q^*[i]}$
end for



$n = 10$

Saturn-Enceladus Distant Prograde Orbit Propagation

Non-Gaussian metric of comparison

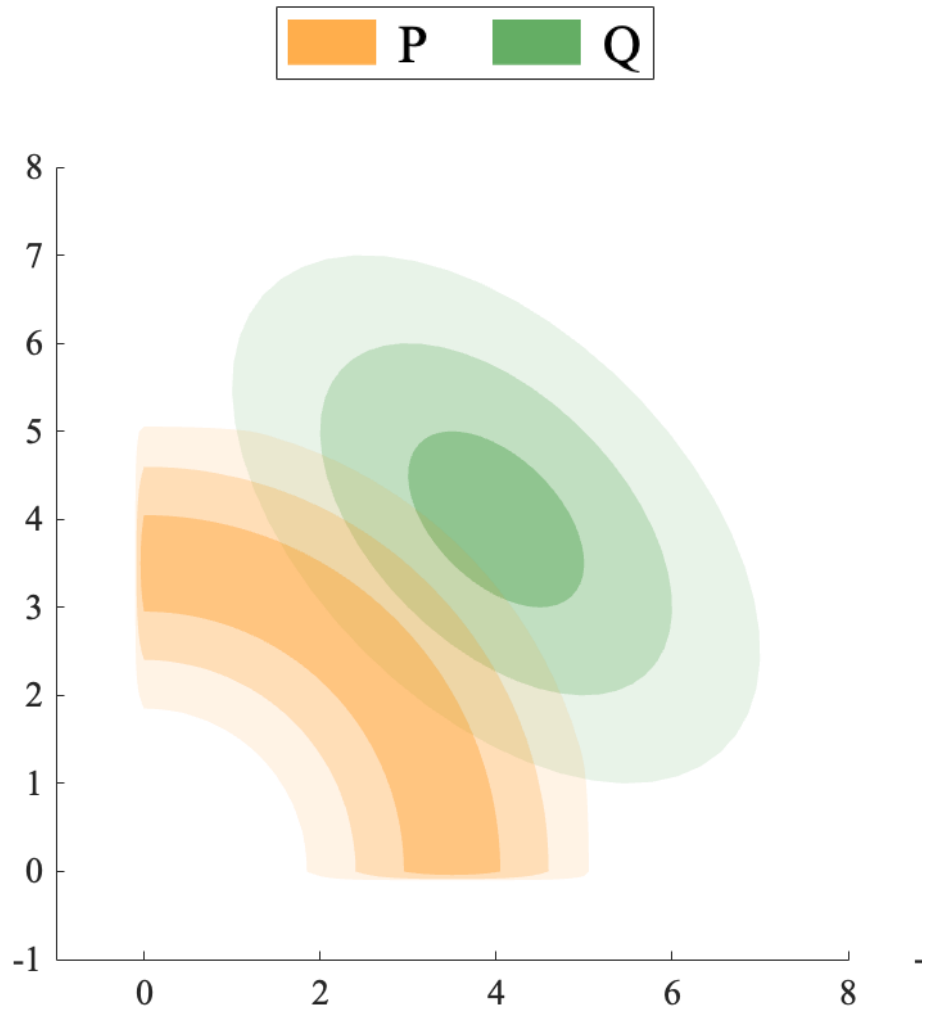
Algorithm 1 Calculate Bhattacharyya Coefficient of (P, Q)

Inputs: $P, Q : \mathbb{R}^d \to [0, 1], \; n > 1$

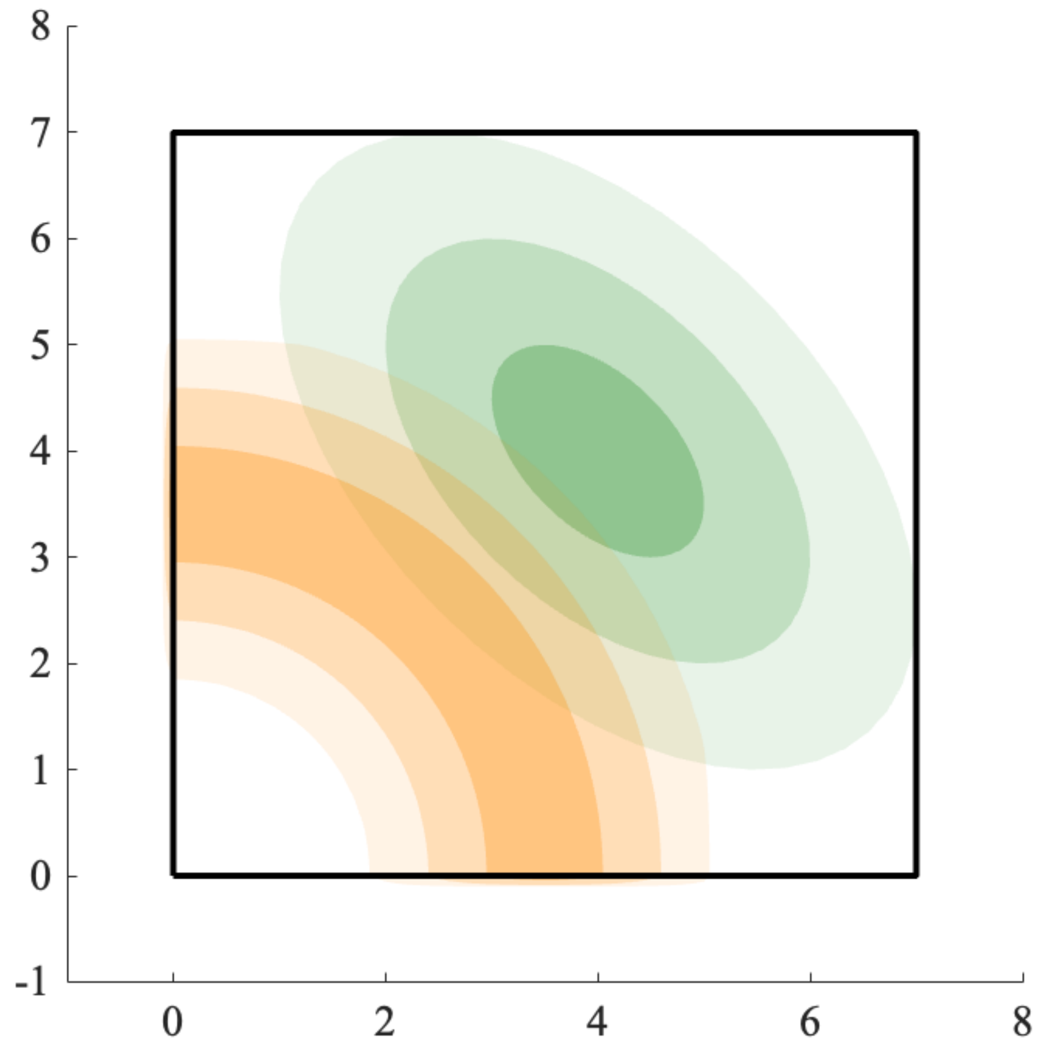
bounds $\leftarrow [\infty \cdot \mathbf{1}_{d\times 1}, -\infty \cdot \mathbf{1}_{d\times 1}]$
grid $\leftarrow \mathbf{0}_{d\times n}$ ▷ **finding bounding region**
for $i = 1$ to d **do**
  bounds$[i, 1] \leftarrow \min\{b[i, 1], \min\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  bounds$[i, 2] \leftarrow \max\{b[i, 2], \max\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  grid$[i] \leftarrow$ {linear set from bounds$[i, 1]$ to bounds$[i, 2]$ of size n}
end for

$\chi \leftarrow \mathbf{0}_{n^d \times d}$
$P^*, Q^* \leftarrow \mathbf{0}_{n^d \times 1}$
count $\leftarrow 1$ ▷ **discretizing P, Q over χ**
for $x_1 = 1$ to n **do**
  ⋮
  for $x_d = 1$ to n **do**
    χ[count] $\leftarrow$ {grid$[1, x_1], \ldots,$ grid$[d, x_d]$}
    P^*[count] $\leftarrow P(\chi$[count]$)$
    Q^*[count] $\leftarrow Q(\chi$[count]$)$
    count $\leftarrow$ count $+ 1$
  end for
  ⋮
end for
$P^* \leftarrow P^*/\text{sum}(P^*)$
$Q^* \leftarrow Q^*/\text{sum}(Q^*)$

$BC \leftarrow 0$ ▷ **calculating $BC(P^*, Q^*)$**
for $i = 1$ to n^d **do**
  $BC \leftarrow BC + \sqrt{P^*[i]Q^*[i]}$
end for



$n = 10$



Saturn-Enceladus Distant Prograde Orbit Propagation

Non-Gaussian metric of comparison

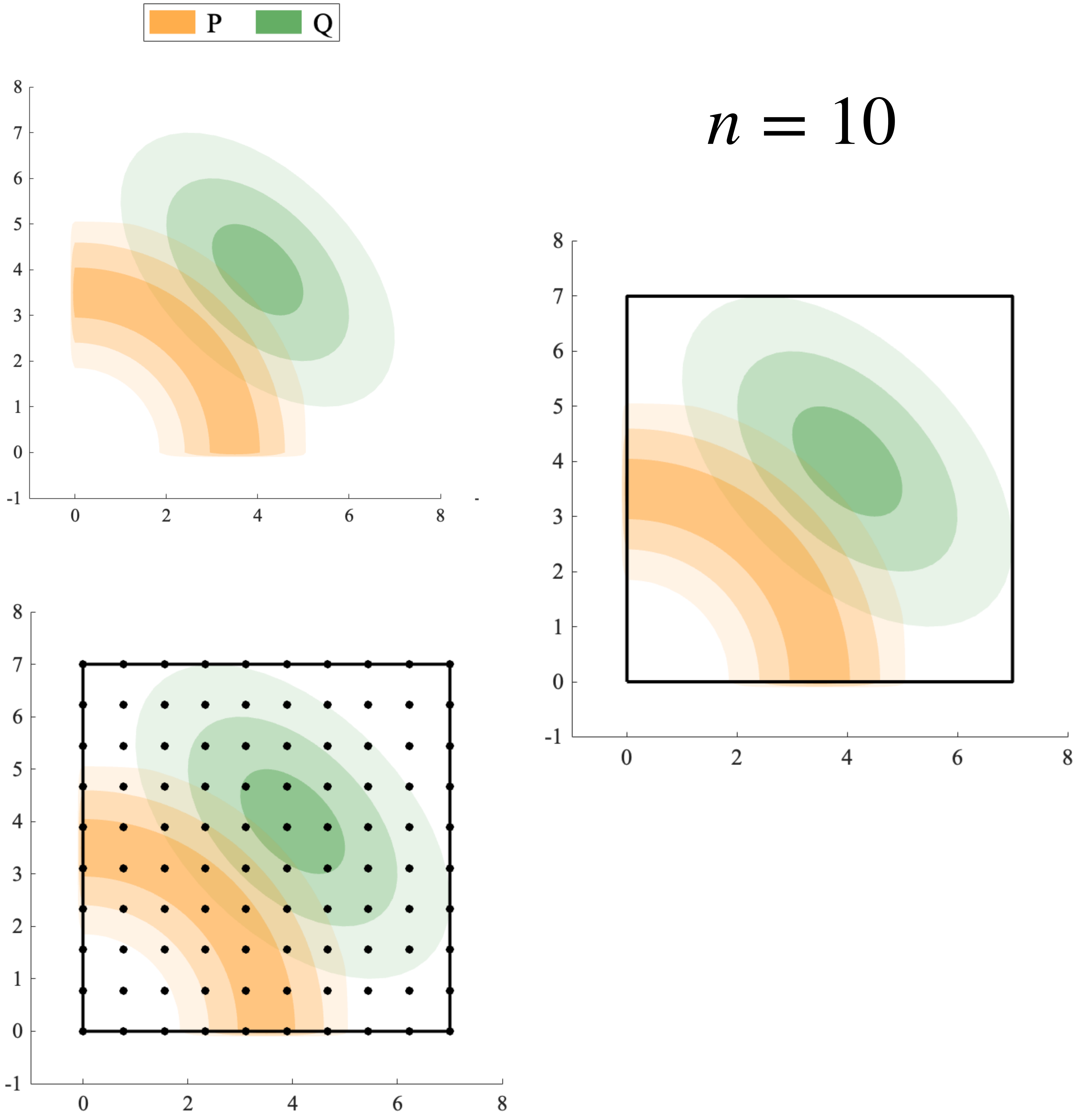
Algorithm 1 Calculate Bhattacharyya Coefficient of (P, Q)

Inputs: $P, Q : \mathbb{R}^d \to [0, 1], \;\; n > 1$

bounds $\leftarrow [\infty \cdot \mathbf{1}_{d\times 1}, -\infty \cdot \mathbf{1}_{d\times 1}]$
grid $\leftarrow \mathbf{0}_{d\times n}$ ▷ **finding bounding region**
for $i = 1$ to d **do**
  bounds$[i, 1] \leftarrow \min\{b[i, 1], \min\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  bounds$[i, 2] \leftarrow \max\{b[i, 2], \max\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  grid$[i] \leftarrow$ {linear set from bounds$[i, 1]$ to bounds$[i, 2]$ of size n}
end for

