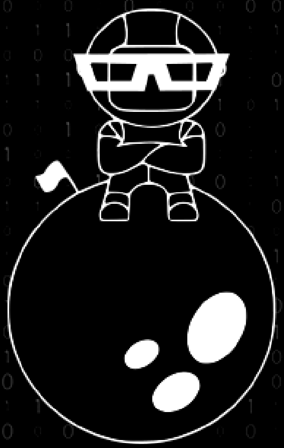


Trajectories Design

SPACE INFORMATICS

SIC Saarland Informatics Campus

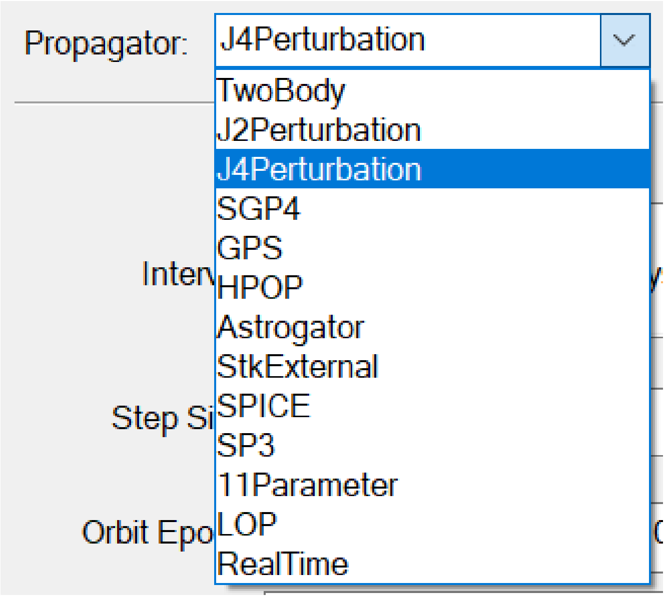
Propagators

Types of Propagators

- Propagators on STK
  - **Two Body:** Analytical
  - **J2/J4 Perturbation:** Analytical
  - **SGP4:** Analytical (TLE)
  - **GPS:** Two-Body propagator
  - **11Parameter:** GEO propagation
  - **LOP:** Long-Term orbit
  - **HPOP:** High-Precision Propagator
  - **External:** From ephemeris file
  - **SPICE:** From SPICE toolkit (JPL)
  - **SP3:** From NGS file
  - **Real Time:** From feed of telemetry

