THE UNIVERSITY OF TEXAS AT AUSTIN GTOC11 SOLUTION

Using Lookup Tables, Lambert Calls, Indirect Optimization, and a Genetic Algorithm Scheduler

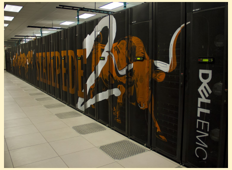
Space Trajectory Computation Lab
Presenter: Sean McArdle
December 18, 2021

Solution Method Outline

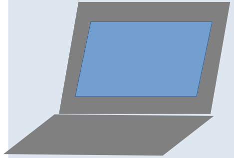
- Preliminary Analyses (backup slides)
- Mothership Trajectory Search
  - Lookup Tables
  - Lambert Calls
- Asteroid-to-Ring Trajectory Solver
  - Indirect Optimization
- Build Station Arrival Scheduler
  - Genetic Algorithm

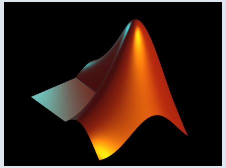
Mothership Trajectory Search:
Desktop Workstations, Fortran

Asteroid-to-Ring Trajectory Search:
Stampede2, Fortran / MPI



Build Station Arrival Scheduler:
Laptop, MATLAB





Mothership Trajectory File

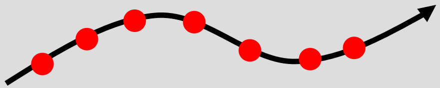
Asteroid-to-Ring Trajectory File

Solution File

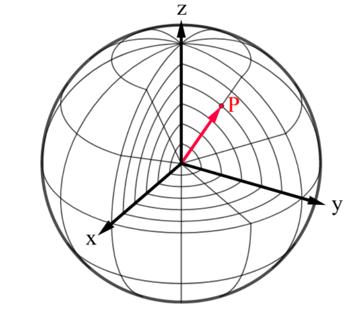
Mothership Trajectory Search

Mothership Trajectory Search

- Precompute asteroid/Earth position lookup table
  - Temporal grid (3 day time steps over 20 years)
  - Spatial grid (latitude/longitude/radius)
- Leg 1 (Earth-to-Asteroid)
  - Minimum dv solutions to every asteroid
- Legs 2 through M (Asteroid-to-Asteroid)
  - Identify asteroids in current/neighboring bins over selected leg TOF
    - Compute flyby maneuvers to considered asteroids
    - Choose asteroid encounter that gets best ‘extrapolated cost’
  - If no asteroids encountered, try again with longer leg TOF
    - Stop after N tries or if 20 year end time is met



Temporal Grid



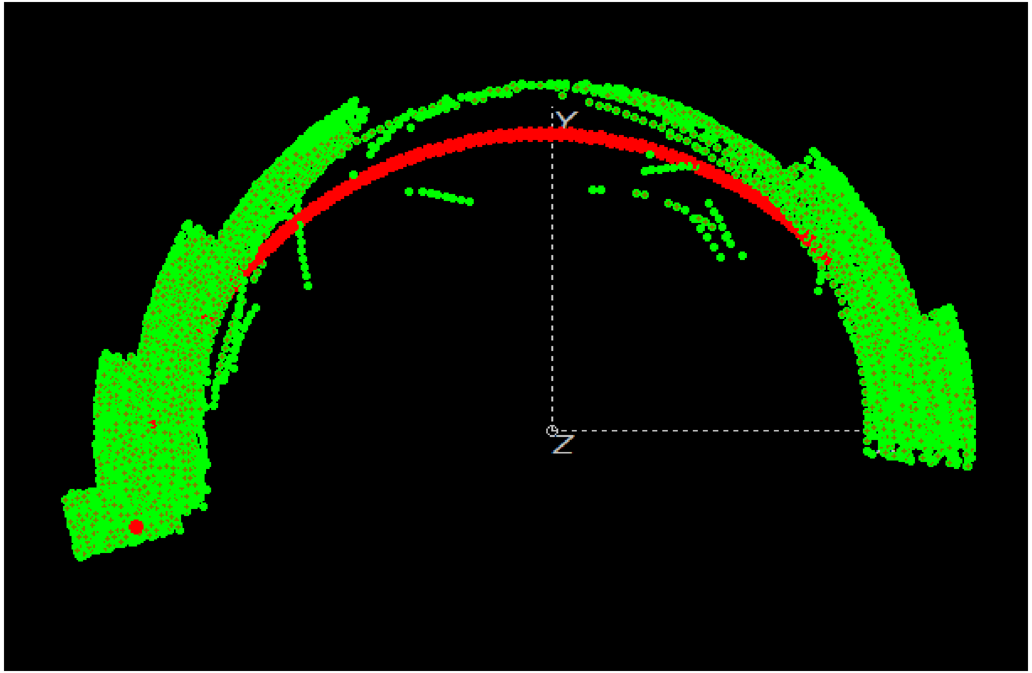
Spatial grid

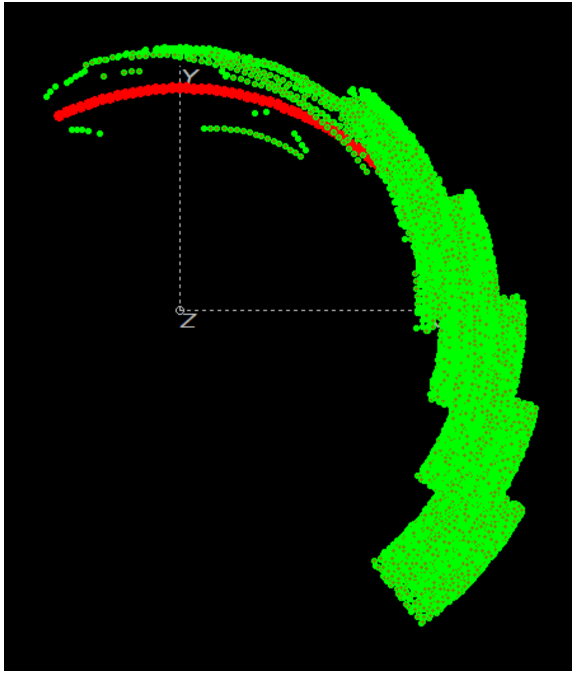
New Lambert Solver

- Russell, Ryan P., “Complete Lambert Solver Including Second-Order Sensitivities”, *Journal of Guidance, Control, and Dynamics*, Online Nov. 2021, https://doi.org/10.2514/1.G006089. (open access)
  - Fortran with MATLAB drivers online: link in paper
  - Coversine formulation
  - Robust/Fast. Converges in 1-2 iterations. (~0.2×10^{-6} s per call)
  - Interpolates whole domain for initial guess. 1 MB coefficient file, works for all N

- GTOC was great testbed for new solver
  - trillions of calls! no failures
  - mainly used 0-rev case

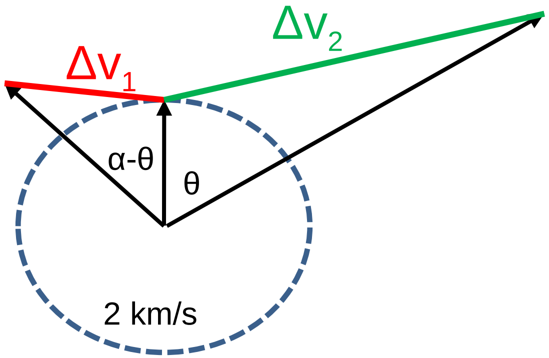
Identifying Nearby Asteroids Over Leg TOF