$\chi \leftarrow \mathbf{0}_{n^d\times d}$
$P^*, Q^* \leftarrow \mathbf{0}_{n^d\times 1}$
count $\leftarrow 1$ ▷ **discretizing** P, Q **over** χ
for $x_1 = 1$ to n **do**
  ⋮
  for $x_d = 1$ to n **do**
    χ[count] $\leftarrow$ {grid$[1, x_1], \ldots,$ grid$[d, x_d]$}
    P^*[count] $\leftarrow P(\chi$[count]$)$
    Q^*[count] $\leftarrow Q(\chi$[count]$)$
    count $\leftarrow$ count $+ 1$
  end for
  ⋮
end for
$P^* \leftarrow P^*/\text{sum}(P^*)$
$Q^* \leftarrow Q^*/\text{sum}(Q^*)$

$BC \leftarrow 0$ ▷ **calculating** $BC(P^*, Q^*)$
for $i = 1$ to n^d **do**
  $BC \leftarrow BC + \sqrt{P^*[i]Q^*[i]}$
end for



Saturn-Enceladus Distant Prograde Orbit Propagation

Non-Gaussian metric of comparison

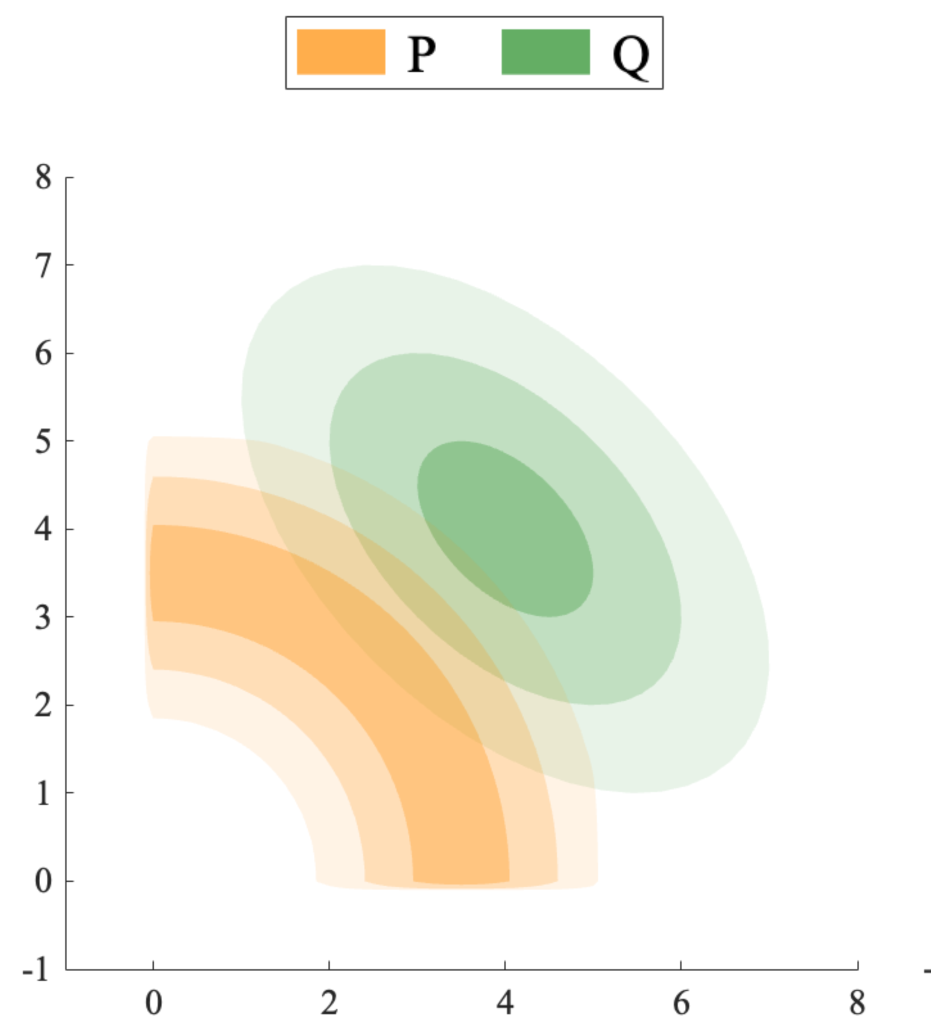
Algorithm 1 Calculate Bhattacharyya Coefficient of (P, Q)

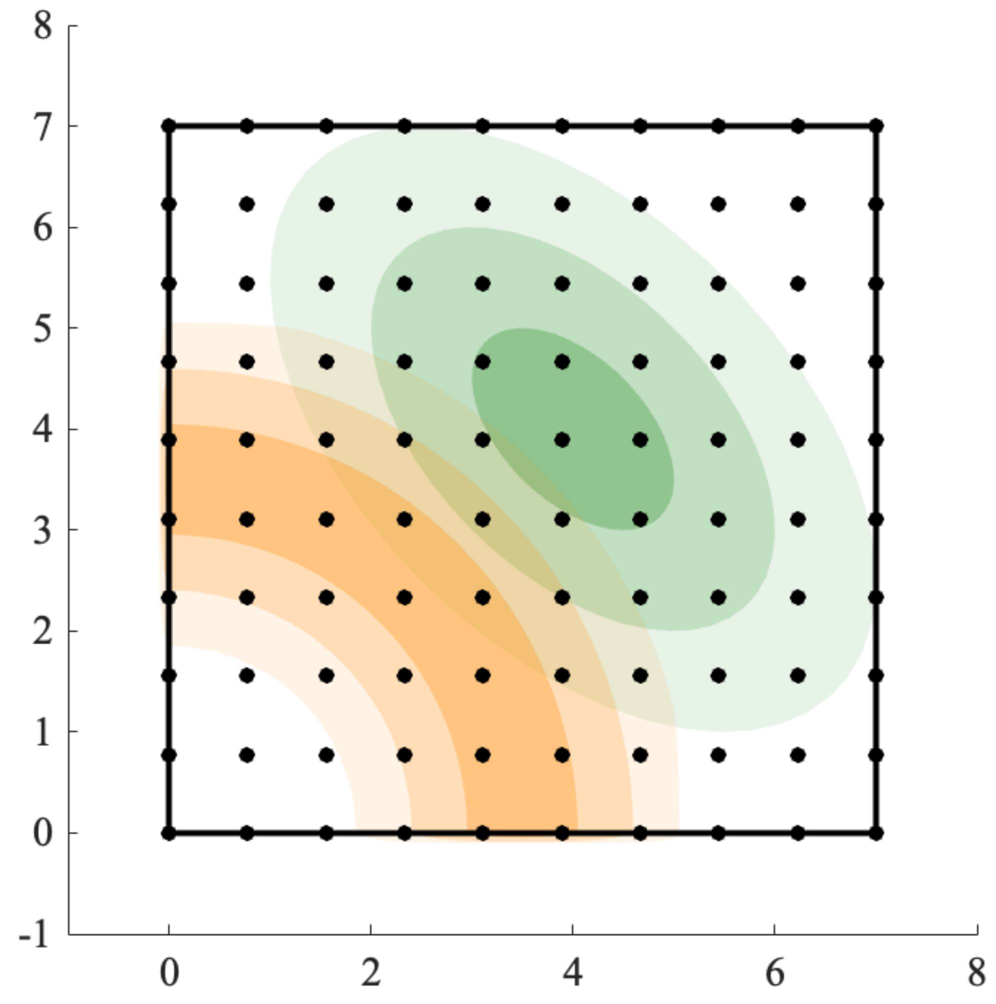
Inputs: $P, Q : \mathbb{R}^d \to [0,1],\ n > 1$

bounds $\leftarrow [\infty \cdot \mathbf{1}_{d\times 1}, -\infty \cdot \mathbf{1}_{d\times 1}]$
grid $\leftarrow \mathbf{0}_{d\times n}$ ▷ **finding bounding region**
for $i = 1$ to d **do**
  bounds$[i,1] \leftarrow \min\{b[i,1], \min\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  bounds$[i,2] \leftarrow \max\{b[i,2], \max\{\boldsymbol{x}[i] \in \text{support}(P \cup Q)\}\}$
  grid$[i] \leftarrow$ {linear set from bounds$[i,1]$ to bounds$[i,2]$ of size n}
end for

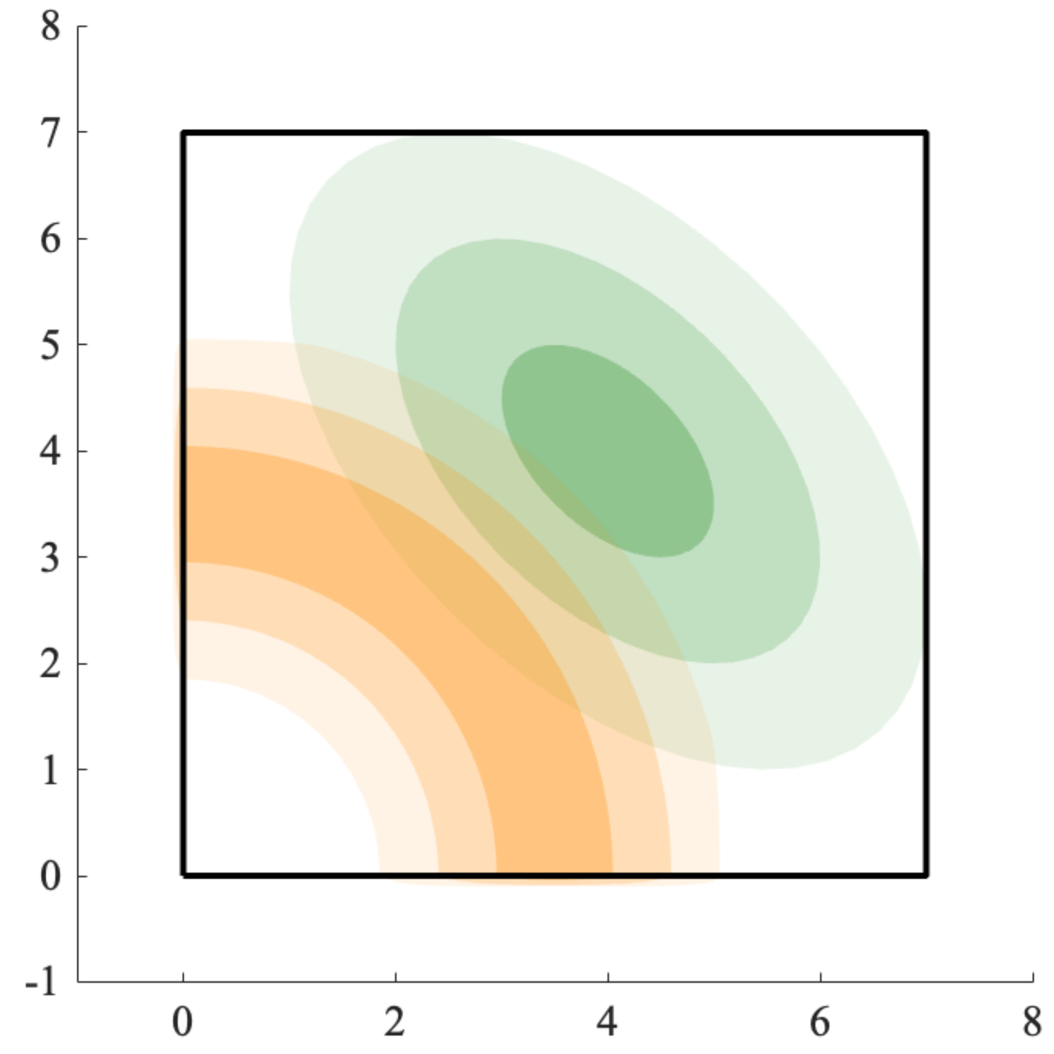
$\chi \leftarrow \mathbf{0}_{n^d \times d}$
$P^*, Q^* \leftarrow \mathbf{0}_{n^d \times 1}$
count $\leftarrow 1$ ▷ **discretizing** $P,\ Q$ **over** χ
for $x_1 = 1$ to n **do**
  ⋮
  for $x_d = 1$ to n **do**
    χ[count] $\leftarrow$ {grid$[1, x_1], \ldots,$ grid$[d, x_d]$}
    P^*[count] $\leftarrow P(\chi$[count]$)$
    Q^*[count] $\leftarrow Q(\chi$[count]$)$
    count $\leftarrow$ count + 1
  end for
  ⋮
end for
$P^* \leftarrow P^*/\text{sum}(P^*)$
$Q^* \leftarrow Q^*/\text{sum}(Q^*)$

