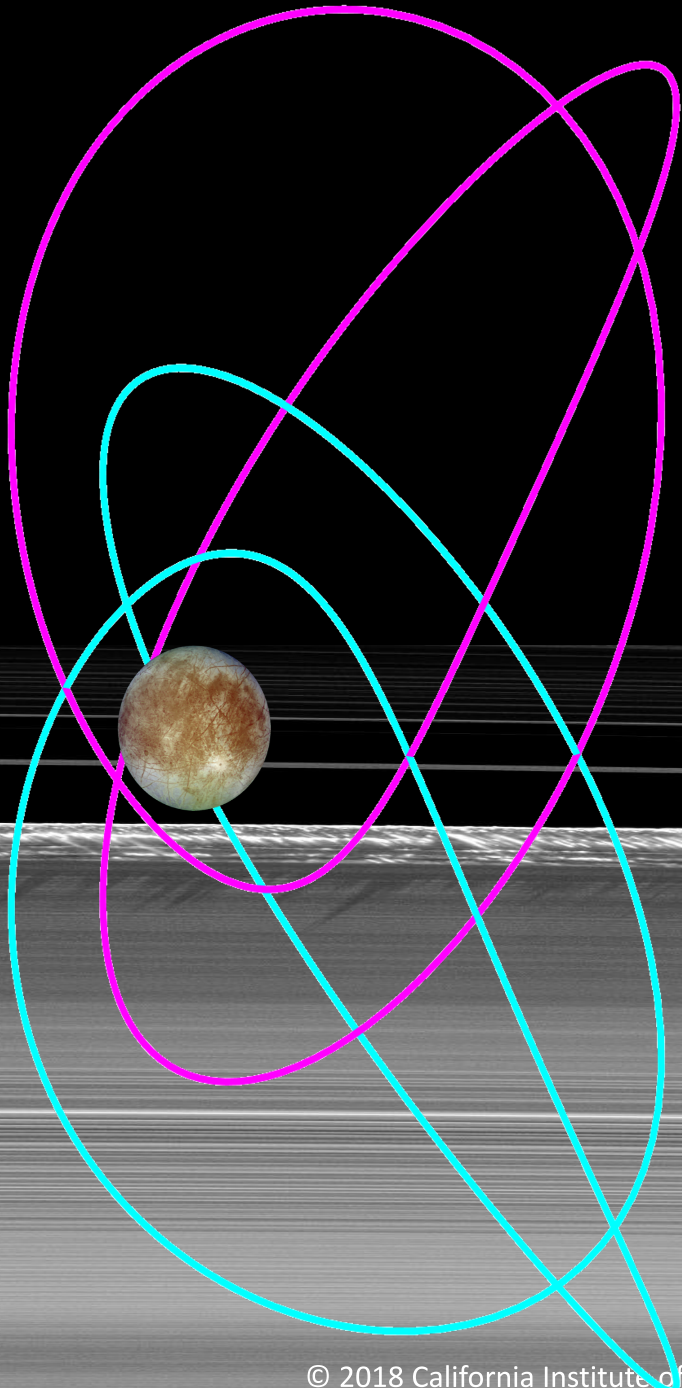


# *Poincare*

## A Multi-Body, Multi-System Trajectory Design Tool in MONTE


Mar Vaquero and Juan Senent

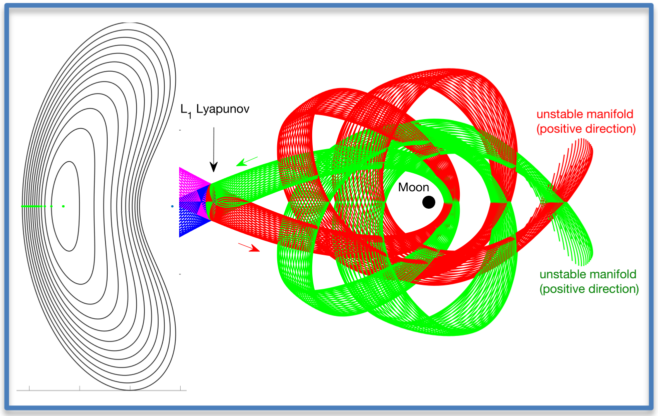
Jet Propulsion Laboratory
California Institute of Technology

