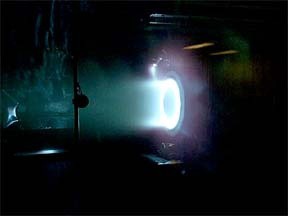
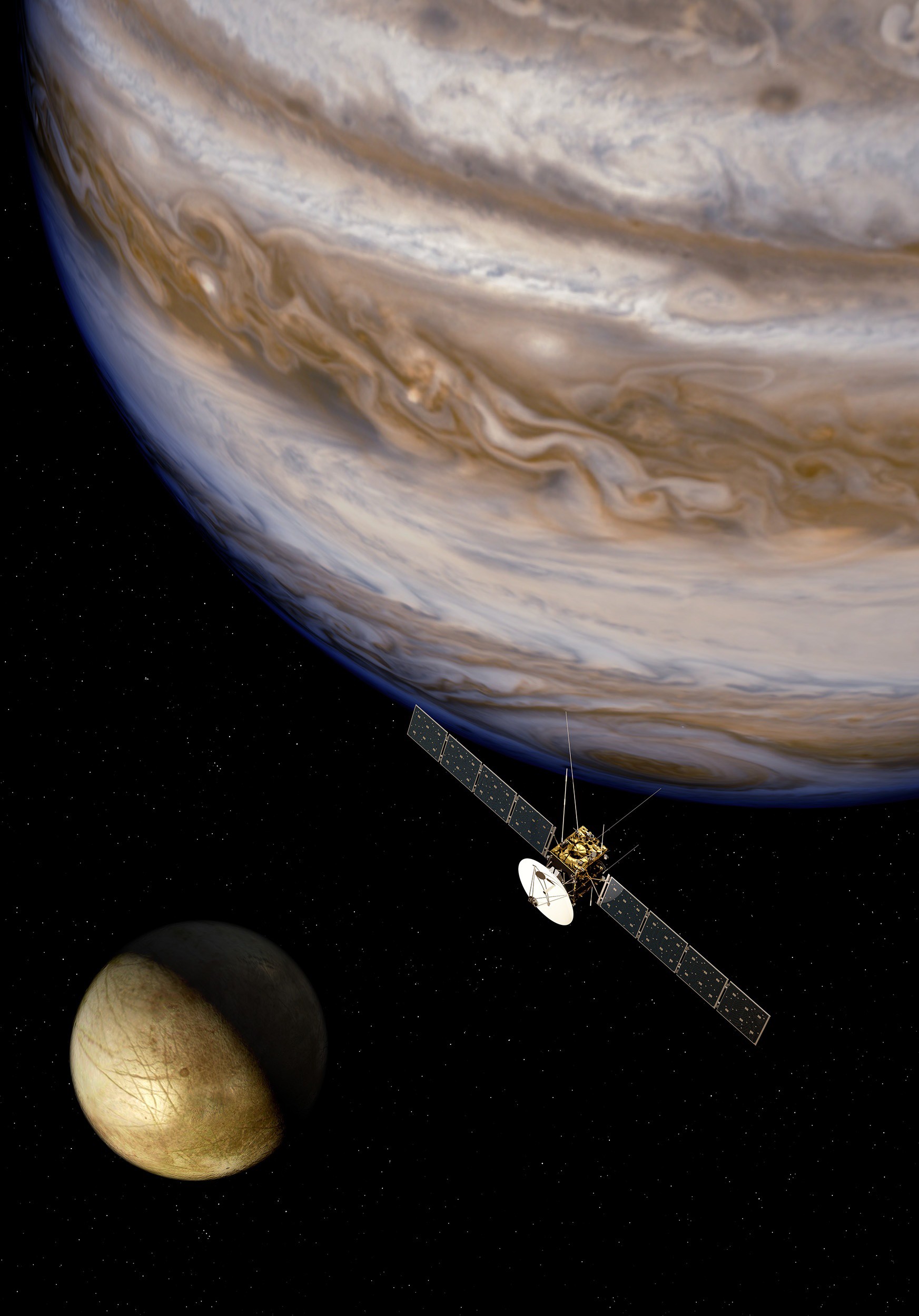
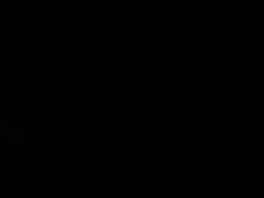
Heuristic Search



for Transfer Design

● Background:

● - Multibody systems are highly nonlinear and chaotic due to

balance of forces from planets and moons.