$BC \leftarrow 0$ ▷ **calculating** $BC(P^*, Q^*)$
for $i = 1$ to n^d **do**
  $BC \leftarrow BC + \sqrt{P^*[i]Q^*[i]}$
end for





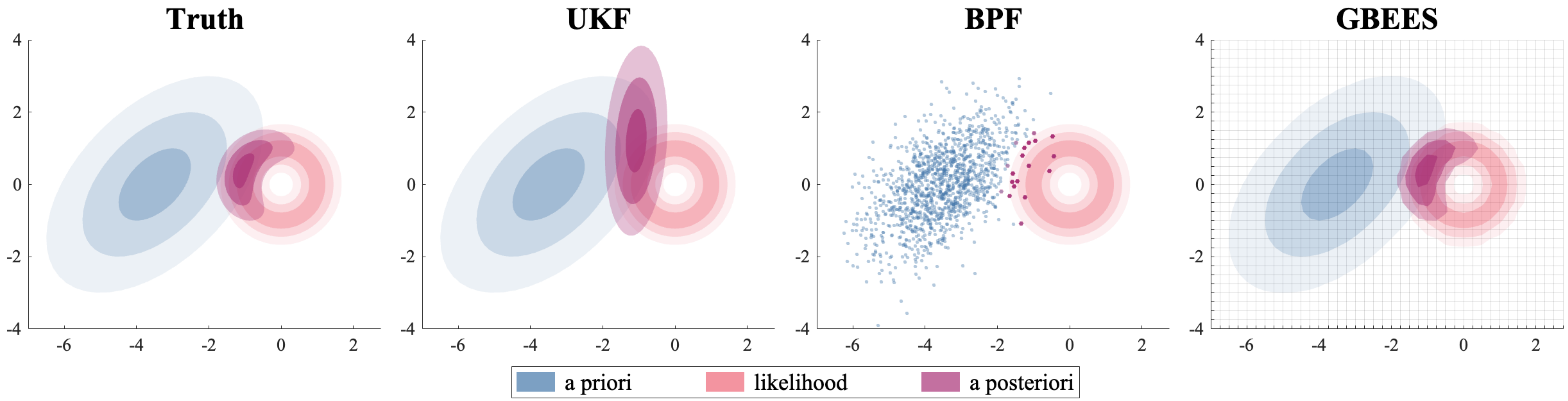
$n = 10$



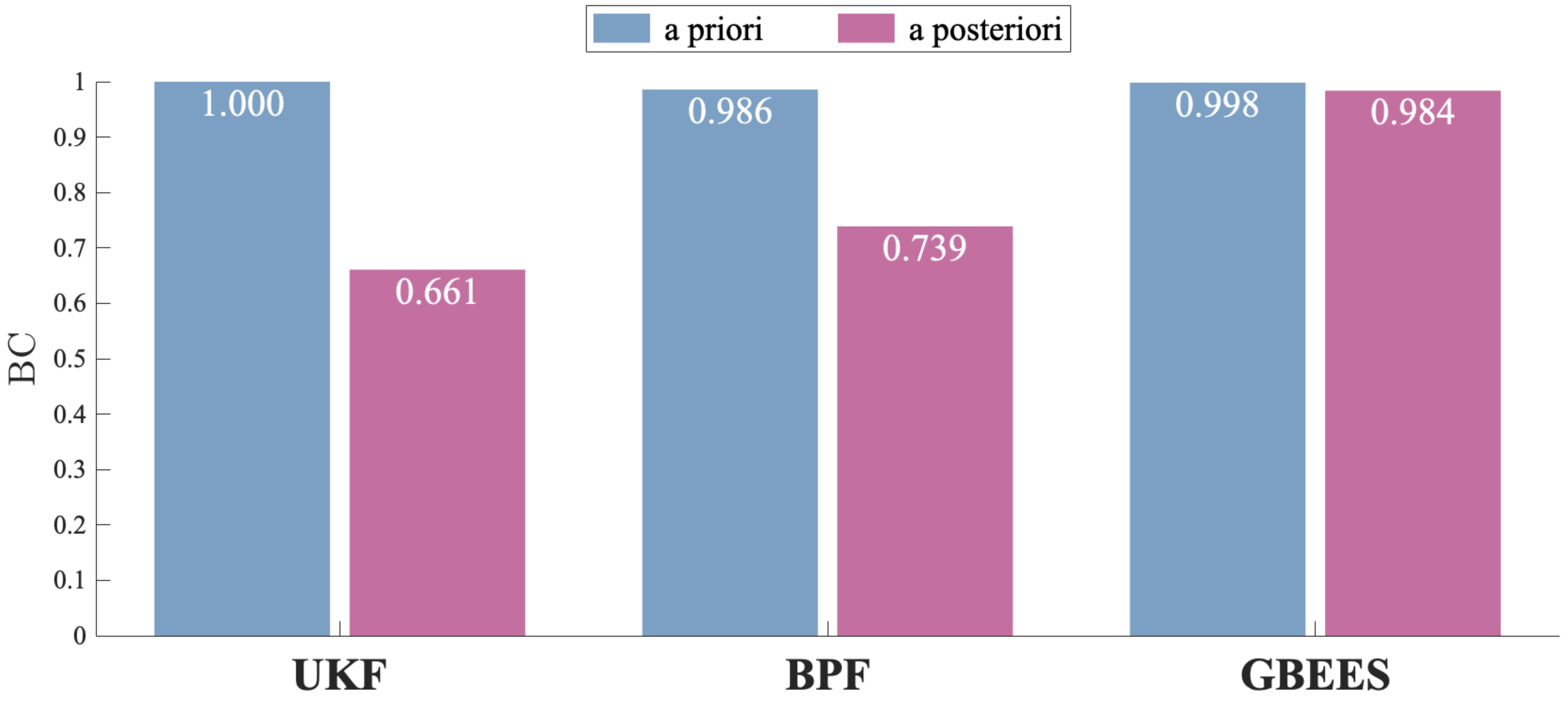
$BC(P, Q) = 0.1762$

Saturn-Enceladus Distant Prograde Orbit Propagation

Non-Gaussian metric of comparison

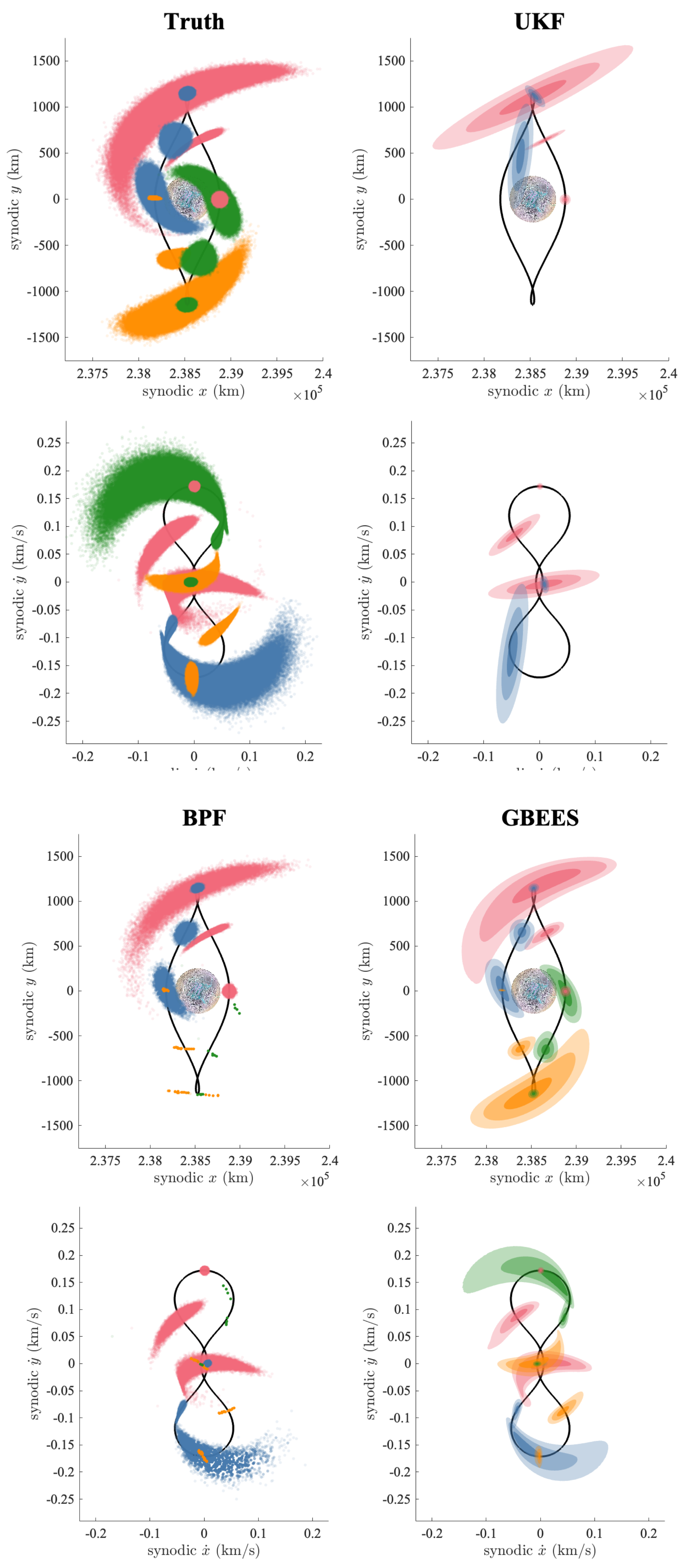


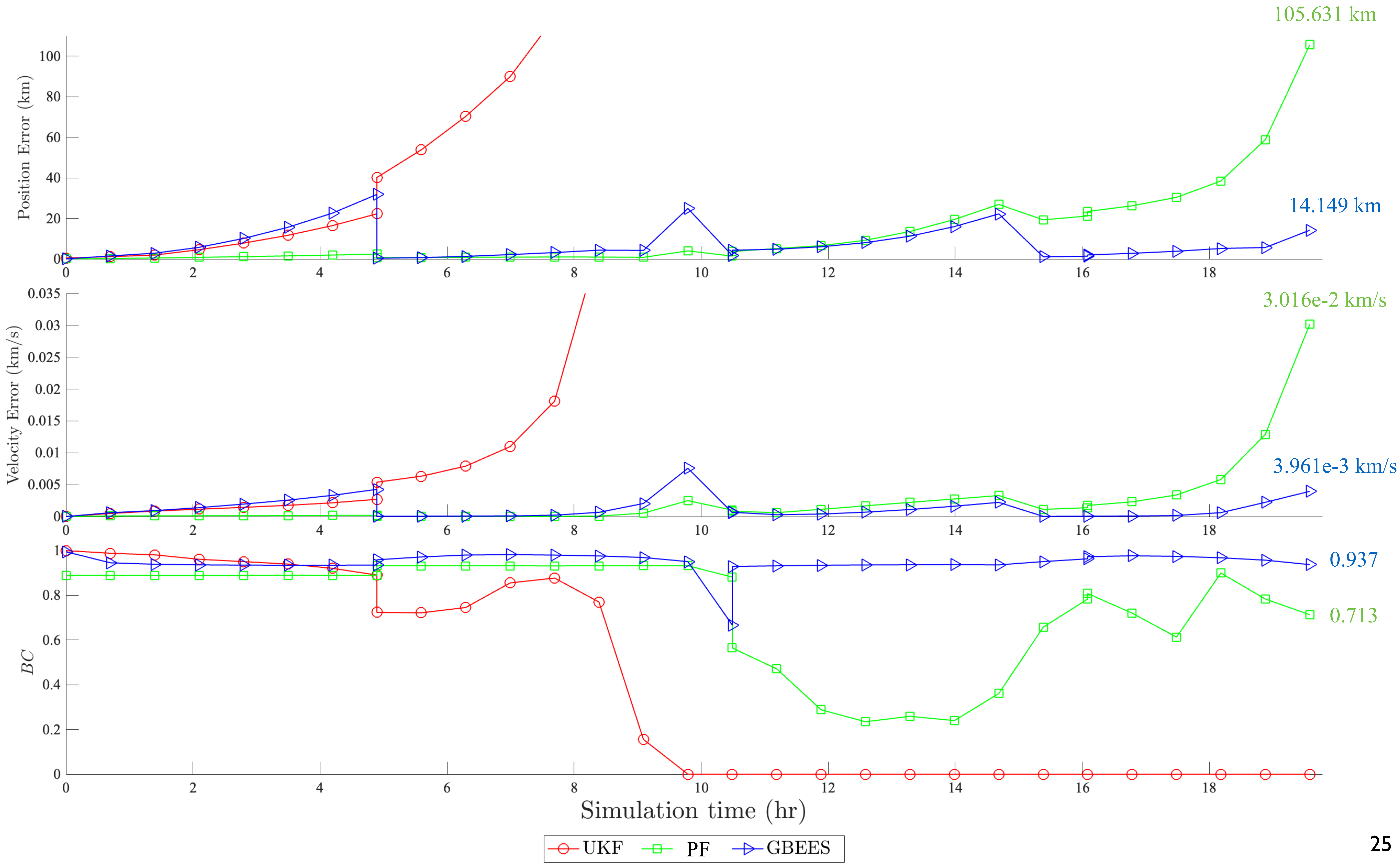
- As expected, GBEES performs best in this 2D test problem
- The GBEES update step is just the truth update step, just at a lower refinement