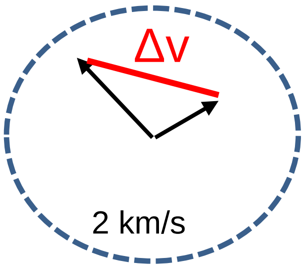


Red: Mothership (covered up by green usually)
Green: Asteroids in mothership bin or neighboring bins
Dark yellow: asteroid that passes {dv,**e**,**h**} filter

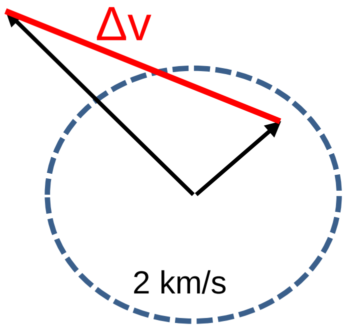
Computing Flyby Maneuver(s)



Case 1



Case 2



Case 3

Minimize J= $\Delta v_1 + \Delta v_2$ over θ

$$J := \sqrt{vA^2 + 4 - 4\, vA \cos(\theta)} + \sqrt{vB^2 + 4 - 4\, vB \cos(\alpha - \theta)}$$

one simple Δv

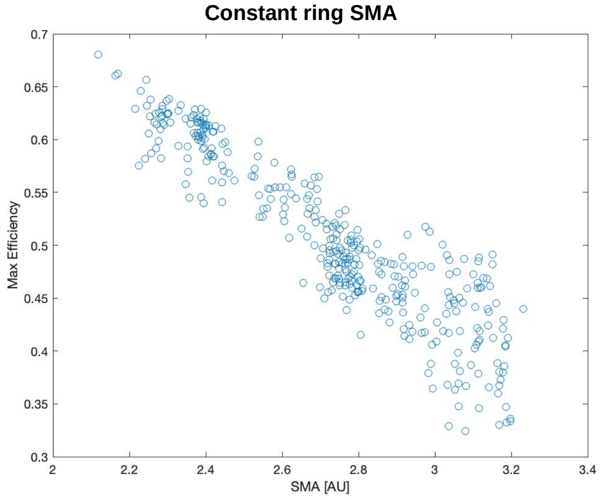
Solve with a 1D grid search, multiple solutions

Mothership Trajectory Selection

- Tens of thousands of available mothership trajectories
- Expensive to compute asteroid-to-ring trajectories
- Quick estimator for delivered mass:
  - Trained with thousands of asteroid-to-ring solutions
  - Linear function of asteroid and ring SMA
- How to pick?
  - Select highest scoring mothership trajectory
  - Remove the asteroids encountered by the selected trajectory from all remaining trajectories and repeat

Efficiency = delivered mass/initial mass

Maximum for the same asteroid over different build station time/phase configurations.



Summary Mothership Comments

- Method was very effective at quickly finding hundreds of ‘high-performing’ itineraries
- All mothership trajectories found using two multicore desktop servers
- Efficient because
  - Lookup bins reduce which asteroids to consider
  - Precomputed asteroid locations
  - 1D search on TOF with Lambert solver leads to low DV for each asteroid option
  - ‘Extrapolated cost’ was a good way to choose ‘best’ next leg of many options
- What we would have done with more competition time….
  - Use TACC or supercomputing to perform mothership searches
  - Include global optimization schemes for decision making instead of ‘best’ or ‘second-best’ etc. option for each next leg choice
  - Couple each mothership search directly with the sequence selection of 10 motherships
  - Locally optimize each mothership itinerary by adjusting flyby times using an end to end optimizer
  - Substitute rather than remove asteroids encountered by multiple selected motherships

Adding DVs between Asteroid Flybys

1 Take all asteroid-to-asteroid legs

- Fixed-state to fixed-state with fixed flight time

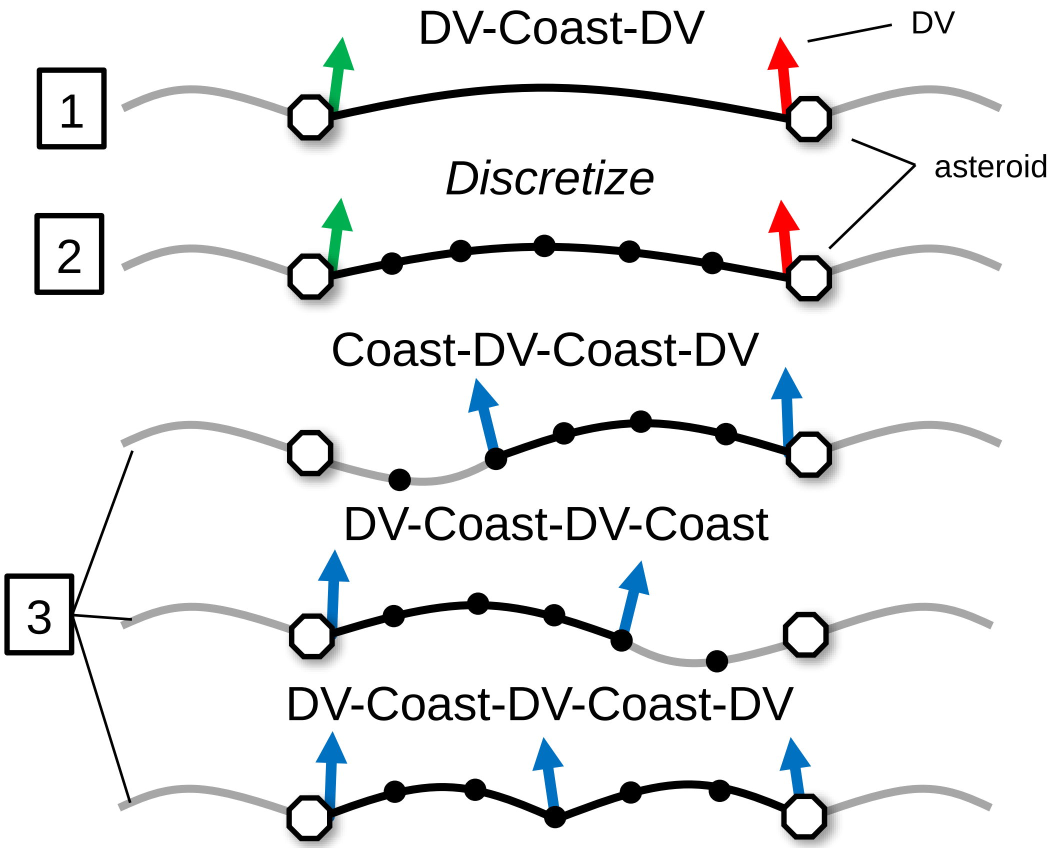
2 Convert one leg from 1 to 20 segments

- Two positions and one flight time per seg.
  - Many solutions to Lambert's problem
- Only positions vary
- Enables *potentially* more DVs and coasts

3 Final structure

- Each mothership trajectory had 0-3 improved legs
- 14 legs in total were improved