Fixed orbit type

External input

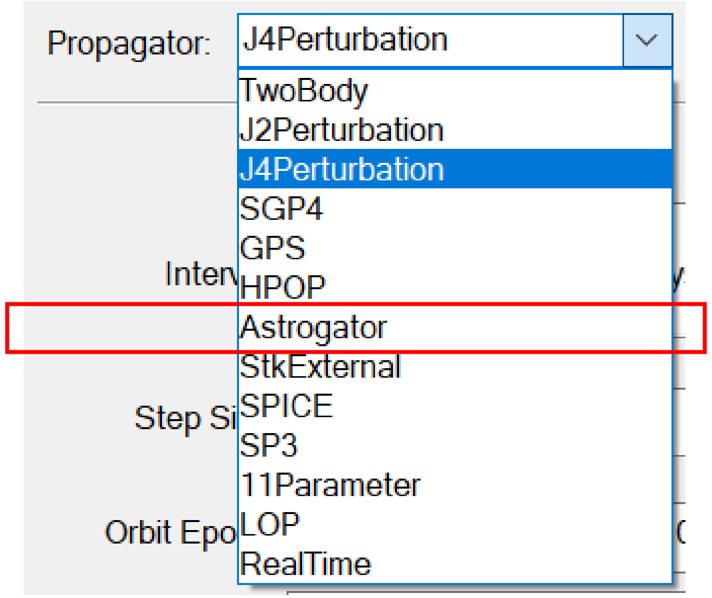


http://help.agi.com/stk/11.2/index.htm#stk/ObjectMap_Vehicle_Satellite.htm

Astrogator

Introduction

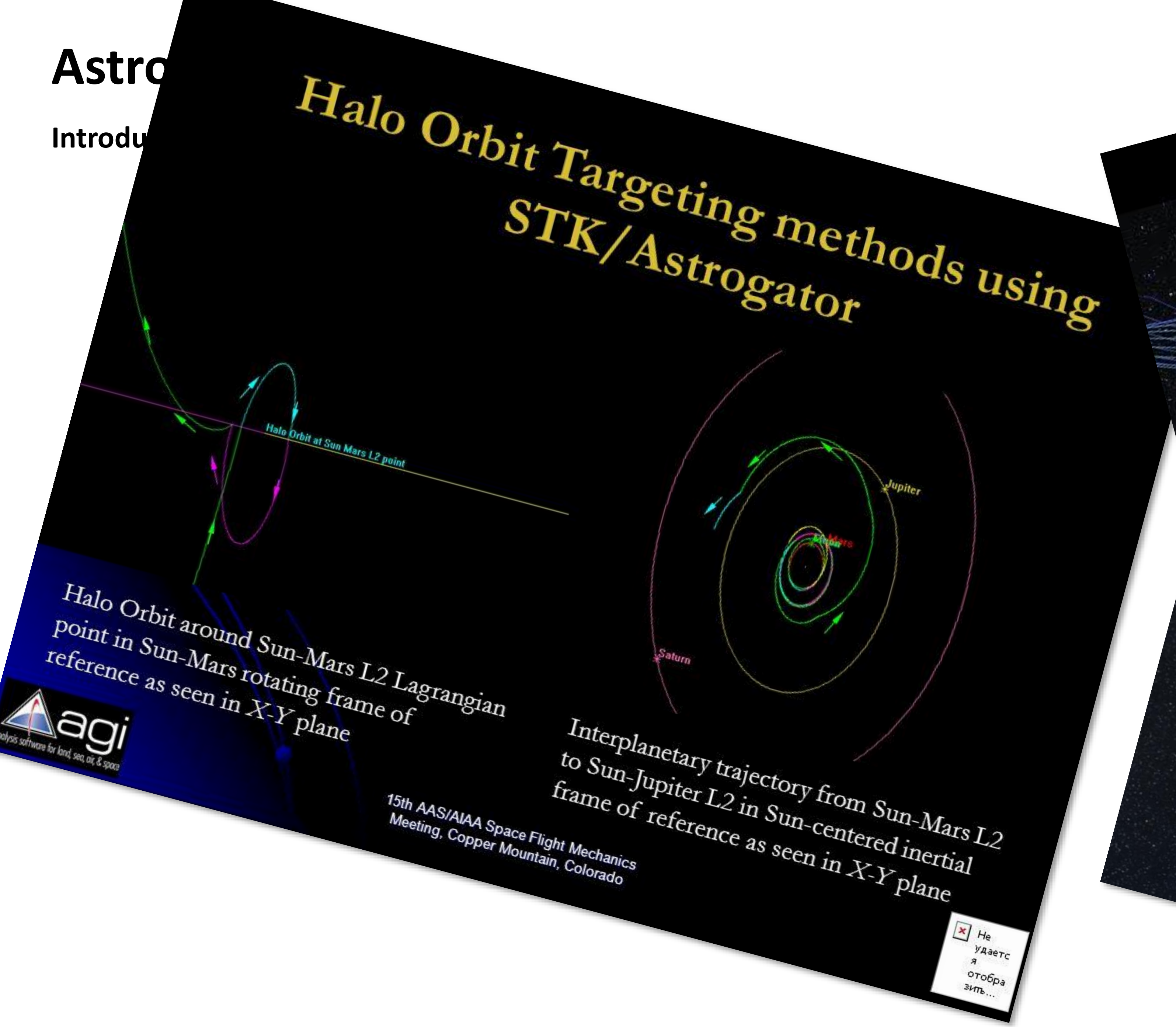
- Astrogator
  - Interactive orbit maneuver and trajectory design
  - **Events** for modeling and targeting trajectories
    - **Impulsive** and **finite** burns
    - High-fidelity orbit **propagation**
    - **Target** specified and optimized orbit states