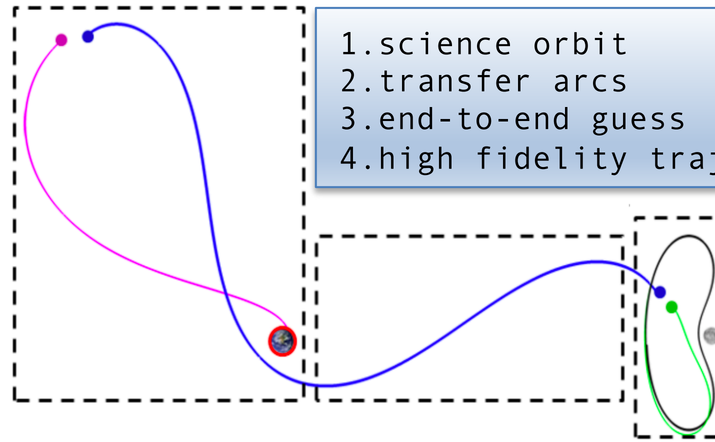
# Poincare

## WHAT Poincare is

- Trajectory design tool based on a catalog of three-body science orbits and a differential corrector to compute connecting transfer arcs between orbits in multi-body systems

mission to an Earth-Moon libration point orbit




## WHY we developed Poincare

- Existing common orbit design approach used in our section is limited and far from intuitive
- Design usually relies on the use of personal code or highly specialized code
- This code generally produces point solutions for a few basic orbital families
- Poincare offers an "all-in-one" integrated search within one interface and setup in MONTE

## HOW we did it

- MONTE Tier2 code with unit tests and formal docs
- Focused on low energy trajectories
- Simple, modular design, easy to maintain, update, and expand

# Orbital Catalog

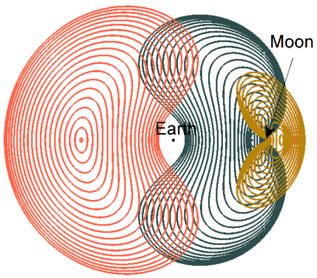
| Libration Point Orbits | P2-Centered Orbits | Resonant Orbits | 3B Systems | Stored Parameters |
|---|---|---|---|---|
| $L_1$, $L_2$, $L_3$ Lyapunov | Distant Retrograde Orbits | 1:1 resonance | Sun-Earth | primary/secondary body |
| $L_1$, $L_2$, $L_3$ Halo | Distant Prograde Orbits | 1:2 resonance | Earth-Moon | gravitational constant mu |
| $L_1$, $L_2$, $L_3$ Vertical | Low Prograde Orbits | 1:3 resonance | Saturn-Titan | Initial state |
| $L_1$, $L_2$, $L_3$ Axial | | 1:4 resonance | Saturn-Enceladus | period |
| $L_4$, $L_5$ Short Period | | 2:1 resonance | Jupiter-Europa | jacobi constant |
| $L_4$, $L_5$ Long Period | | 2:3 resonance | Sun-Mars | eigenvalues |
| $L_4$, $L_5$ Vertical | | 3:1 resonance | Mars-Phobos | stability index $\nu = \frac{1}{2}\left(\lambda_{max} + \frac{1}{\lambda_{max}}\right)$ |
| $L_4$, $L_5$ Axial | | 3:2 resonance | | |
| Butterfly | | 3:4 resonance | | number of patch points |
| Dragonfly | | 4:1 resonance | | diff corrector constraints |
| | | 4:3 resonance | | propagation error and tol |

## How orbital families are calculated

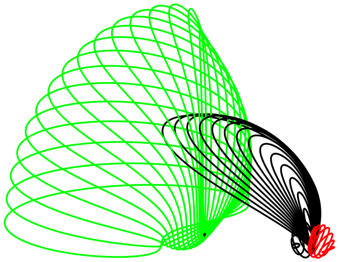
- Multiple shooting differential corrections algorithm based on a generalized constraint and free variable method within pseudo-arclength continuation scheme
- One algorithm for all – easy to implement and maintain
- 36 families per three-body system, over 800,000 orbits currently stored in catalog
- Easy access and search for orbits in standalone SQLite database
- Current plan to make the database publicly available in JPL's Horizons website