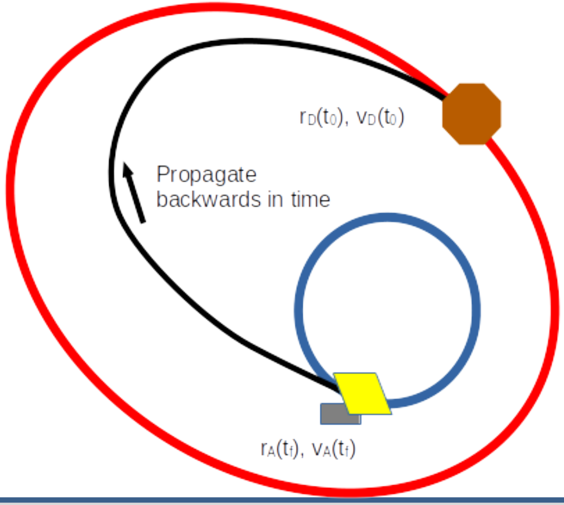
Summary of DV Totals
OLD 146.1760 km/s
NEW 145.9726 km/s
SAVE 0.20340 km/s **(-0.1391%)**



Asteroid-to-Ring Trajectory Solver

Asteroid-to-Ring Trajectory Solver

Constant Acceleration, Fixed Final Arrival Time
Initial Position/Velocity on Orbit D (Departure)
Final Position/Velocity on Orbit A (Arrival)



Dynamics

Keplerian Departure/Arrival Orbits

$\mathbf{r}_D(t_0)$, $\mathbf{v}_D(t_0)$ $\mathbf{r}_\mathbf{A}(t_f)$, $\mathbf{v}_\mathbf{A}(t_f)$

Constant Acceleration Transfer (12 States)

$H = \boldsymbol{\lambda}_r^T\mathbf{v} - \boldsymbol{\lambda}_v^T(\mu\mathbf{r}/||\mathbf{r}||^3 + \Gamma_{ATD}\boldsymbol{\lambda}_v/||\boldsymbol{\lambda}_v||)$

$d\mathbf{r}/dt = \mathbf{v}$

$d\mathbf{v}/dt = -\mu\mathbf{r}/||\mathbf{r}||^3 - \Gamma_{ATD}\boldsymbol{\lambda}_v/||\boldsymbol{\lambda}_v||$

$d\boldsymbol{\lambda}_r/dt = -\partial H/\partial\mathbf{r}$

$d\boldsymbol{\lambda}_v/dt = -\partial H/\partial\mathbf{v}$

Two Point Boundary Value Problem:

Unknowns (7)

$\boldsymbol{\lambda}_{rf}$ (3)
$\boldsymbol{\lambda}_{vf}$ (3)
t_0 (1)

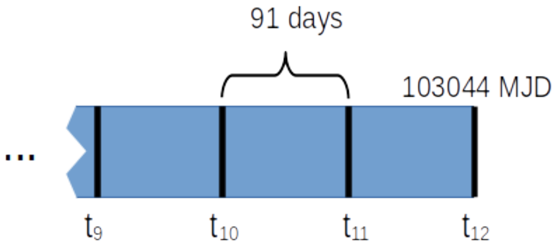
Constraints (6)

$\mathbf{r}_0 = \mathbf{r}_D(t_0)$ (3)
$\mathbf{v}_0 = \mathbf{v}_D(t_0)$ (3)

Not strictly minimum time!

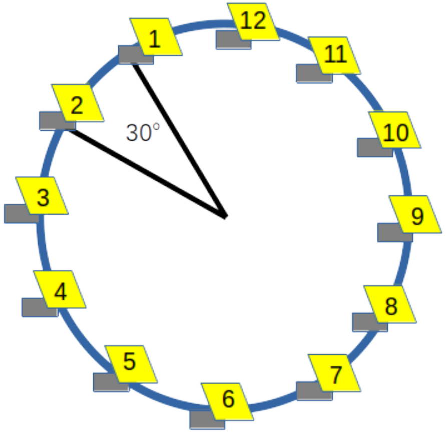
Asteroid-to-Ring Trajectory Solver

- Try every **Time Slot** / **Build Station** combination
  - 144M (~43,000) total trajectories per run
  - M is number of asteroids encountered by motherships



Time Slots

- Levenberg-Marquardt trust region solver
  - 1st order derivatives – complex step
  - 2nd order derivatives – finite difference

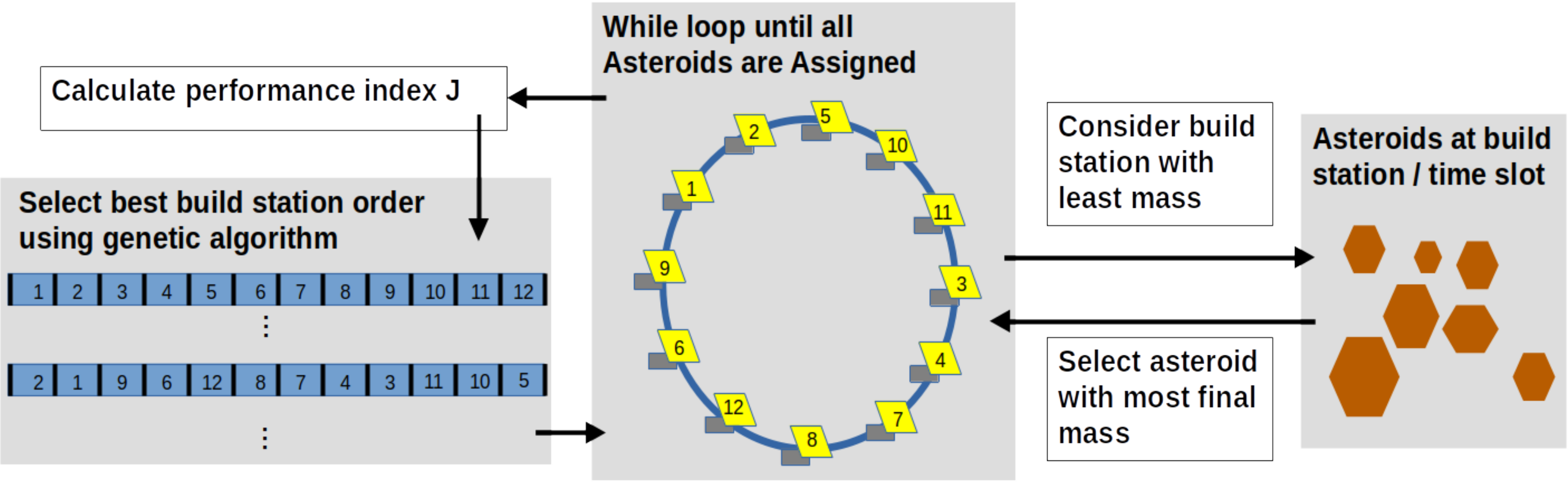


Build Stations

- Message Passing Interface (MPI) parallelization
  - Fortran implementation on Stampede2
  - One MPI task per asteroid / build station combo.
  - 3000-4000 tasks, ~50 nodes, ~2 hour runtime