Saturn-Enceladus Distant Prograde Orbit Propagation

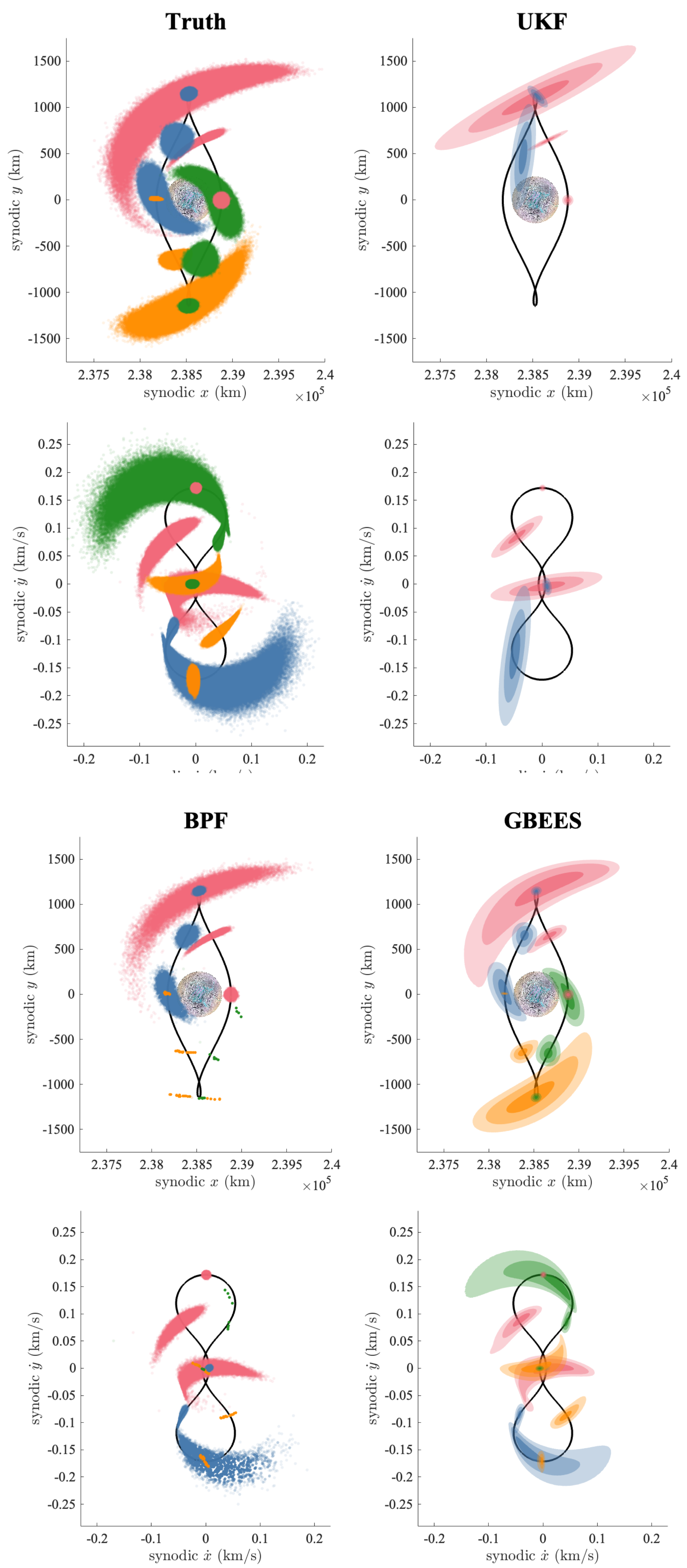
Quantitative comparison

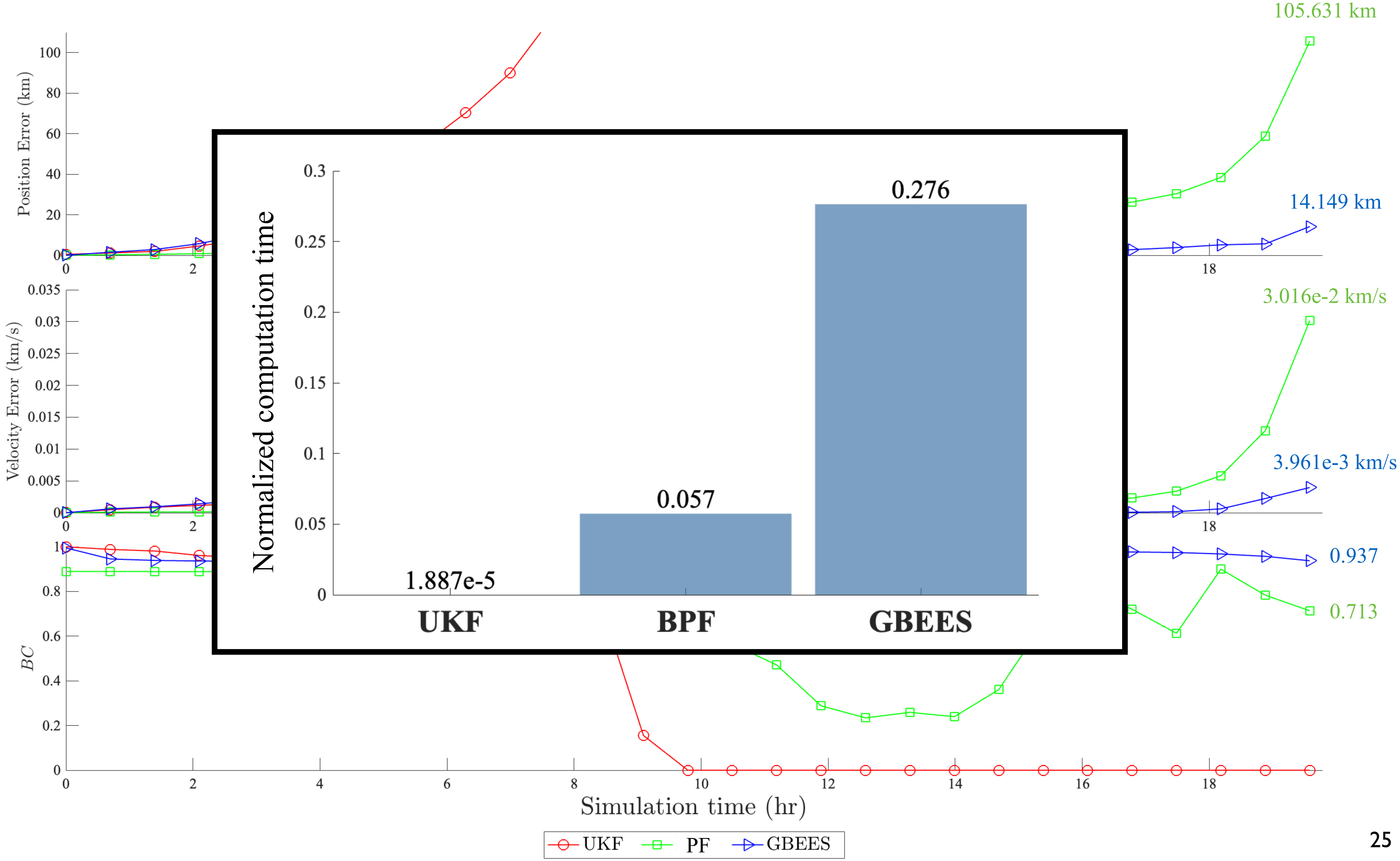




Saturn-Enceladus Distant Prograde Orbit Propagation

Quantitative comparison





Goal: Embed Monte within GBEES for ephemeris-quality orbital uncertainty propagation

Monte Python Wrapper

- For computational efficiency, GBEES runs in C — embedding Monte within GBEES requires a **Python wrapper**

GBEES

filetype = .c

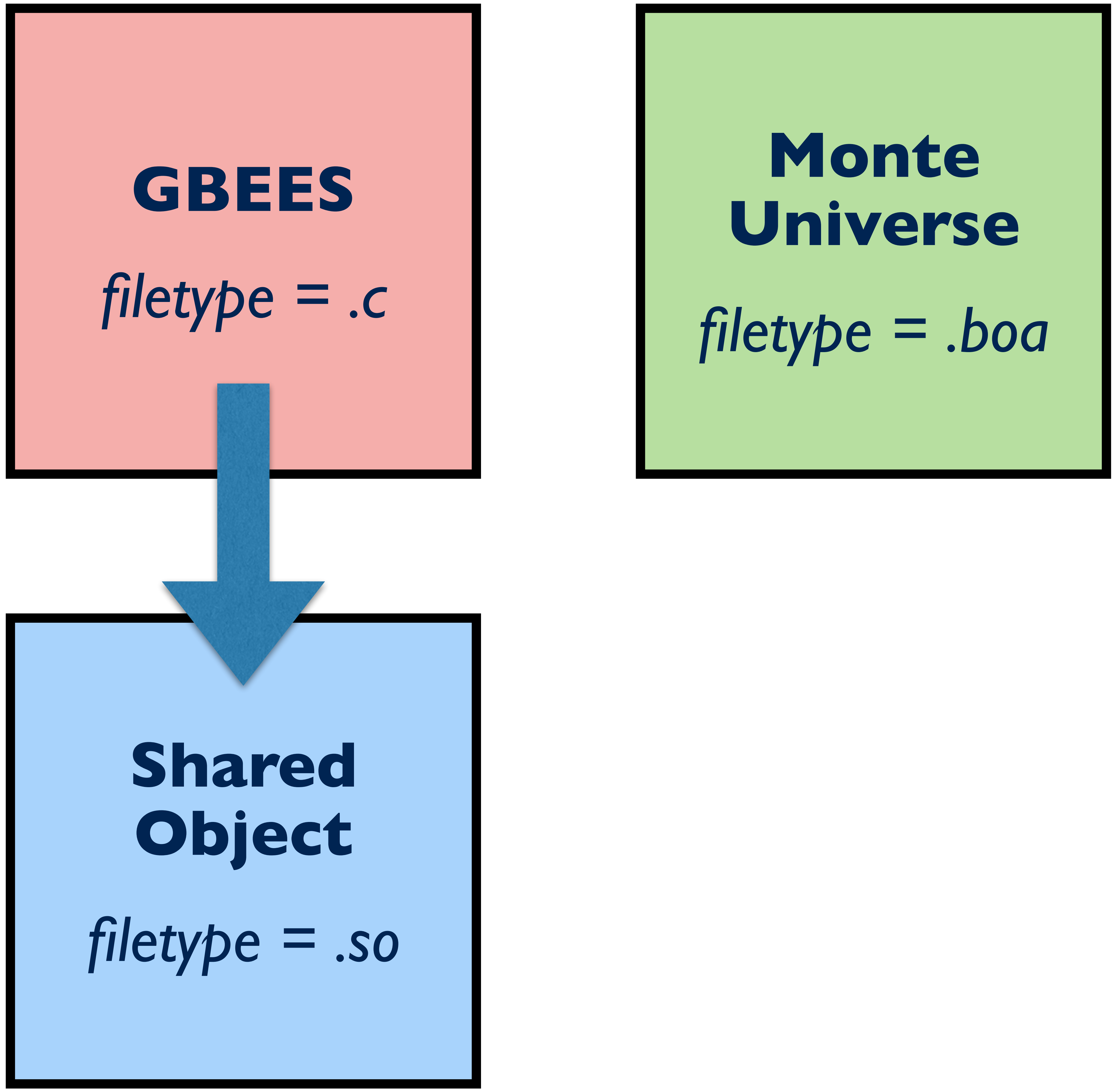
Monte Universe

filetype = .boa

<u>Order of Operations</u>

Monte Python Wrapper

- For computational efficiency, GBEES runs in C — embedding Monte within GBEES requires a **Python wrapper**