# Orbital Catalog

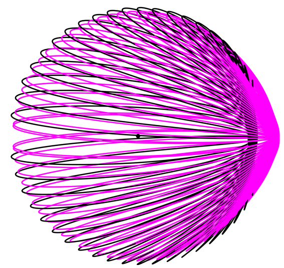
## Lyapunov



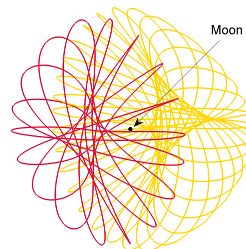

## Halo



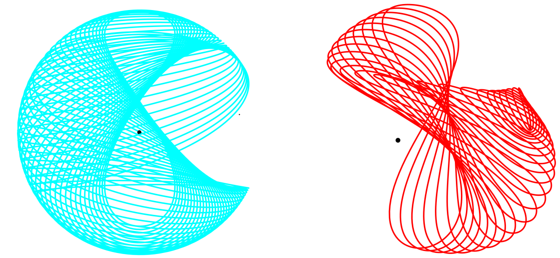
## Vertical



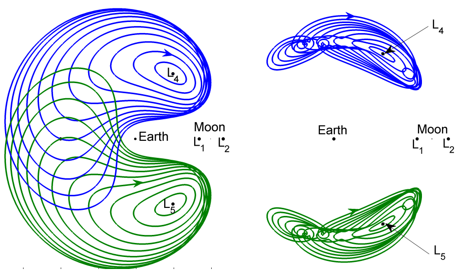
## Axial



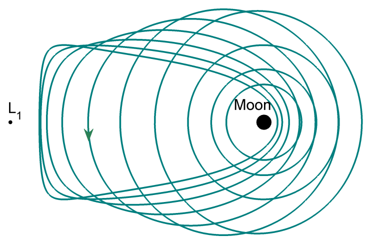

## Triangular Vertical/Axial



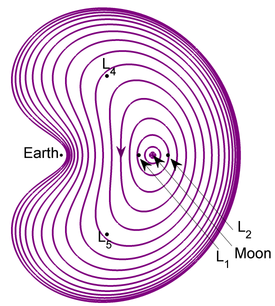
## Short/Long Period



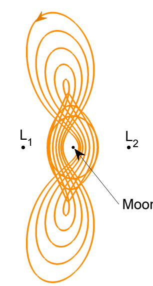

## LPO



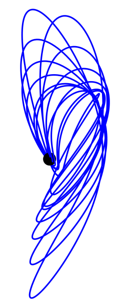

## DRO




## DPO

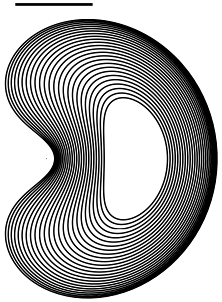



## Dragonfly

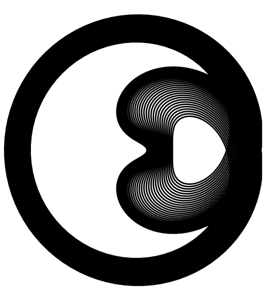


## Butterfly

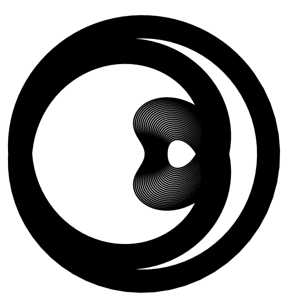
## 1:1



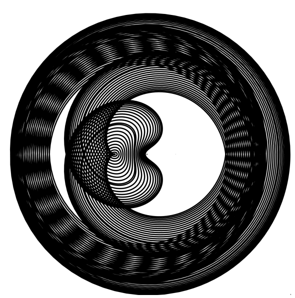
## 1:2



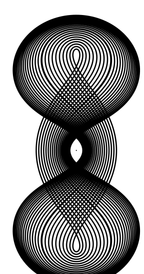
## 1:3



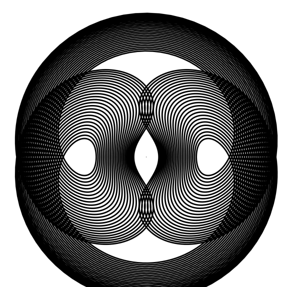
## 1:4



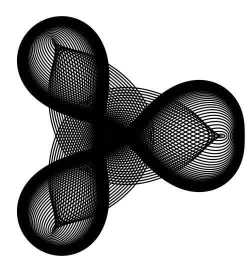