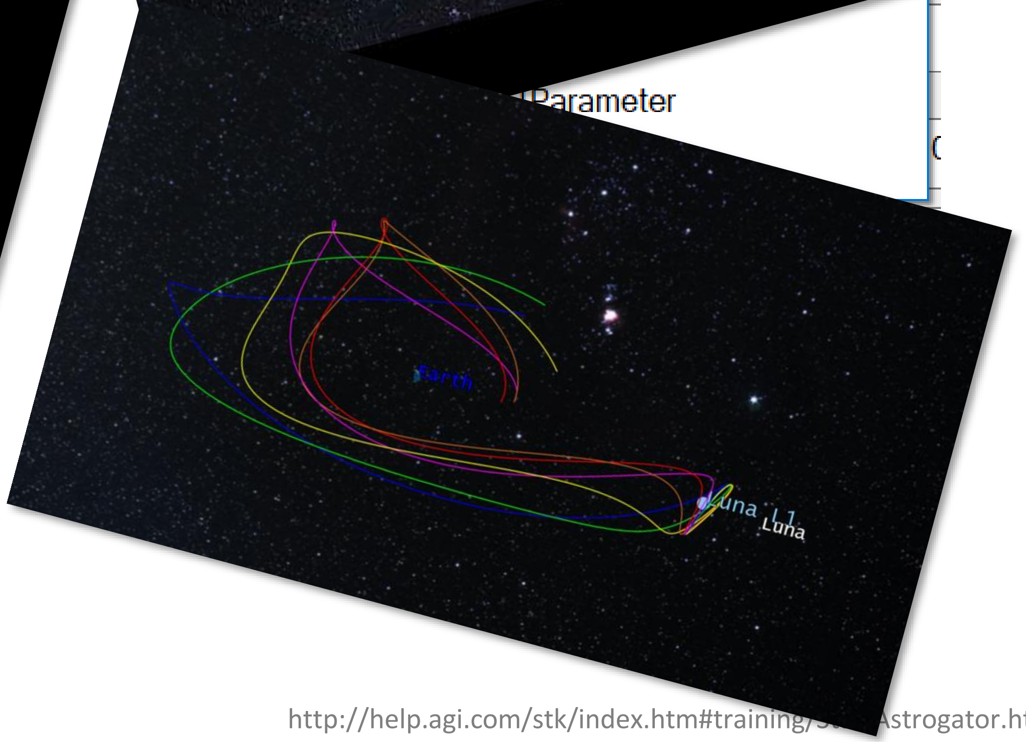


http://help.agi.com/stk/index.htm#training/StartAstrogator.htm

Astro

Introdu





http://help.agi.com/stk/index.htm#training/S... Astrogator.htm

- **Mission Control Sequence**
  - Defines the **trajectory** as a **sequence** of events a.k.a. “segments”
  - Functions as a **graphical programming** language
  - Each segment dictates how Astrogator calculates the trajectory before passing the spacecraft’s state on the next segment