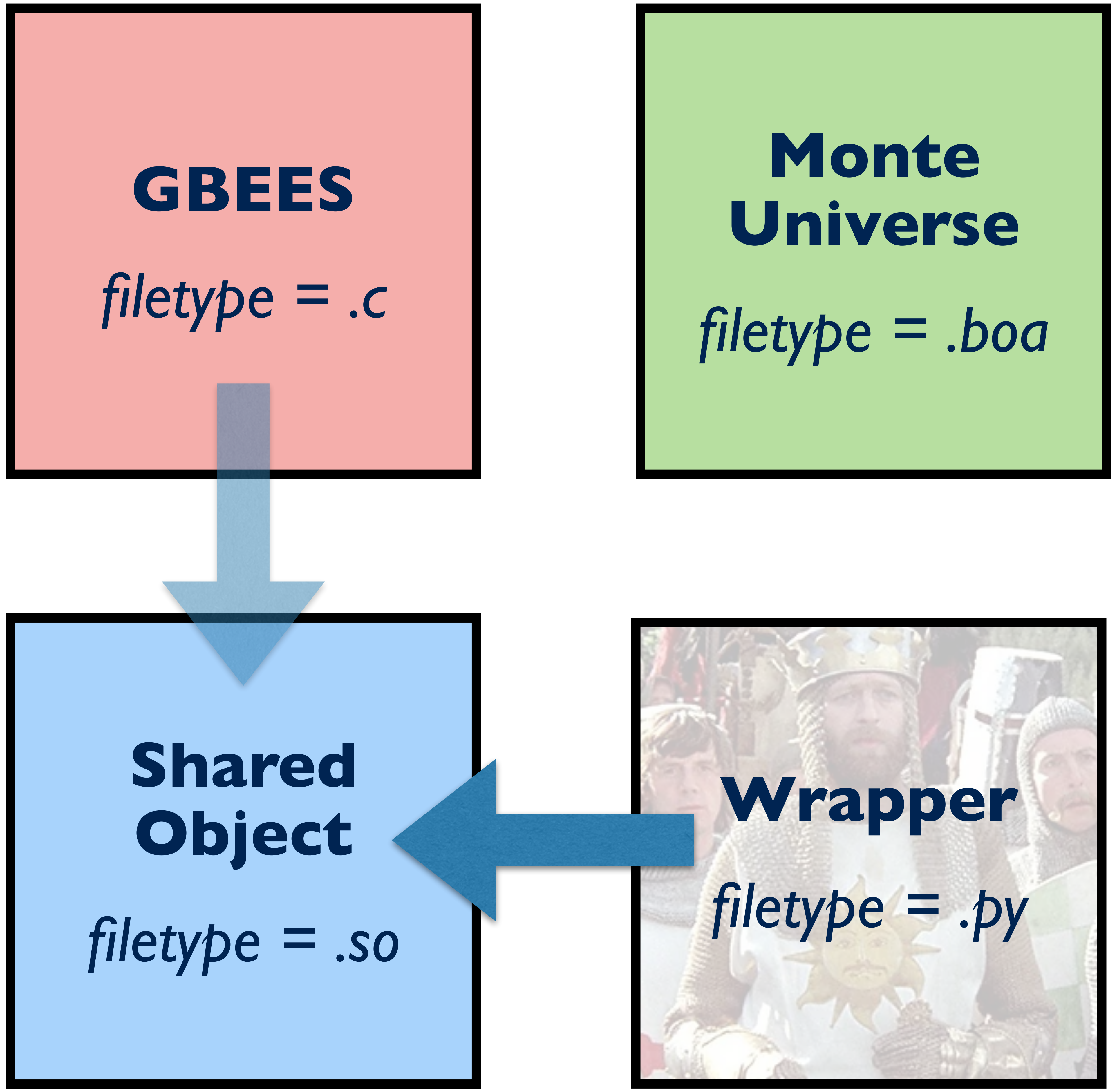
Order of Operations

1. Compile *GBEES.c* with Monte to a shared object

```
$ mdock run gcc -shared -o GBEES.so GBEES.c
```

Monte Python Wrapper

- For computational efficiency, GBEES runs in C — embedding Monte within GBEES requires a **Python wrapper**



Order of Operations

1. Compile *GBEES.c* with Monte to a shared object

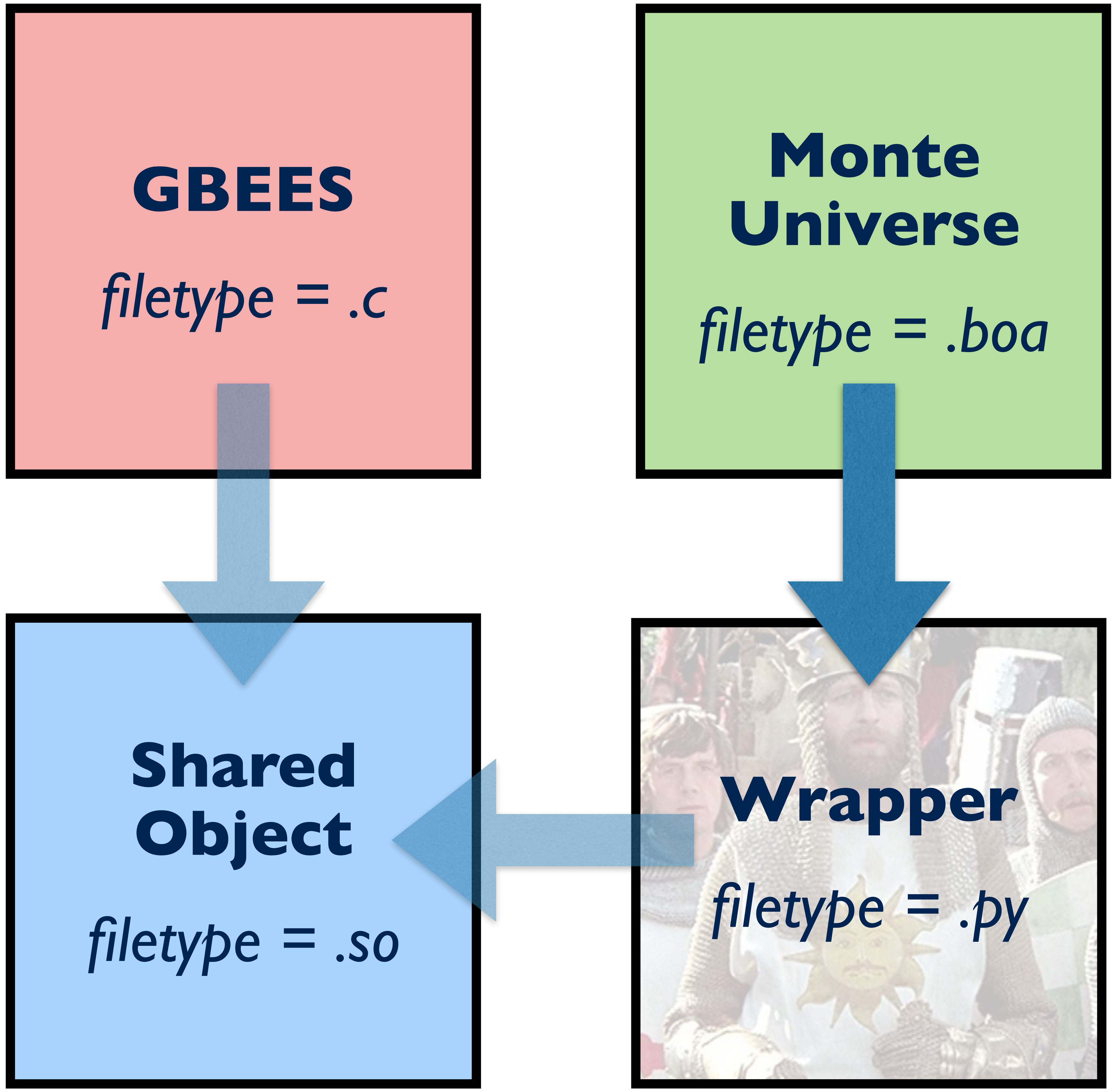
```
$ mdock run gcc -shared -o GBEES.so GBEES.c
```

2. Dynamically link *Wrapper.py* to *GBEES.so*

```
>> lib = ctypes.CDLL(“GBEES.so")
```

Monte Python Wrapper

- For computational efficiency, GBEES runs in C — embedding Monte within GBEES requires a **Python wrapper**



Order of Operations

1. Compile *GBEES.c* with Monte to a shared object

```
$ mdock run gcc -shared -o GBEES.so GBEES.c
```

2. Dynamically link *Wrapper.py* to *GBEES.so*

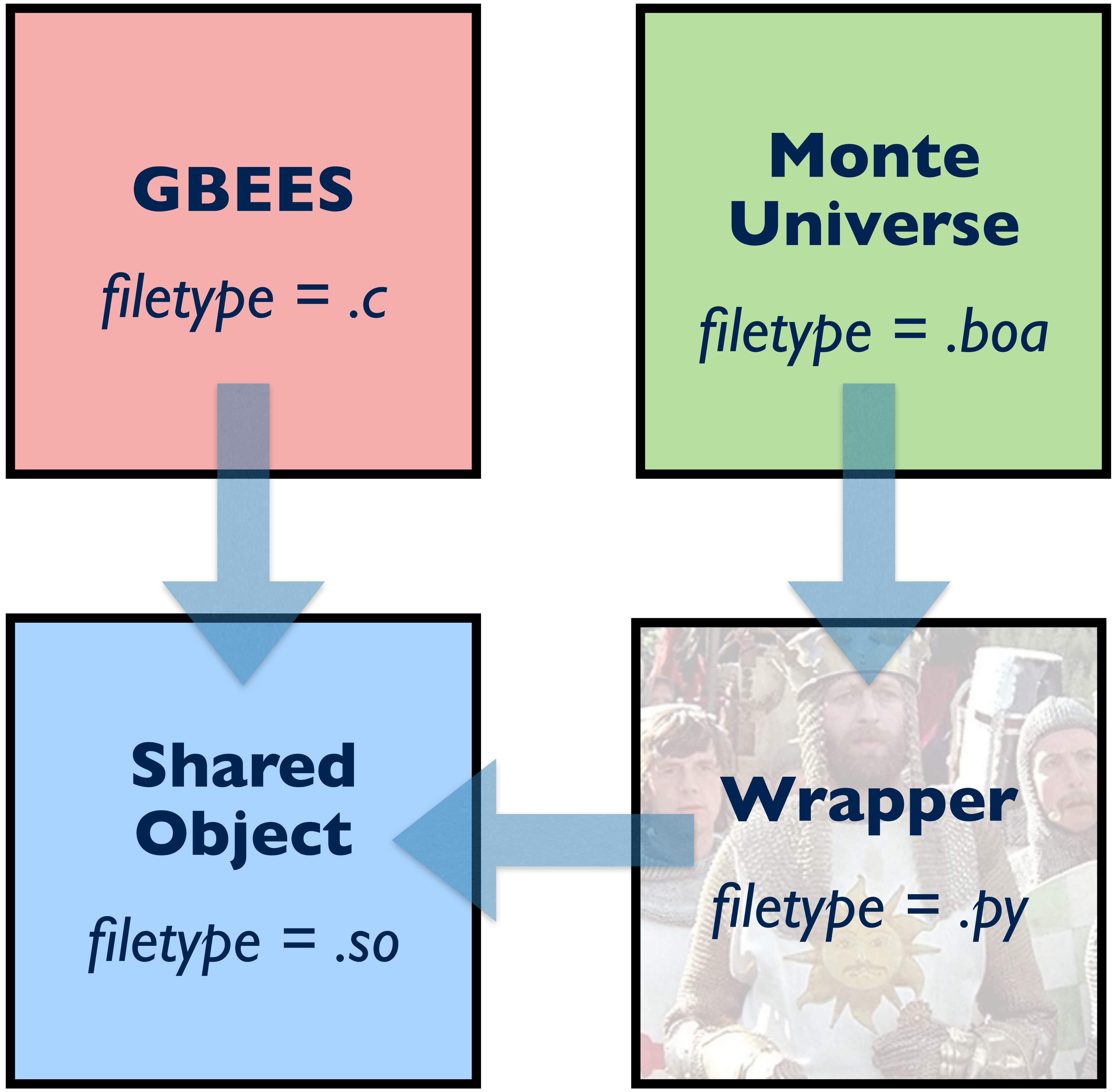
```
>> lib = ctypes.CDLL("GBEES.so")
```

3. Pass *MonteUniverse.boa* to *Wrapper.py*

```
>> boa = Monte.BoaLoad("MonteUniverse.boa")
```

Monte Python Wrapper

- For computational efficiency, GBEES runs in C — embedding Monte within GBEES requires a **Python wrapper**