## 2:1



## 2:3



## 3:1



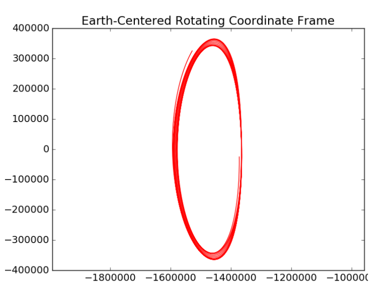
# Dynamical Model

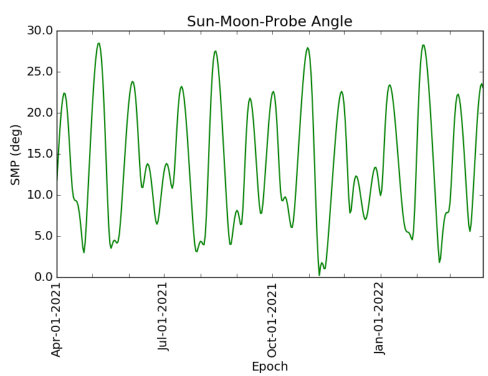
## Circular Restricted Three-Body Model BOA universe

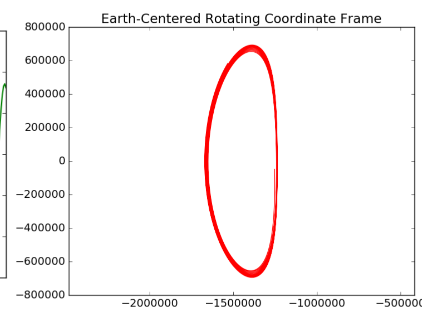
- Provides independent MONTE python packages:
    - set up a three-body universe → `poincare.makeCR3BP()`
    - access and search for orbits in catalog → `poincare.queryDB()`
    - parameterize units if needed → `poincare.makeDimensional()`
    - easy state propagator interface → `poincare.propagate()`
- Visualize and save trajectory in a boa, then utilize full MONTE capabilities

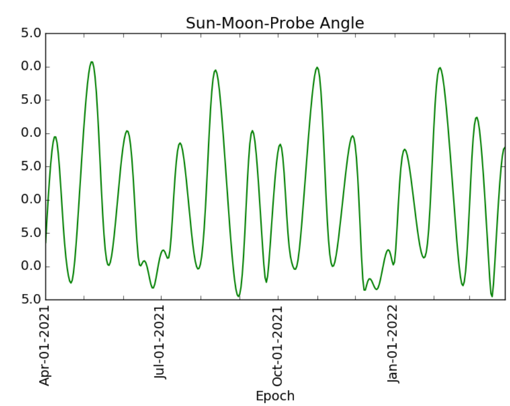
## Common Applications

- Almost one-million multi-body periodic orbits readily available
- Fast search and selection based on parameters like orbital energy or stability
- Ideal for quick science orbit trade-off studies