Build Station Arrival Scheduler

Build Station Arrival Scheduler

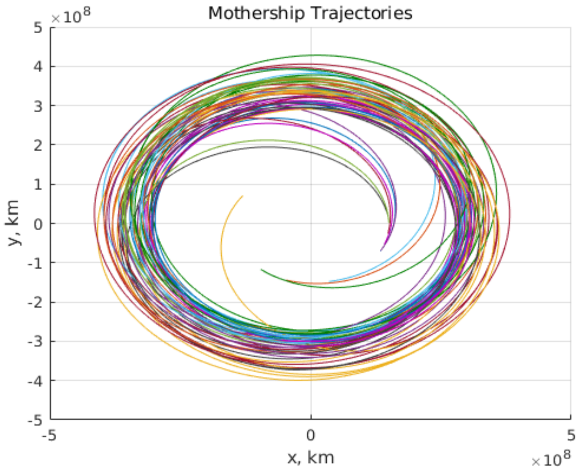


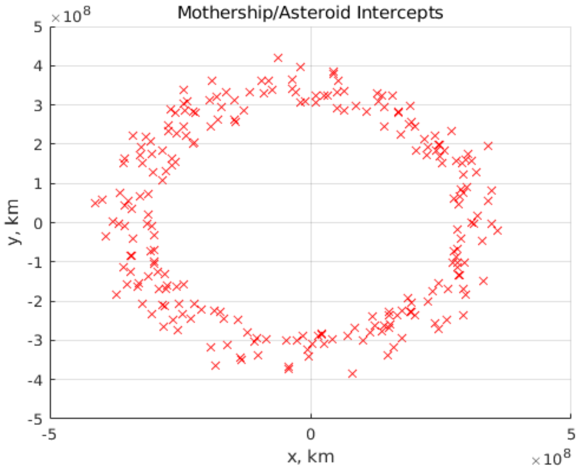
Best Solution Details

J	5885.469300
Mmin	1.13283e+15 kg
N	235 asteroids
Ring Geometry	
Semimajor Axis	1.1 AU
Inclination	0.0 deg
RAAN	0.0 deg
Phase	0.0 deg

Ring SMA grid search

Tried matching average asteroid orbital plane geometry, did not perform better





UT Austin GTOC11 Team

- Billy Brandenberg
- David Cunningham
- Courtney Hollenberg
- Sean McArdle
- David Ottesen
- Ryan Russell
- Chun-Yi Wu
- Burton Yale
- Enrico Zucchelli

Image Credit: https://cockrell.utexas.edu/images/campus/ase-building.jpg

Questions?

Backup Slides

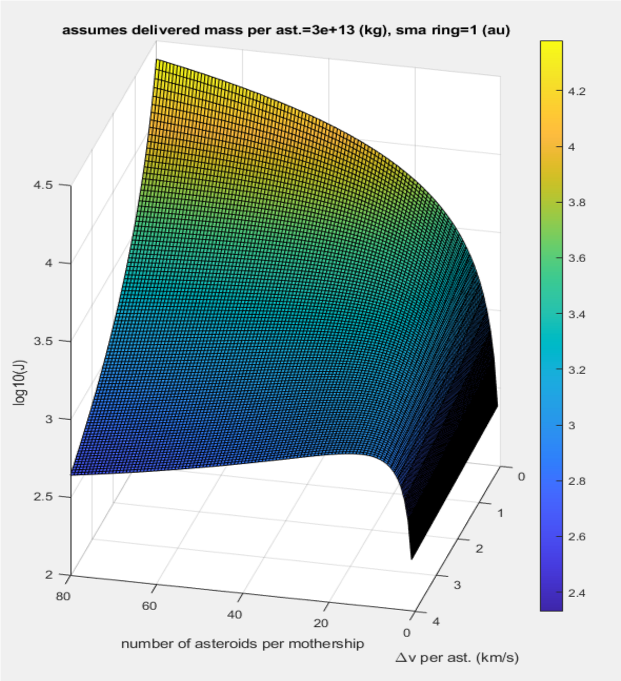
Preliminary Analyses

- Sensitivities of the performance index
- Orbital element distributions of the asteroids
- Proxy filters for asteroid to asteroid transfers…
  - Not including details but…
  - Performed many min dv transfers between asteroid pairs
  - Characterized times of flight and bounds for deltavs
  - Developed proxy filters using distance between ecc. vector and ang. momentum vector as to limit which asteroids are reachable in reasonable times/costs

More asteroids or less Δv ?

Possible J (1 AU ring)

Number of Asteroids vs. Δv/ast.



$$J := \frac{1.000000000\,10^{-11}\, mPer\, nA}{a^2 \left(1 + \frac{dvA\, nA}{50}\right)^2}$$

Analytical max occurs for fixed Δv/asteroid (call it Δv_a) occurs at n/asteroid (call it N_a) of

$N_a = 50/(\Delta v_a)$

So… if you use 1 km/s per asteroid, you can get up to 50 before starting to reduce J

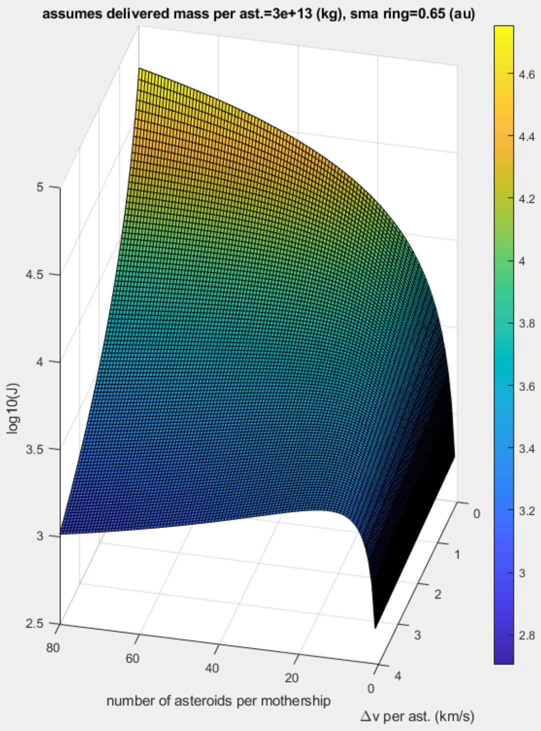
if you use 5 km/s per asteroid, you can get up to only 10 before starting to reduce J

Relationship holds regardless of sma ring (a) or delivered mass per asteroid (mPer)

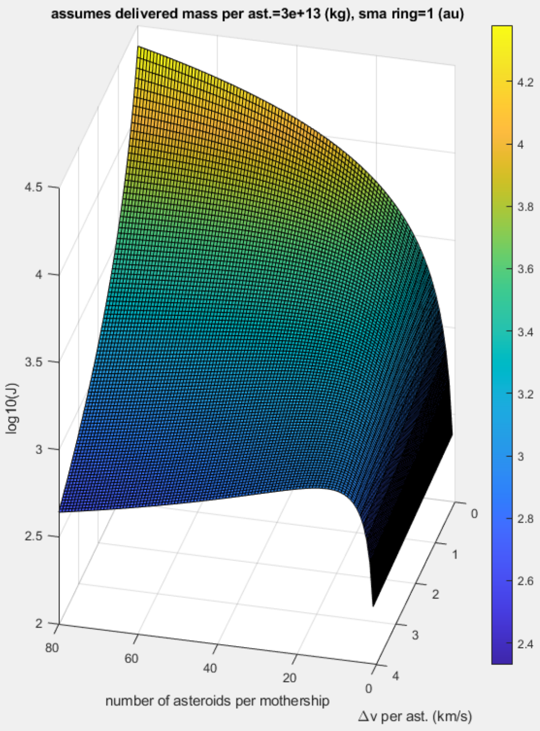
Bottom line: better plan to reduce Δv per asteroid than get more of them. (makes sense because squared vs linear)