Order of Operations

1. Compile *GBEES.c* with Monte to a shared object

```
$ mdock run gcc -shared -o GBEES.so GBEES.c
```

2. Dynamically link *Wrapper.py* to *GBEES.so*

```
>> lib = ctypes.CDLL(“GBEES.so")
```

3. Pass *MonteUniverse.boa* to *Wrapper.py*

```
>> boa = Monte.BoaLoad(“MonteUniverse.boa”)
```

4. Run GBEES with Monte by passing the *.boa* to the linked library

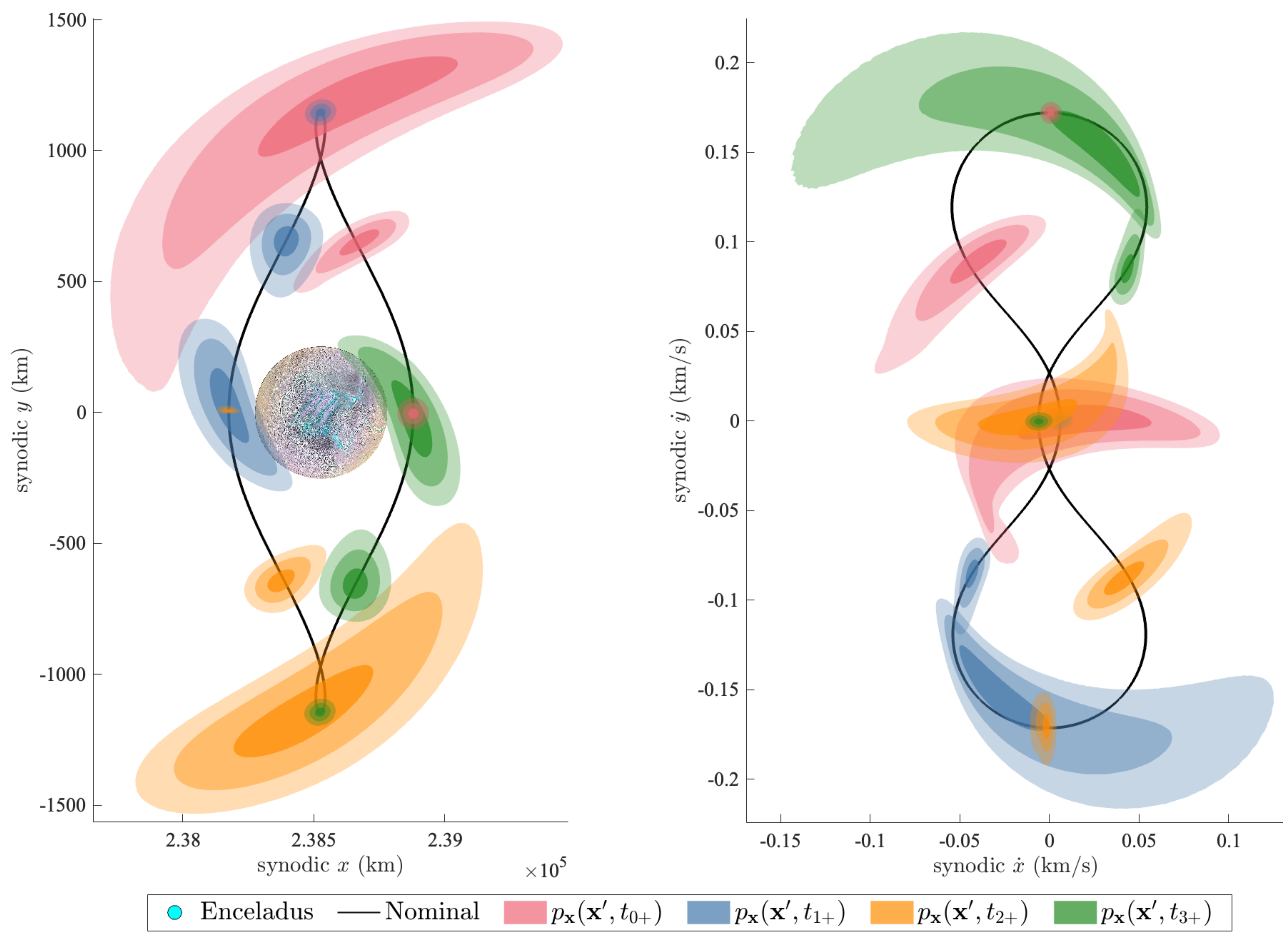
```
>> lib.run_gbees(boa)
```

Saturn-Enceladus Distant Prograde Orbit Propagation

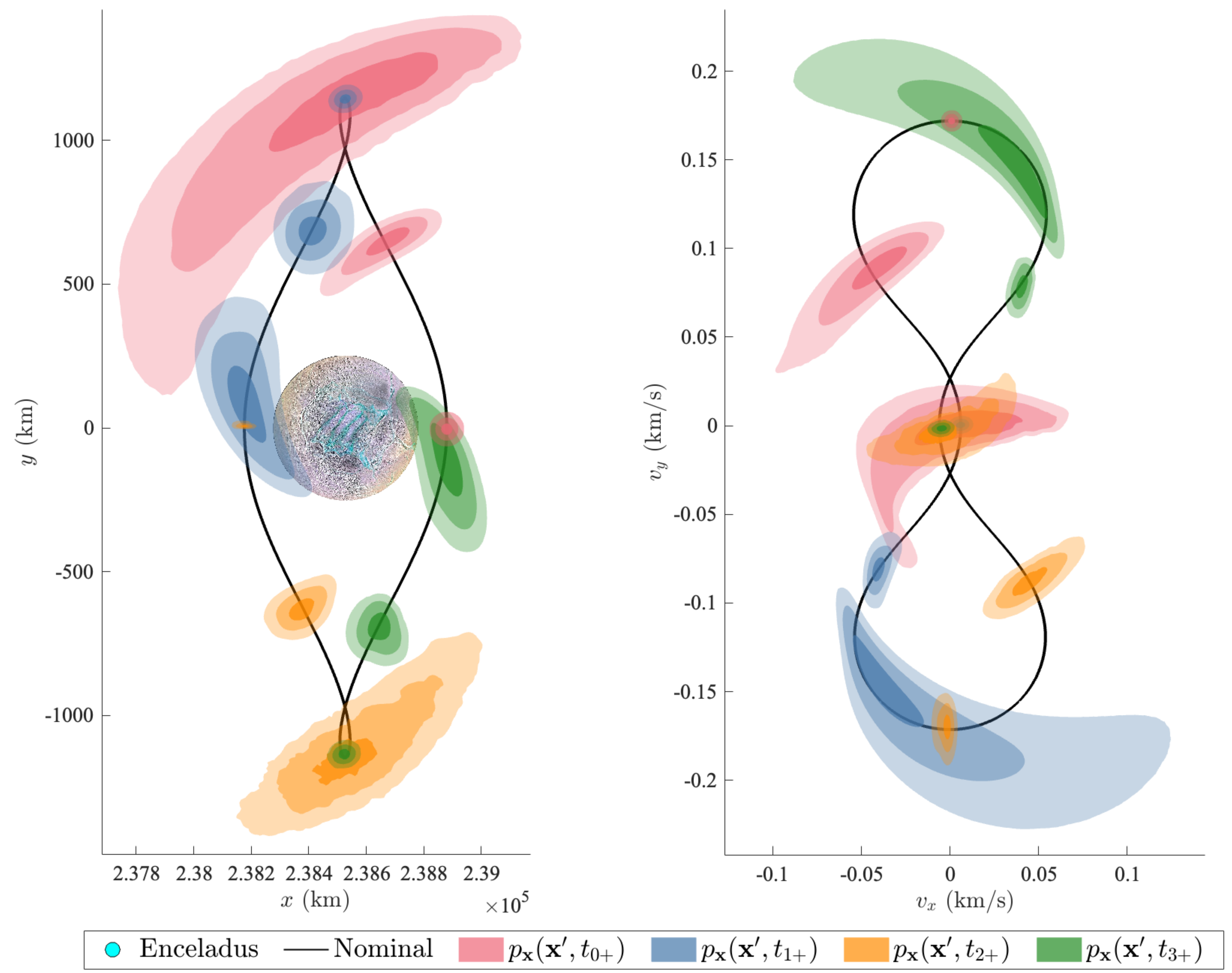
Monte vs. Analytical comparison - accuracy

- We compare the **PDFs** when propagating GBEES with the analytical solution to the CR3BP vs. when propagating GBEES with dynamics sourced from Monte

Analytical



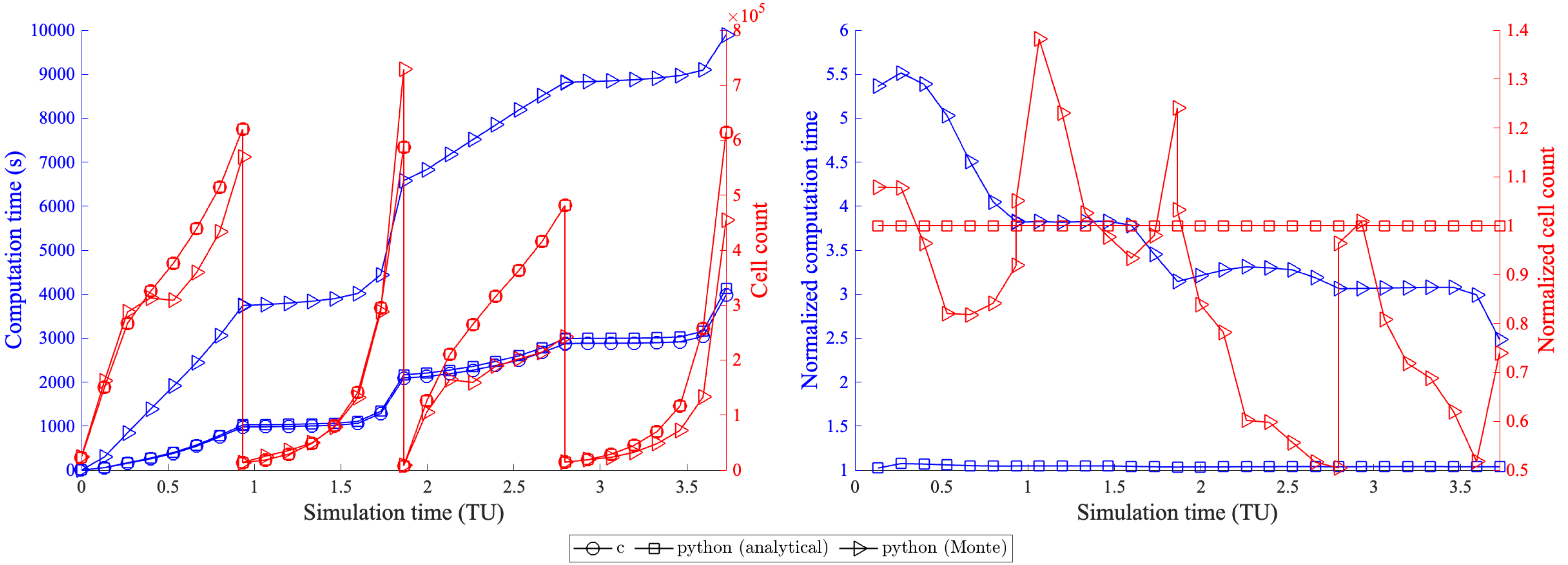
Monte



Saturn-Enceladus Distant Prograde Orbit Propagation

Monte vs. Analytical comparison - efficiency

- We compare the **computation time** when propagating GBEES with the analytical solution to the CR3BP vs. when propagating GBEES with dynamics sourced from Monte