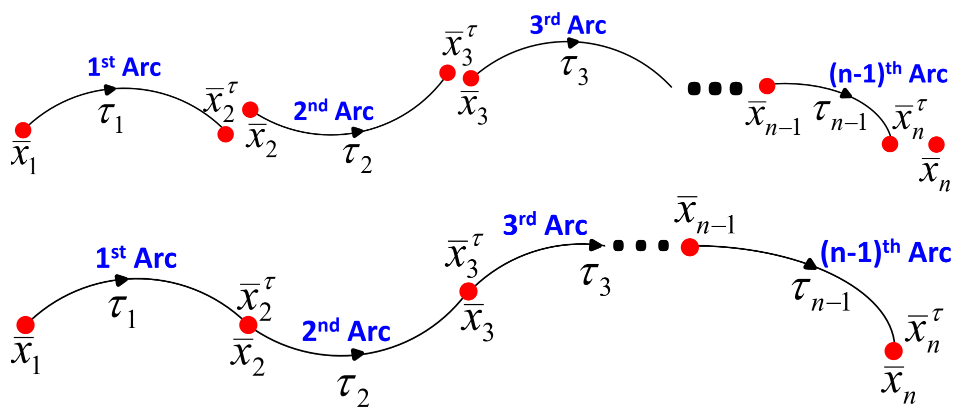



# Differential Corrector

- Expands the use of the periodic orbit catalog
- Straightforward free variables and constraints implementation in MONTE Tier-2




```
poincare.corrector.makePatchPoints()
poincare.Corrector()
poincare.corrector.solve()
poincare.addClosedOrbitConstraints()
poincare.corrector.addConstraint[Cartesian.y()]
```

```
7  Cosmic/CORRECTOR/P1/X -1.151972689405265e+04 *km -1.151972689405265e+04 *km -1.519726894052646e+03 *km
15 Cosmic/CORRECTOR/P2/Y  9.604395582072197e+03 *km   1.955322770694432e+04 *km  1.960439558207220e+04 *km
------------------------------------------------------------------------------------------------------
9  P0->P1 State: Y  -1.000000000000000e+00 *km      1.026111093939107e+00 *km      1.000000000000000e+00 *km
11 P0->P1 State: DX -1.000000000000000e-05 *km/sec -1.111310906364300e-05 *km/sec  1.000000000000000e-05 *km/sec
15 P1->P2 State: Y  -1.000000000000000e+00 *km      1.063062499204534e+00 *km      1.000000000000000e+00 *km
20 P2->P3 State: X  -1.000000000000000e+00 *km     -1.005207620703004e+00 *km      1.000000000000000e+00 *km
23 P2->P3 State: DX -1.000000000000000e-05 *km/sec -1.104945924335077e-05 *km/sec  1.000000000000000e-05 *km/sec
24 P2->P3 State: DY -1.000000000000000e-05 *km/sec -1.009549871732254e-05 *km/sec  1.000000000000000e-05 *km/sec
------------------------------------------------------------------------------------------------------
Problem:   CORRECTOR
Optimizer:   DBLSE
Status:  NOT_SET
Iterations:   5
======================================================================================================

TOTAL( SUM )  0.000000000000000e+00

7  Cosmic/CORRECTOR/P1/X -1.151972689405265e+04 *km -1.151972689405265e+04 *km -1.519726894052646e+03 *km
15 Cosmic/CORRECTOR/P2/Y  9.604395582072197e+03 *km   1.955334137018752e+04 *km  1.960439558207220e+04 *km
------------------------------------------------------------------------------------------------------
17 P1->P2 State: DX -1.000000000000000e-05 *km/sec -1.000001326710749e-05 *km/sec  1.000000000000000e-05 *km/sec
23 P2->P3 State: DX -1.000000000000000e-05 *km/sec -1.000002003226363e-05 *km/sec  1.000000000000000e-05 *km/sec
------------------------------------------------------------------------------------------------------
Problem:   CORRECTOR
Optimizer:   DBLSE
Status:  CONVERGED
Iterations:   6
======================================================================================================

TOTAL( SUM )  0.000000000000000e+00

7  Cosmic/CORRECTOR/P1/X -1.151972689405265e+04 *km -1.151972689405265e+04 *km -1.519726894052646e+03 *km
15 Cosmic/CORRECTOR/P2/Y  9.604395582072197e+03 *km   1.955334136972732e+04 *km  1.960439558207220e+04 *km
```