Possible J (consider different ring SMA)

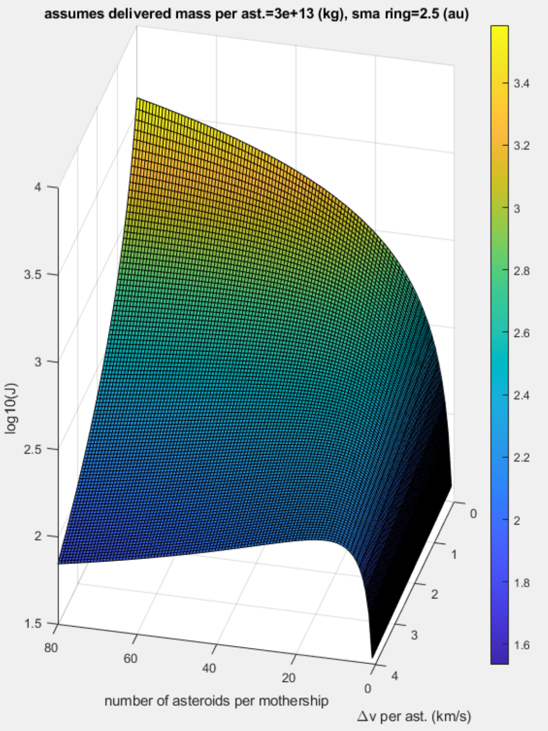
0.65 AU



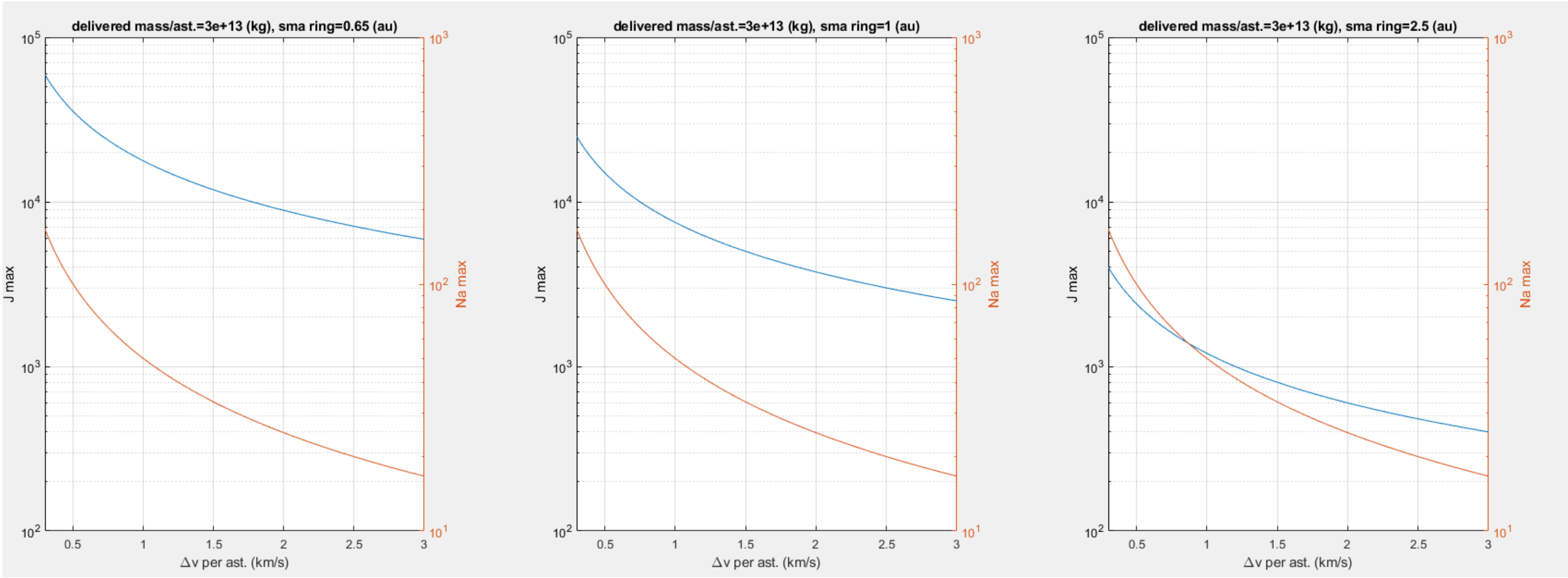
1 AU

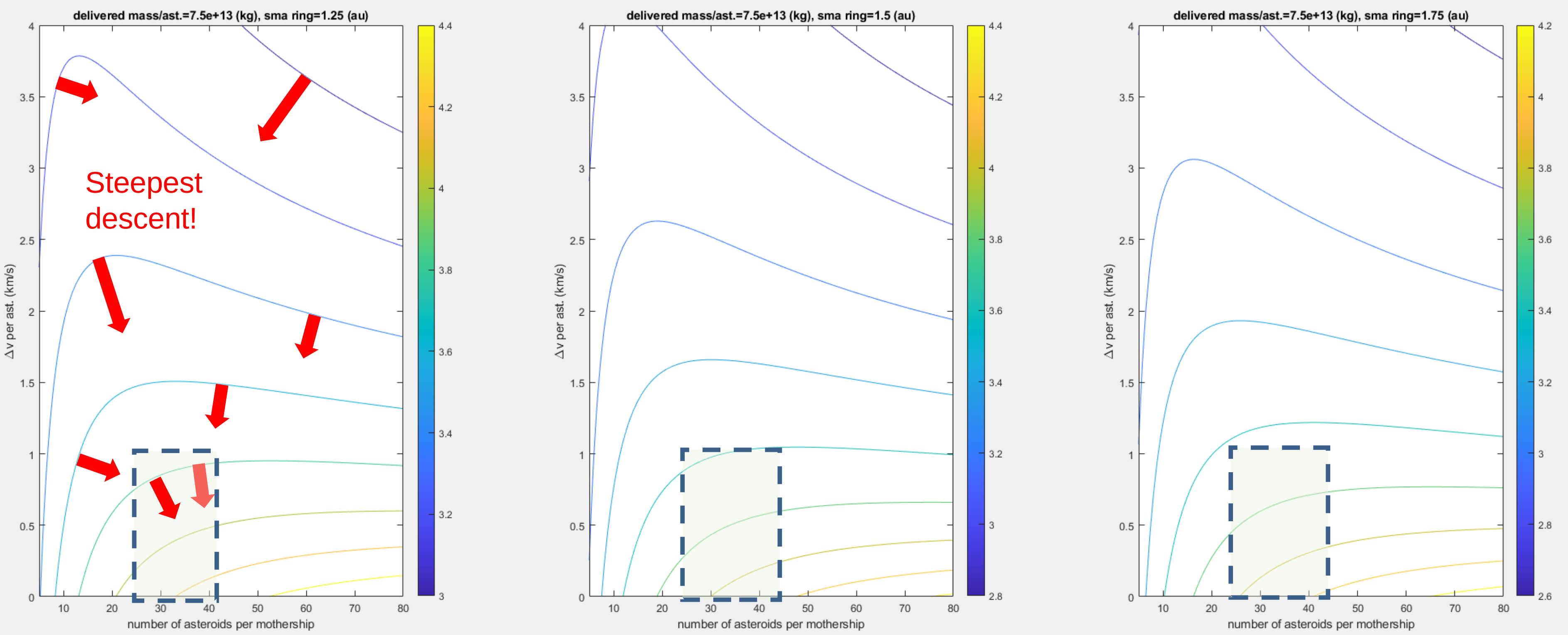


2.5 AU



Max possible J as function of dv/asteroid



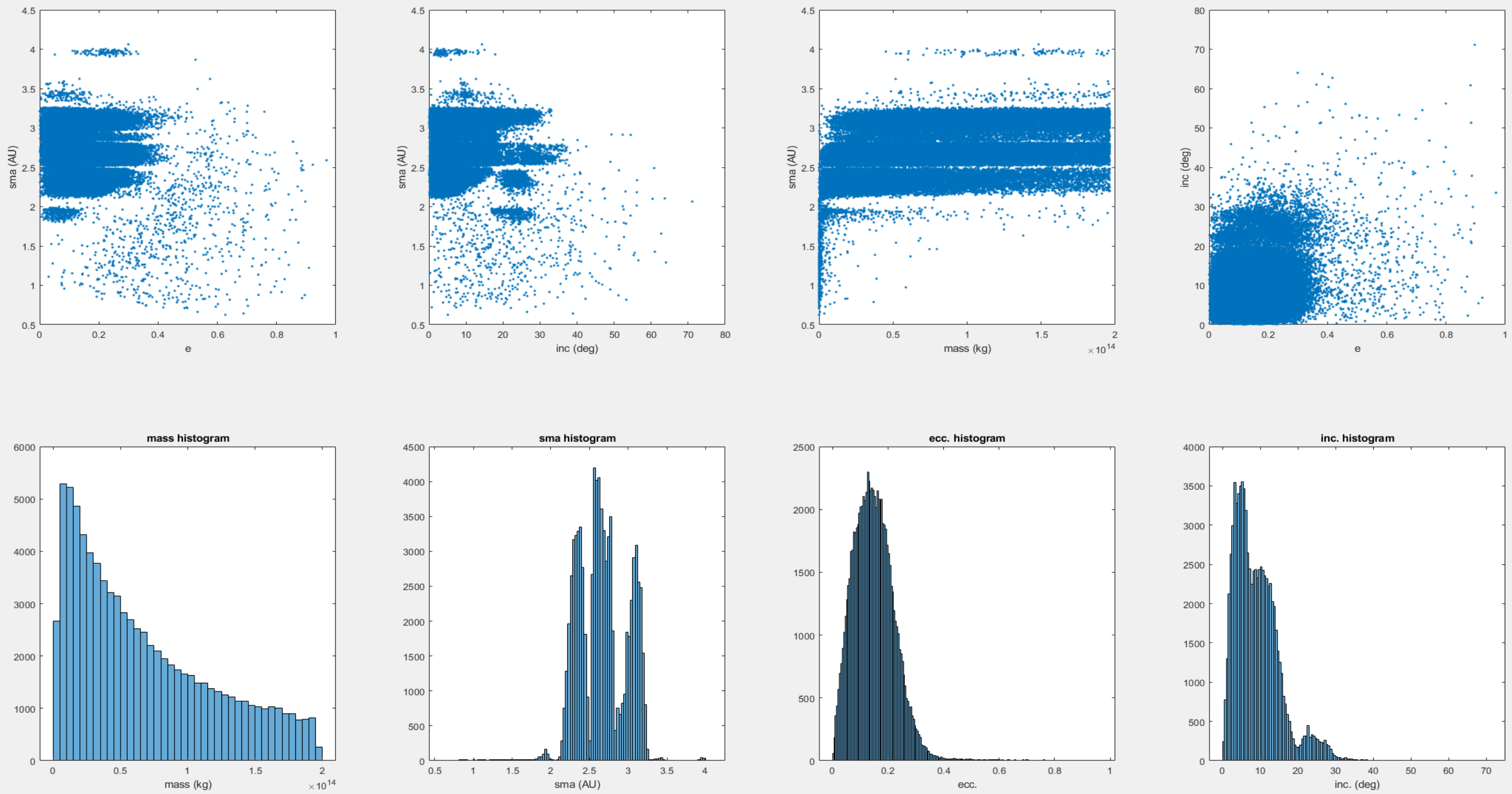


Wound up operating in box where its good to reduce dv/ast and increase #asteroids, but more efficient to reduce dv/asteroid than increase #asteroids

TEXAS
The University of Texas at Austin

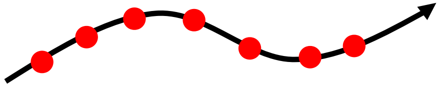
Asteroid Distributions

WHAT STARTS HERE CHANGES THE WORLD

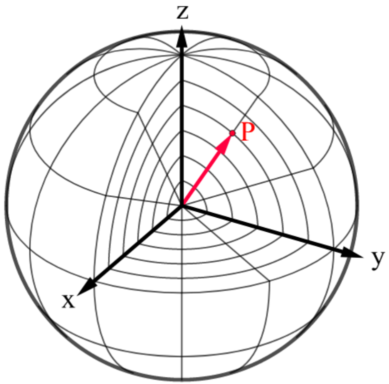


Mothership Trajectory Search

- Time discretization:
  - Increment all 20 years into 3 day steps
- Space discretization:
  - Spherical grid (radius, longitude, latitude)
  - 4 AU max radius: ~1 million bins
  - Low inclination: avoids issues at poles
- Precompute Asteroid/Earth Position Lookup Table:
  - Assign asteroids/Earth to spherical bins
  - Update bin contents for each 3 day step