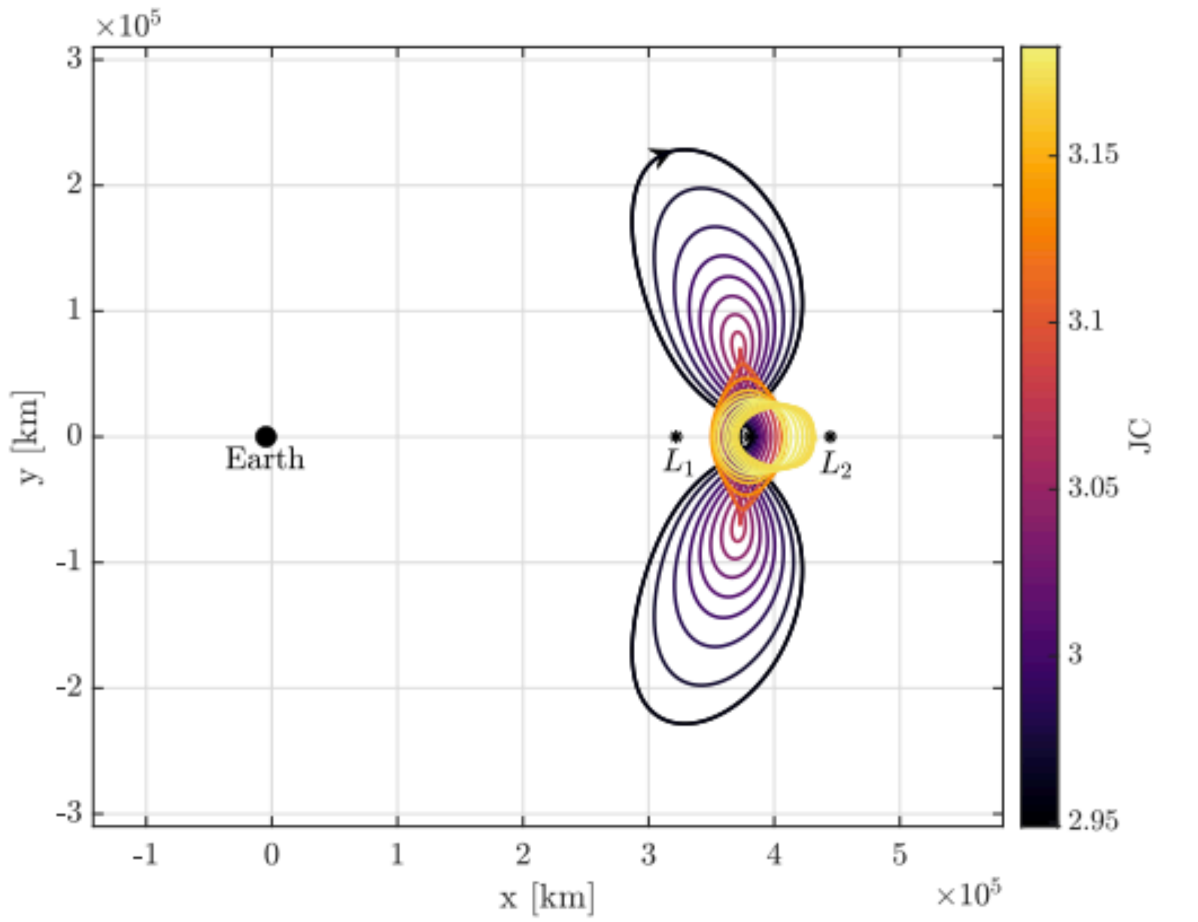


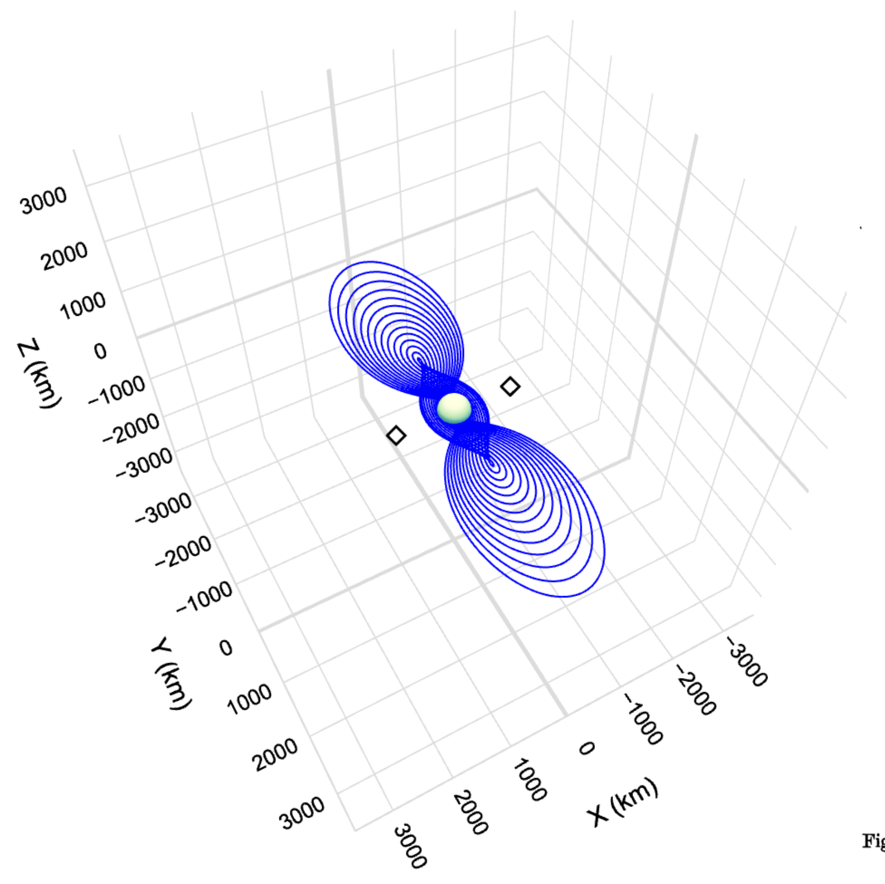
- There are still some bugs to fix here!

Conclusions

Conclusions

- There exist **favorable trajectories** in deep space where uncertainty may **realistically** become non-Gaussian





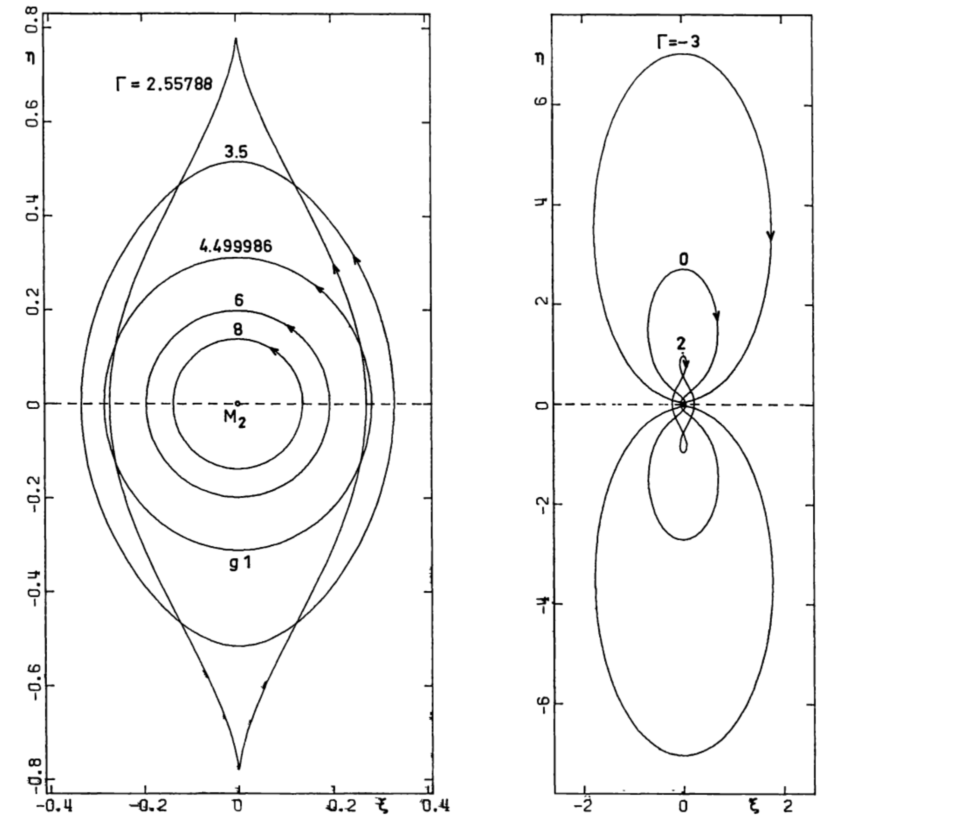
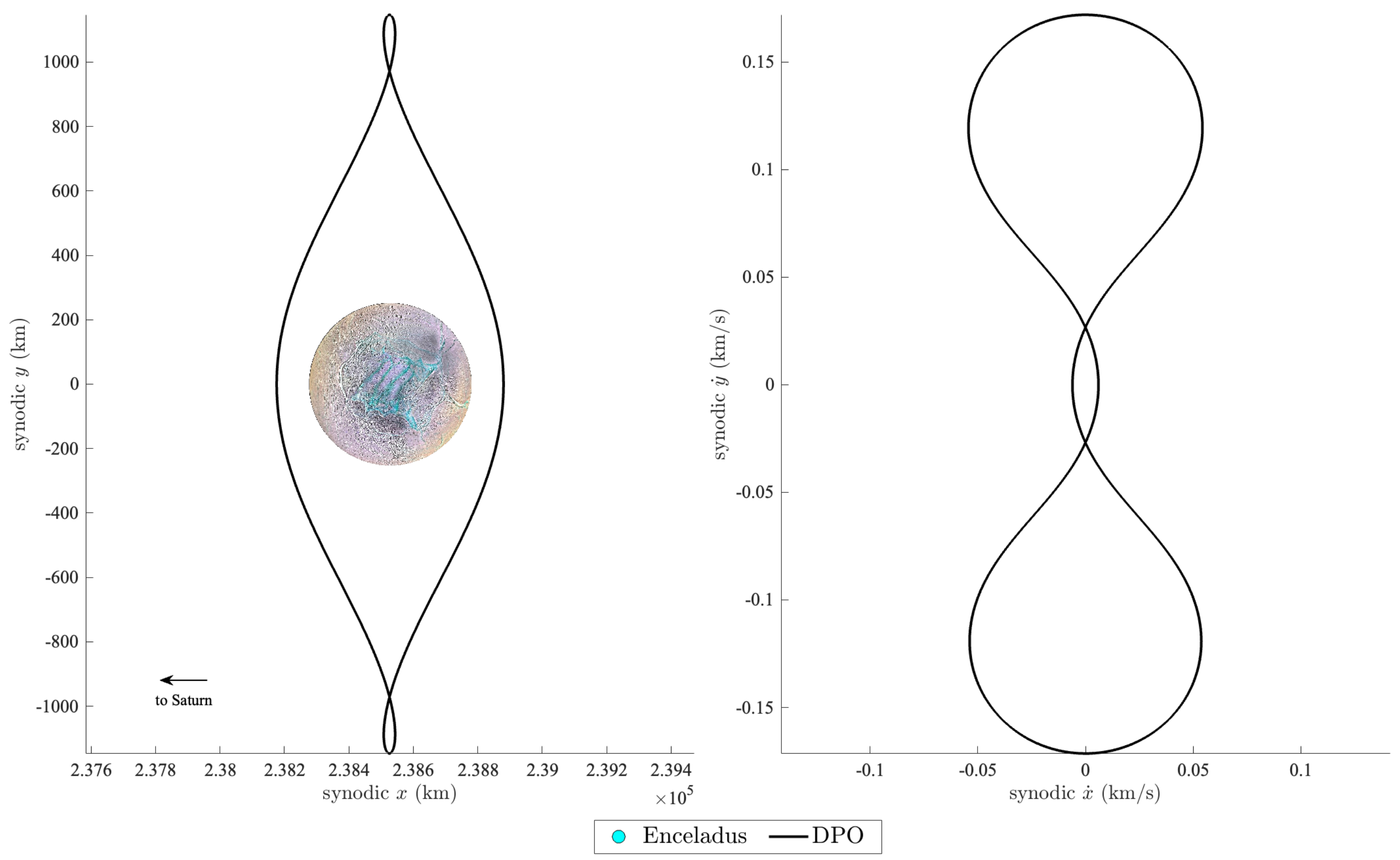
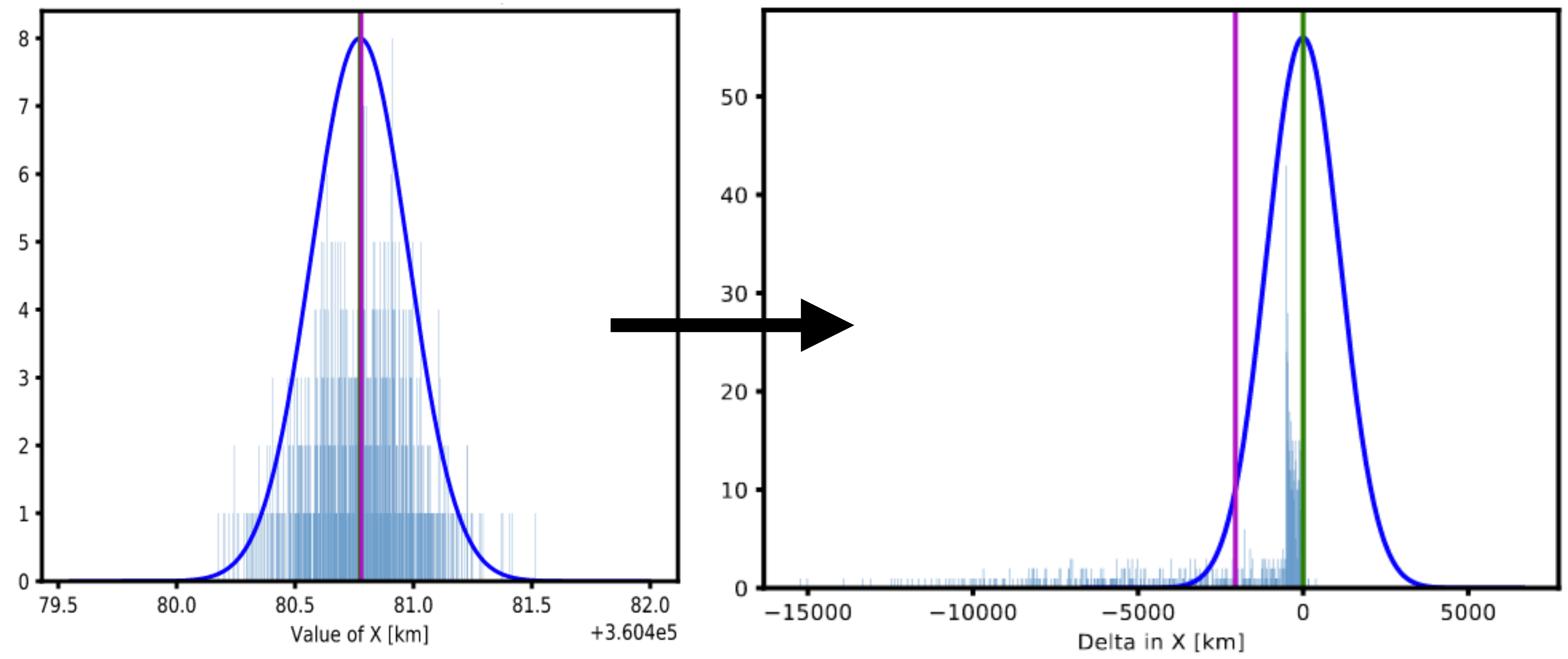
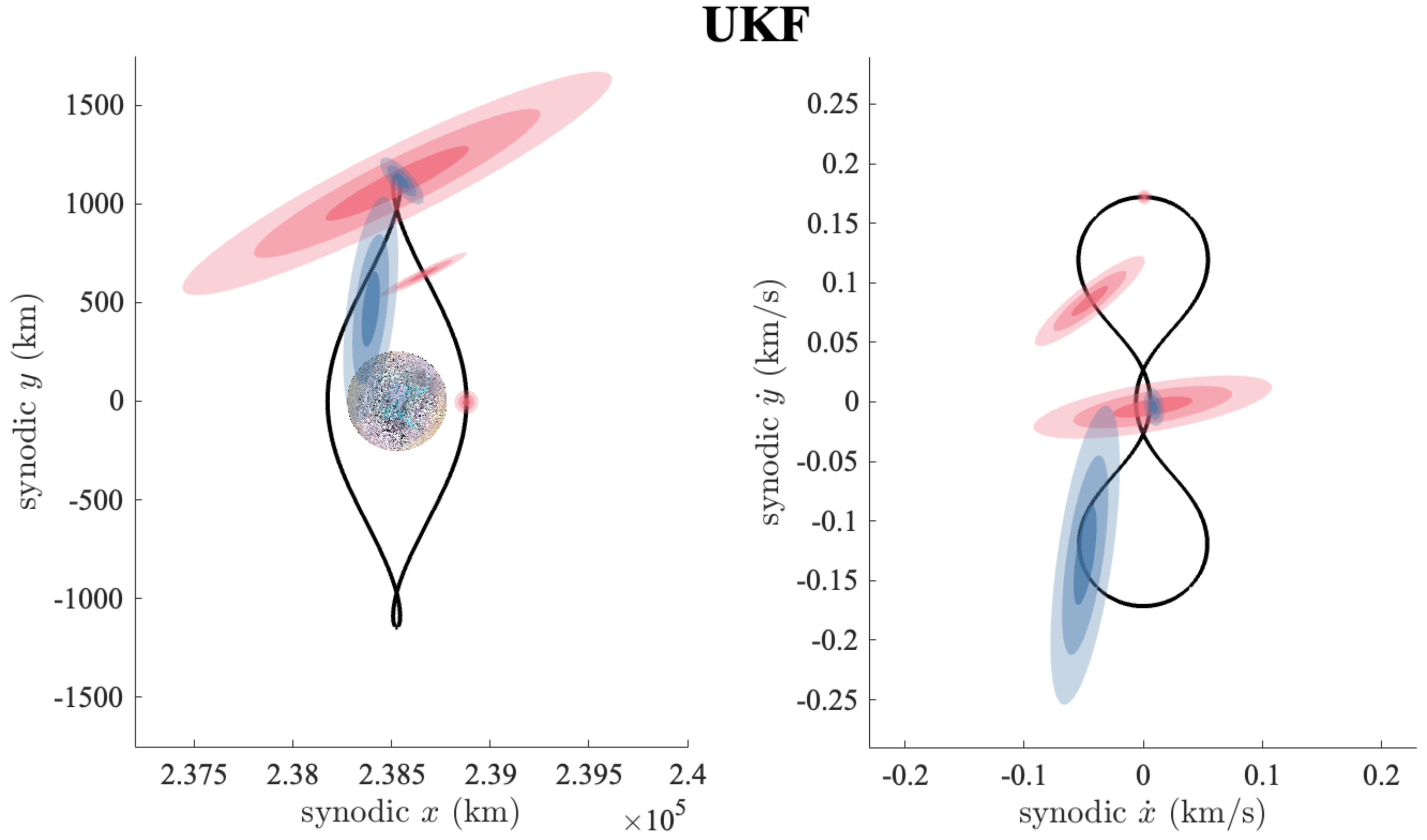