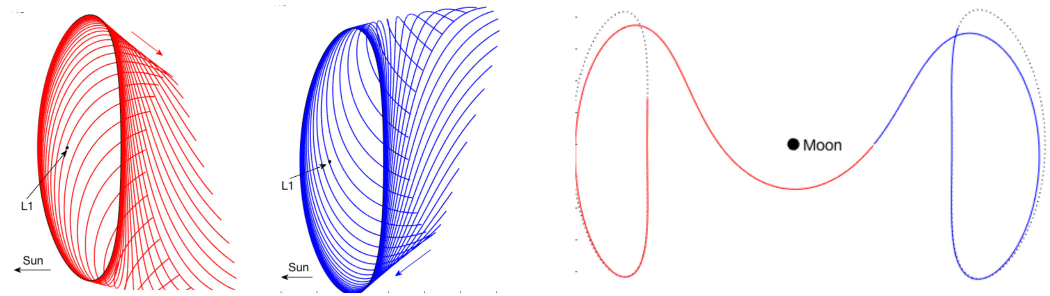
## Common Uses

- Compute a specific periodic orbit by specific Jacobi constant, period, or amplitude
- Compute quasi-periodic orbits starting from a periodic orbit by selecting the number of revolutions or amplitude (e.g. Lissajous orbits)
- Compute quasi-periodic orbits in a full ephemeris system from a periodic orbit in the CR3BP

# Manifold Generator

- Manifold trajectories are efficient mechanisms for transport between orbits

  unstable → depart the orbit
  stable → approach the orbit



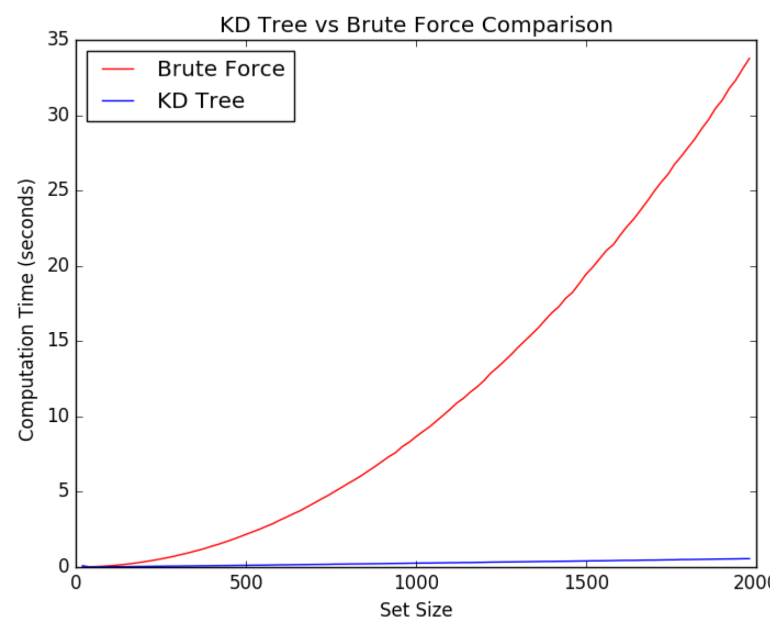

- Defined parameters include:
  - number of stable and unstable manifold trajectories to be integrated
  - propagation time for each manifold trajectory
  - perturbation parameter
- Optional algorithm to systematically determine the step size for each periodic orbit
- Output includes:
  - dynamical system set-up
  - initial conditions associated with each propagated manifold trajectory
- Fast and efficient on-the-fly computation
- Output BOA file is small-sized and contains all information needed for re-integration or post-processing of manifold trajectories