● - For “low energy” missions to orbit moons and small bodies, traditional methods are unable to provide reliable initial guesses to optimizers.

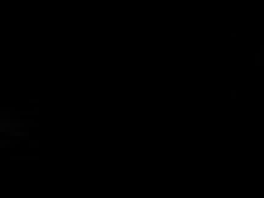
● Central Concept: Approximate system with directed graph. Use to create a transfer skeleton.

● - Partition domain into small regions. 1 region = 1 node

● - Ballistic dynamics: Add directed edge from node X to node Y if region X flows into node Y. Weight = nominal control cost

● - Impulsive maneuvers: Add directed edge between nodes with same position but with different energy levels or headings. Weight is minimum fuel cost between these pairs.

Using A\* Search



in Itinerary Selection

● Need for Heuristics:

● - For minimum fuel problem, cost of large impulse >> nominal coasting cost.

● - Most important scenario: Orbit insertion from approach or higher orbit. Requires large drop in energy.

● - Result w/ uninformed search: Search explores too many coasting nodes before considering necessary impulsive maneuvers.

● Simplified Energy Heuristic:

● - Between any two energy levels with a given compact domain and moon radius

there is a calculable most efficient maneuver cost.

● [This is a tangential maneuver at the highest velocity and/or lowest potential]

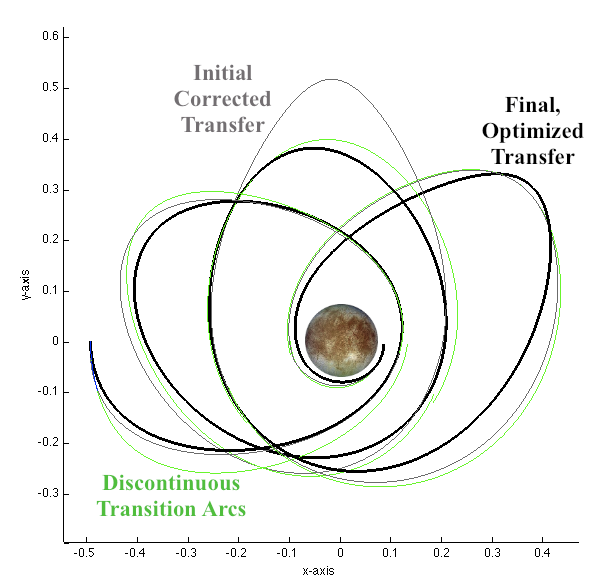
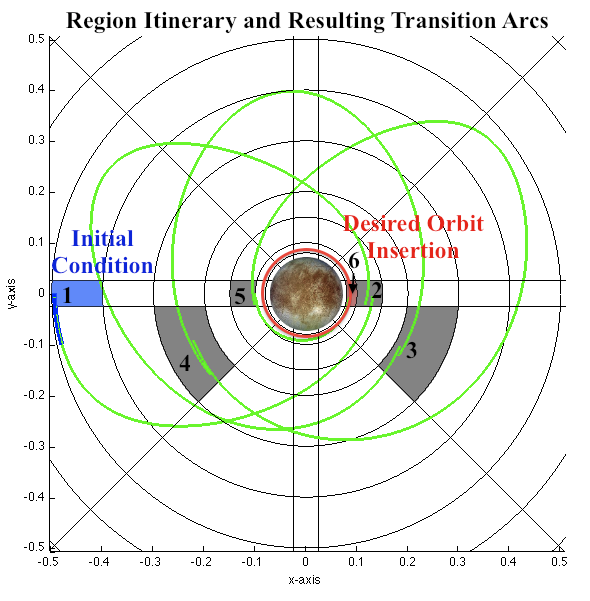
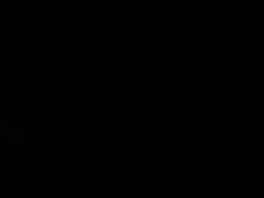
● - Heuristic as rule relaxation: it assumes this most efficient maneuver is possible everywhere in the domain.

● - This makes h(n) a simple (consistent) function of energy(n') and energy(goal).

● - Heuristic steers search towards correct energy.

●

Applying the Search Results



● Impact of the heuristic: Decrease in search times from ~2-3 minutes to ~1-10 seconds in simple test system.

● Example in context: Search provides region and maneuver sequence. This allows for selection of arcs from a table which feed into a Sequential Convex Programming process to restore continuity and optimize.

● [Note: the actual partition is 4D and contains ~9,000 regions per energy]