time



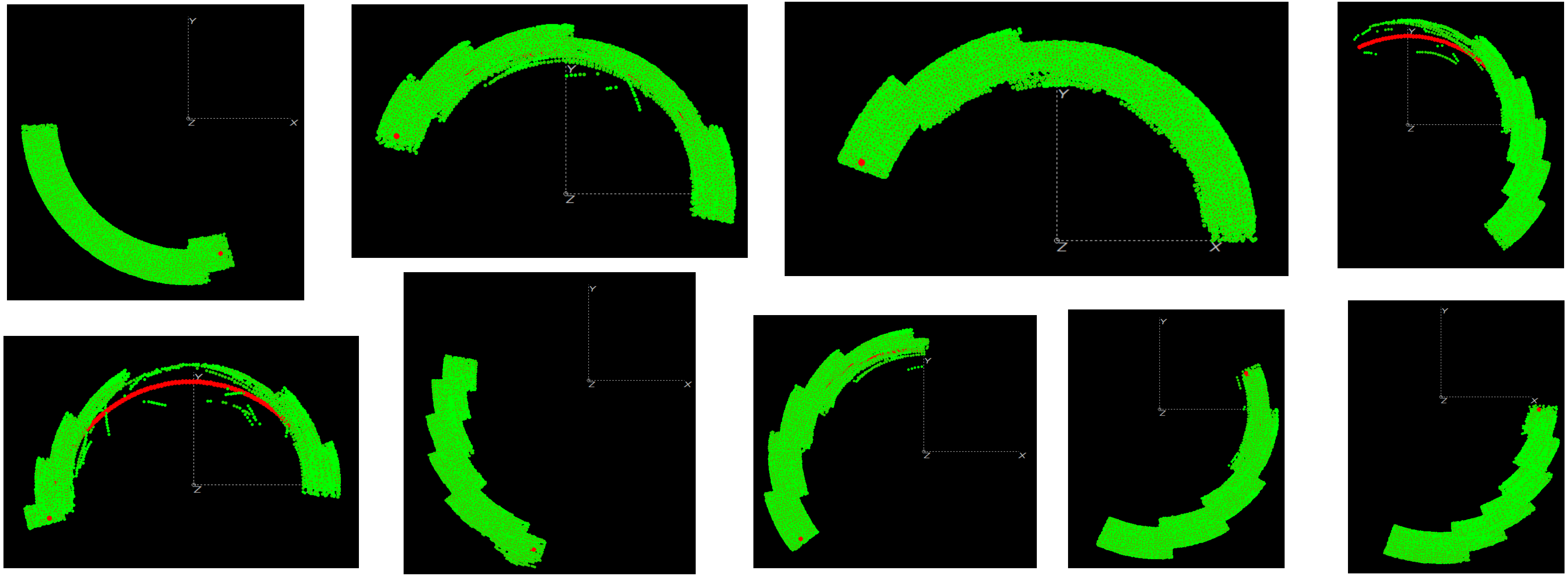
space

Mothership Trajectories

- Leg 1 Earth to Asteroids
  - From earliest t0 through the first x years (~1-2), for every time step compute minimum Δv solutions to every asteroid, filter out for
    - departure vinf too big (e.g. >8km/s, knowing above 6 km/s requires dv)
    - Arrival vinf too big (e.g. >4km/s, knowing above 2 km/s requires dv)
  - Store all N resulting options for ballistic Earth to Asteriod1, each completely defined
- Asteroids1 to Asteroid M
  - After each flyby use full 6 state to propagate
  - At each 3 day step:
    - Identify your s/c current bin
    - Identify all asteroids in the current bin and 8 neighboring bins
  - Find the union of all sets of asteroids encountered until some max leg TOF.
  - For each asteroid in the union set
    - Filter out based on geometry, simple OE filters
    - For remaining ones do a Δv minimization of all TOF up to the max for that leg. Record the 'best' {dv,TOF} for each asteroid
  - Choose from option asteroids using an 'extrapolated cost' idea. Assuming all remaining asteroids have same {dv,TOF}, compute the final Performance index
  - IF no options, THEN repeat prior leg with longer TOF. IF no options persist for N tries or if 20 year end of time is met. THEN terminate mothership.
  - Repeat process to find many mothership itineraries. Pick 10 good ones later.

More Example Look Ahead Options

Red: Mothership (covered up by green usually)
Green: Asteroids in mothership bin or neighboring bins
Dark yellow: asteroid that passes {dv,**e**,**h**} filter



GTOC min time indirect method derivation

Constant thrust acceleration

minimize: $J = t_f - t_0$

where states $\mathbf{x} \equiv \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix}$ and controls $\mathbf{u} = \left[\hat{\mathbf{q}} \right]$

are subject to dynamics: $\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{a} + T_{acc}\hat{\mathbf{q}} \end{cases}$ and constraints: $\begin{cases} \boldsymbol{\theta}\left(\mathbf{x}_0, t_0\right) = 0 \\ \boldsymbol{\psi}\left(\mathbf{x}_f, t_f\right) = 0 \end{cases}$

Augment the performance index:

$$J' = t_f - t_0 + \boldsymbol{\upsilon}_a{}^T \boldsymbol{\theta}\left(\mathbf{x}_0, t_0\right) + \boldsymbol{\upsilon}_b{}^T \boldsymbol{\psi}\left(\mathbf{x}_f, t_f\right) + \int_{t_0}^{t_f} \left[\boldsymbol{\lambda}^T \left(\mathbf{f}\left(\mathbf{x}, \mathbf{u}, t\right) - \dot{\mathbf{x}} \right) \right] dt$$

DEFINE: $\begin{cases} G\left(\mathbf{x}_f, t_f, t_0, \mathbf{v}\right) \equiv t_f - t_0 + \boldsymbol{\upsilon}_a{}^T \boldsymbol{\theta}\left(\mathbf{x}_0, t_0\right) + \boldsymbol{\upsilon}_b{}^T \boldsymbol{\psi}\left(\mathbf{x}_f, t_f\right) \\ H\left(\mathbf{x}, \mathbf{u}, t, \boldsymbol{\lambda}\right) \equiv \boldsymbol{\lambda}^T \mathbf{f}\left(\mathbf{x}, \mathbf{u}, t\right) = \boldsymbol{\lambda}_{\mathbf{r}}{}^T \mathbf{v} + \boldsymbol{\lambda}_{\mathbf{v}}{}^T \left(\mathbf{a} + T_{acc}\hat{\mathbf{q}}\right) \end{cases}$

$$\rightarrow \boxed{J' = G\left(\mathbf{x}_f, t_f, t_0, \mathbf{v}\right) + \int_{t_0}^{t_f} \left[H\left(\mathbf{x}, \mathbf{u}, t, \boldsymbol{\lambda}\right) - \boldsymbol{\lambda}^T \dot{\mathbf{x}} \right] dt}$$

Minimize Hamiltonian with respect to controls:

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T (\mathbf{a} + T_{acc}\hat{\mathbf{q}})$$

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{a} + \boldsymbol{\lambda}_v^T T_{acc}\hat{\mathbf{q}}$$

$$\text{by inspection to min } H \rightarrow \boxed{\hat{\mathbf{q}} = (-\boldsymbol{\lambda}_v / \lambda_v)}$$

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{a} + \boldsymbol{\lambda}_v^T T_{acc} (-\boldsymbol{\lambda}_v / \lambda_v)$$

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{a} - \lambda_v (\boldsymbol{\lambda}_v / \lambda_v)^T T_{acc} (\boldsymbol{\lambda}_v / \lambda_v)$$

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{a} - \lambda_v \left(\hat{\boldsymbol{\lambda}}_v \cdot \hat{\boldsymbol{\lambda}}_v\right) T_{acc}$$

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{a} - \lambda_v^2 T_{acc} (1 / \lambda_v)$$

$$\boxed{H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{a} - \lambda_v T_{acc}}$$

$$\boxed{\dot{H} = 0} \text{ since no } t \text{ explicitly in } \mathbf{a} \text{ and } H_{\hat{\mathbf{q}}} \dot{\hat{\mathbf{q}}} = 0$$

$$H_{\hat{\mathbf{q}}} \dot{\hat{\mathbf{q}}} = (\boldsymbol{\lambda}_v^T T_{acc}) \dot{\hat{\mathbf{q}}} = (-\lambda_v \hat{\mathbf{q}}^T T_{acc}) \dot{\hat{\mathbf{q}}} = -\lambda_v T_{acc} (\hat{\mathbf{q}} \cdot \dot{\hat{\mathbf{q}}}) = -\lambda_v T_{acc} \left(\|\hat{\mathbf{q}}\| \cdot \frac{d\|\hat{\mathbf{q}}\|}{dt} \right) = -\lambda_v T_{acc} (1 \cdot 0) = 0$$