# Map Generator

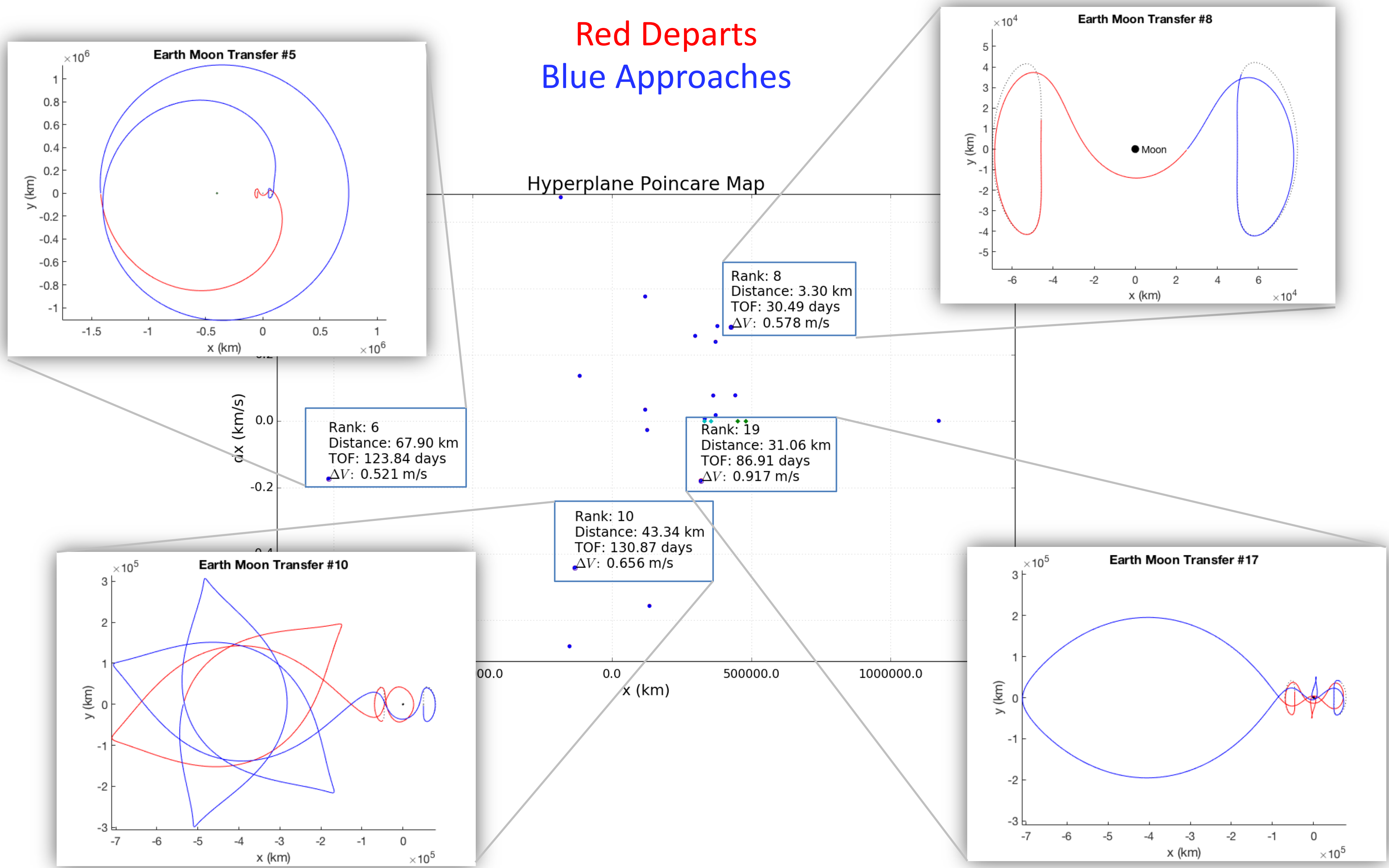
- Current prototype available, final delivery expected in 2019
- Real-time computation of orbit connections via stable and unstable manifolds
- Allows user to generate and interact with Poincaré Maps
- Available types of maps include any user-specified surfaces of section, periapsis, apoapsis, and close-pass (or flybys) maps.
- Hooks exist to easily define any other type of map
- Efficient nearest neighbor search available for systematic detection of connections:
  - Similar to a Binary Search
  - Implemented in SciPy
  - $O(\log n)$

**Preliminary results based on current prototype**

| Step | Time Taken |
|---|---|
| Manifold trajectory propagation (10k) | 10 Minutes |
| Map generation (event search) | 2 Minutes |
| KD tree search | 3 Minutes |



# Optimal Transfer Finder Prototype



# Poincare in a Nutshell

Theory regarding computation and connection of multi-body orbits is pre-existing and well-understood; however, it was not available in a user-friendly way and within an all-in-one environment in MONTE