Fig. 4. Family g of periodic orbits. Note that the two parts of the figure have different scales. Orbits with $\Gamma > 4.499986$ are stable, the others are unstable



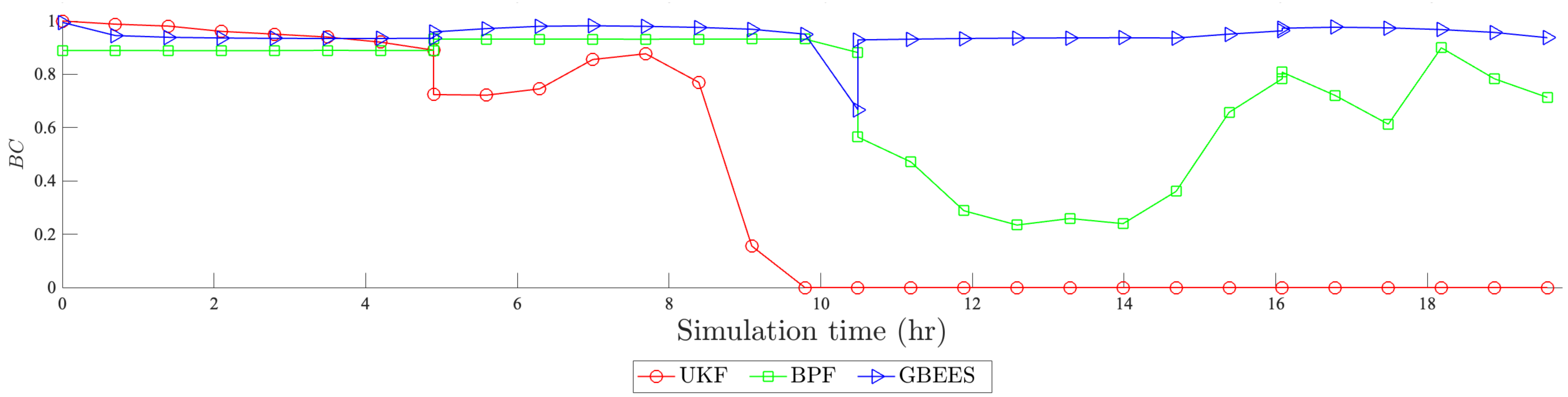
- For these trajectories, with nonlinear measurement updates, Gaussian filters tend to **diverge**



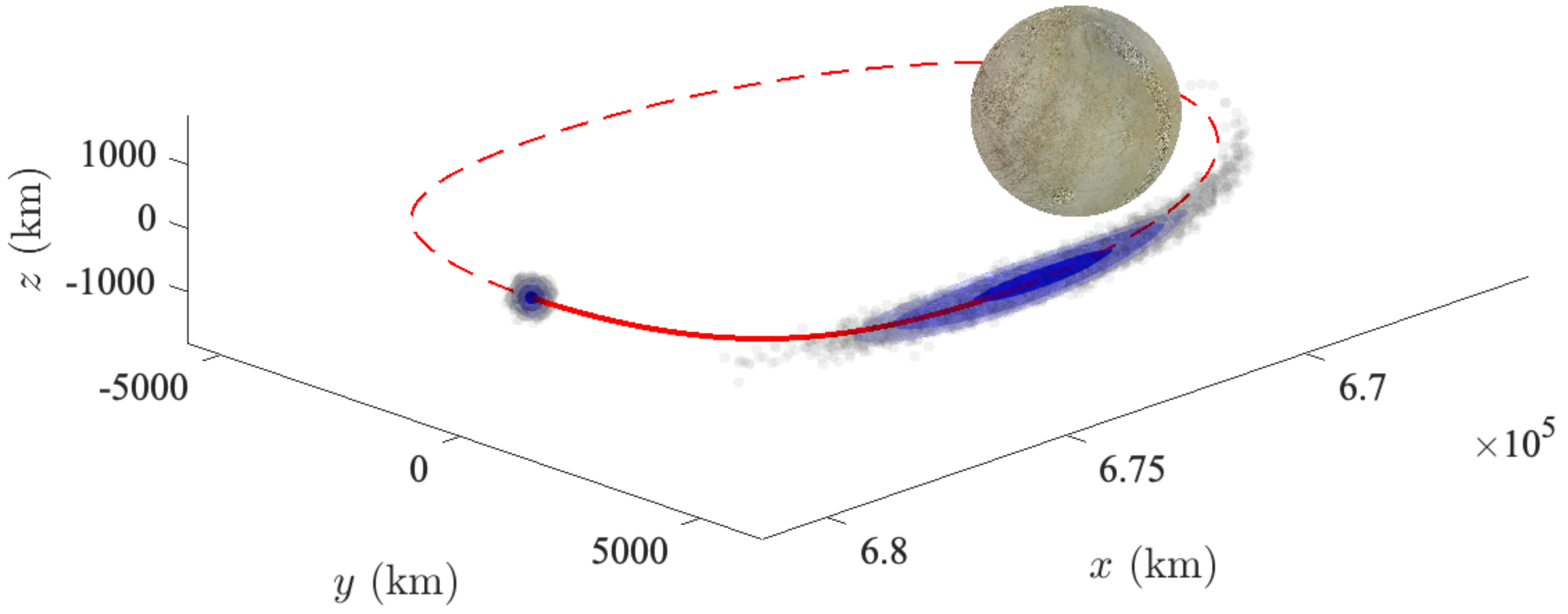


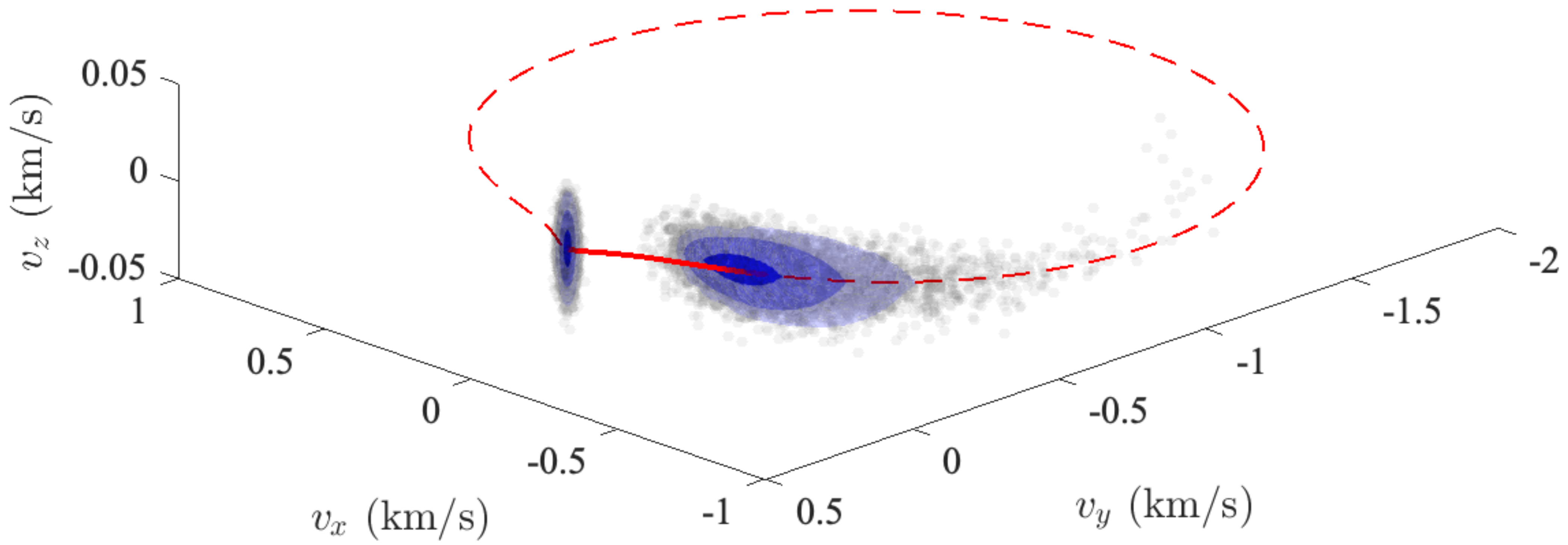
Conclusions

- GBEES proves to be an accurate, robust, and efficient alternative to the landscape of non-Gaussian RBFs



- **Up next:** higher-dimensional systems, parallelization, and ephemeris models (oh my!)





This investigation was supported by the NASA Space Technology Graduate Research Opportunities Fellowship (Grant #80NSSC23K1219)

Thanks to Dr. Ely and Dr. Lo for mentoring and co-mentoring me this summer, as well as for their invaluable insight and contributions.

GBEES can be found at: https://github.com/bhanson10/GBEES

Thank you for your time. Questions?