Use Pontryagin or Weierstrass to choose **u** that globally optimizes H
(because control terms appear linearly in H)

Show H is still an integral of motion if no t in EOM

Boundary Conditions

minimize TOF
fixed initial position & velocity to start orbit
fixed final position & velocity to final orbit

$$
\begin{aligned}
G \equiv\ & t_f - t_0 \\
&+ \boldsymbol{\eta}_r^T\left(\mathbf{r}_0 - \mathbf{r}_D(t_0)\right) + \boldsymbol{\eta}_v^T\left(\mathbf{v}_0 - \mathbf{v}_D(t_0)\right) \\
&+ \mathbf{v}_r^T\left(\mathbf{r}_f - \mathbf{r}_A(t_0)\right) + \mathbf{v}_v^T\left(\mathbf{v}_f - \mathbf{v}_A(t_f)\right)
\end{aligned}
$$

$$
\begin{cases}
\boldsymbol{\lambda}_0^T = -G_{\mathbf{x}_0} \rightarrow \begin{cases} \boldsymbol{\lambda}_{\mathbf{r}_0}^T = -G_{\mathbf{r}_0} = -\boldsymbol{\eta}_r^T \\ \boldsymbol{\lambda}_{\mathbf{v}_0}^T = -G_{\mathbf{v}_0} = -\boldsymbol{\eta}_v^T \end{cases} \\
\boldsymbol{\lambda}_f^T = G_{\mathbf{x}_f} \rightarrow \begin{cases} \boldsymbol{\lambda}_{\mathbf{r}_f}^T = G_{\mathbf{r}_f} = \mathbf{v}_r^T \\ \boldsymbol{\lambda}_{\mathbf{v}_f}^T = G_{\mathbf{v}_f} = \mathbf{v}_v^T \end{cases}
\end{cases}
$$

$$
\begin{cases}
H_0 = \dfrac{\partial G}{\partial t_0} \\
\dfrac{\partial G}{\partial t_0} = -1 - \boldsymbol{\eta}_r^T \dfrac{\partial \mathbf{r}_D(t_0)}{\partial t_0} - \boldsymbol{\eta}_v^T \dfrac{\partial \mathbf{v}_D(t_0)}{\partial t_0} \\
\qquad = -1 + \boldsymbol{\lambda}_{\mathbf{r}_0}^T \dfrac{\partial \mathbf{r}_D(t_0)}{\partial t_0} + \boldsymbol{\lambda}_{\mathbf{v}_0}^T \dfrac{\partial \mathbf{v}_D(t_0)}{\partial t_0} \\
H_0 = \boldsymbol{\lambda}_{\mathbf{r}_0}^T \mathbf{v}_0 + \boldsymbol{\lambda}_{\mathbf{v}_0}^T \mathbf{a}_0 - \lambda_{\mathbf{v}_0} T_{acc} \\
\boldsymbol{\lambda}_{\mathbf{r}_0}^T \mathbf{v}_0 + \boldsymbol{\lambda}_{\mathbf{v}_0}^T \mathbf{a}_0 - \lambda_{\mathbf{v}_0} T_{acc} = -1 + \boldsymbol{\lambda}_{\mathbf{r}_0}^T \mathbf{v}_0 + \boldsymbol{\lambda}_{\mathbf{v}_0}^T \mathbf{a}_0 \\
-\lambda_{\mathbf{v}_0} T_{acc} = -1 \\
\boxed{\lambda_{\mathbf{v}_0} = \dfrac{1}{T_{acc}}}
\end{cases}
$$

$$
\begin{cases}
H_f = -\dfrac{\partial G}{\partial t_f} \\
-\dfrac{\partial G}{\partial t_f} = -1 + \mathbf{v}_r^T \dfrac{\partial \mathbf{r}_A(t_f)}{\partial t_f} + \mathbf{v}_v^T \dfrac{\partial \mathbf{v}_A(t_f)}{\partial t_f} \\
H_f = \boldsymbol{\lambda}_{\mathbf{r}_f}^T \mathbf{v}_f + \boldsymbol{\lambda}_{\mathbf{v}_f}^T \mathbf{a}_f - \lambda_{\mathbf{v}_f} T_{acc} \\
\boxed{\lambda_{\mathbf{v}_f} = \dfrac{1}{T_{acc}}}
\end{cases}
$$

How to implement with forward single shooting:

UNKNOWNS (8):	CONSTRAINTS (8):
$\boldsymbol{\lambda}_{\mathbf{r}_0}$	$\mathbf{r}_f = \mathbf{r}_A(t_f)$
$\boldsymbol{\lambda}_{\mathbf{v}_0}$	$\mathbf{v}_f = \mathbf{v}_A(t_f)$
t_0	$\lambda_{\mathbf{v}_0} = \dfrac{1}{T_{acc}},\ \lambda_{\mathbf{v}_f} = \dfrac{1}{T_{acc}}$
t_f	

or auto-enforce initial magnitude of $\boldsymbol{\lambda}_{\mathbf{v}_0}$

UNKNOWNS (7):	CONSTRAINTS (7):
$\boldsymbol{\lambda}_{\mathbf{r}_0}$ (3)	$\mathbf{r}_f = \mathbf{r}_A(t_f)$
$\hat{\boldsymbol{\lambda}}_{\mathbf{v}_0}$ (2) where $\boldsymbol{\lambda}_{\mathbf{v}_0} = \dfrac{1}{T_{acc}} \hat{\boldsymbol{\lambda}}_{\mathbf{v}_0}$	$\mathbf{v}_f = \mathbf{v}_A(t_f)$
$t_0,\ t_f$ (2)	$\lambda_{\mathbf{v}_f} = \dfrac{1}{T_{acc}}$

note: EOM are invariant to length of $\boldsymbol{\lambda}$ so we can pick any scale

equivalent to solving $J = k(t_f - t_0)$ instead

UNKNOWNS (7):	CONSTRAINTS (7):
$\boldsymbol{\lambda}_{\mathbf{r}_0}$ (3)	$\mathbf{r}_f = \mathbf{r}_A(t_f)$
$\hat{\boldsymbol{\lambda}}_{\mathbf{v}_0}$ (2) where $\boldsymbol{\lambda}_{\mathbf{v}_0} = 1\, \hat{\boldsymbol{\lambda}}_{\mathbf{v}_0}$	$\mathbf{v}_f = \mathbf{v}_A(t_f)$
$t_0,\ t_f$ (2)	$\lvert\boldsymbol{\lambda}_{\mathbf{v}_f}\rvert = \lvert\boldsymbol{\lambda}_{\mathbf{v}_0}\rvert = 1$

for free t_0, but fixed t_f and $\phi = 0$, then no boundary condition on $\lvert\boldsymbol{\lambda}_{\mathbf{v}_f}\rvert$

still boundary condtion on $\lvert\boldsymbol{\lambda}_{\mathbf{v}_0}\rvert$ but due to scaling you can ignore