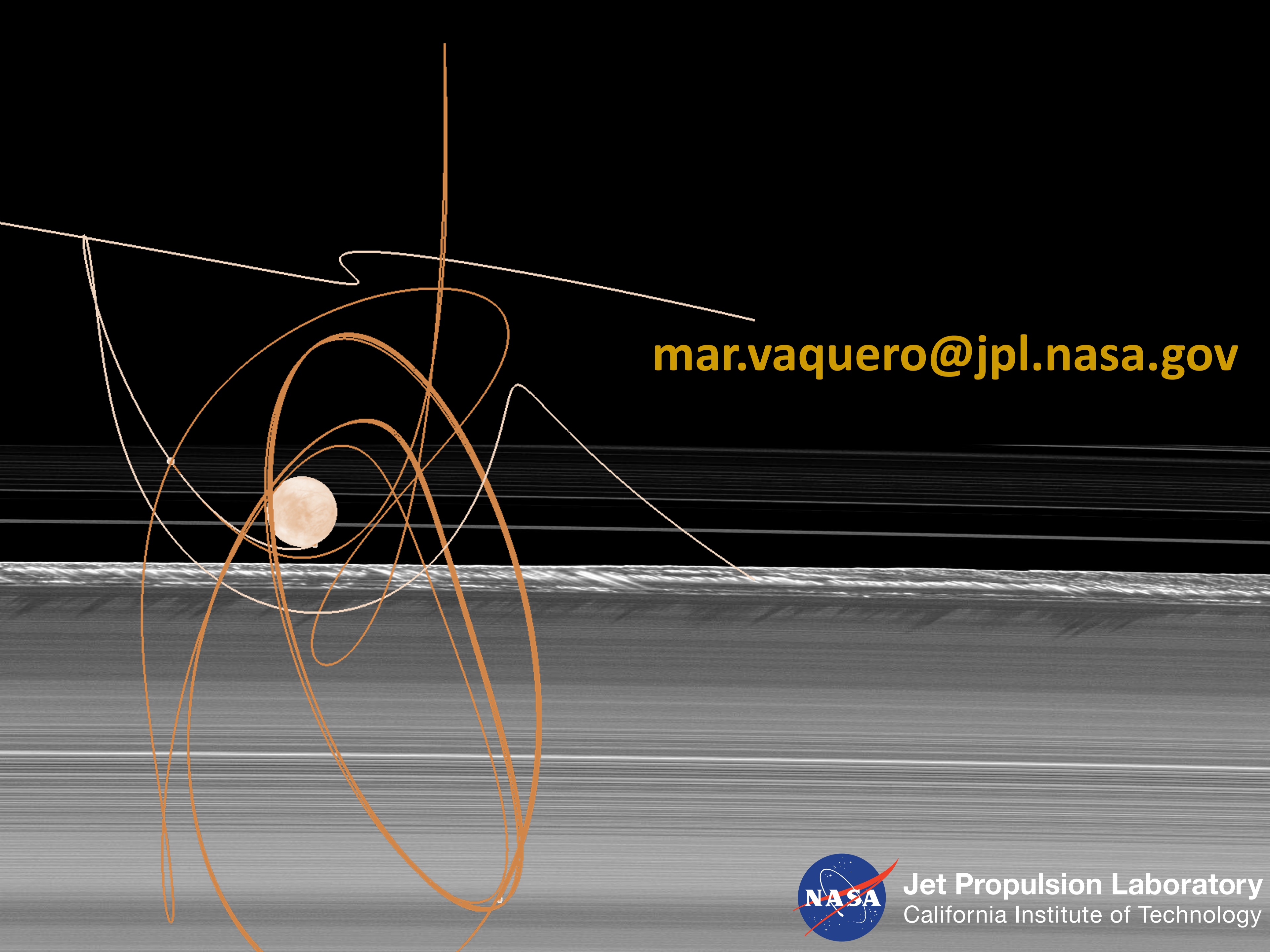
Allows mission designers to explore destinations, to or from a science orbit, that fit within time of flight and propellant constraints by leveraging low energy multi-body dynamical techniques to produce low-cost, or even $\Delta V$-free transfers

- User-friendly, modular design
- Easy maintenance and highly adaptable to changes in core scripts

extensive multi-body orbital catalog

robust differential corrector

manifold and map generators


mar.vaquero@jpl.nasa.gov
Jet Propulsion Laboratory
California Institute of Technology