Astrogator

Mission Control Sequence

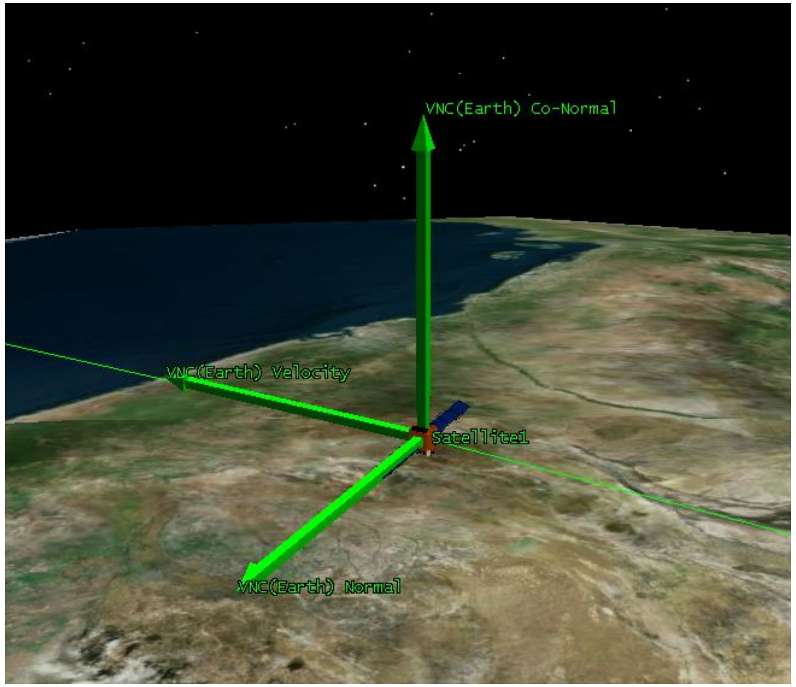
- Mission Control Sequence Segments
  - **Initial State** ( ) – This segment can be used to define the initial conditions of your MCS or a subsequence within the MCS.
  - **Launch** ( ) – This segment can be used to model a simple spacecraft launch from Earth or another central body.
  - **Follow** ( ) – This segment can set the spacecraft to follow another vehicle and separate from that vehicle upon meeting specified conditions.
  - **Maneuver** ( ) – This segment can be used to model a spacecraft maneuver.
  - **Propagate** ( ) – This segment can be used to model the movement of the spacecraft along its current trajectory until meeting specified stopping conditions

http://help.agi.com/stk/index.htm#gator/mc-segmentsintro.htm

Astrogator

Mission Control Sequence - Maneuver

- **Maneuver Attitude Control.**
  - **Along Velocity Vector:** Attitude is such that the delta-V vector is aligned with the spacecraft’s inertial velocity vector.
  - **Anti-Velocity Vector:** Attitude is such that the delta-V vector is opposite to the spacecraft’s inertial velocity vector.
  - **Attitude:** Attitude is defined using Euler Angles or a Quaternion.
  - **File:** Import an attitude file to set the maneuver.
  - **Thrust Vector:** The total delta-V can be specified in cartesian or spherical form with respect to the thrust axes.
    - **VNC** = Velocity, Normal, Co-Normal



http://help.agi.com/stk/index.htm#gator/mc-segmentsintro.htm

Astrogator

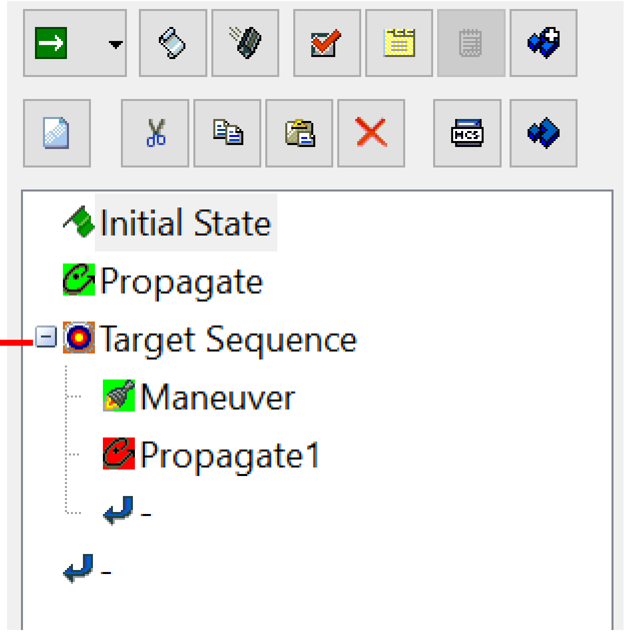
Target Sequences

- **Target Sequences** are used to
  - Calculate and define the required maneuver characteristics…
  - **to meet specified or search for optimal mission parameters**.

A target sequence runs the **segments nested within it**

Applies **profiles** to the run according to its configuration.

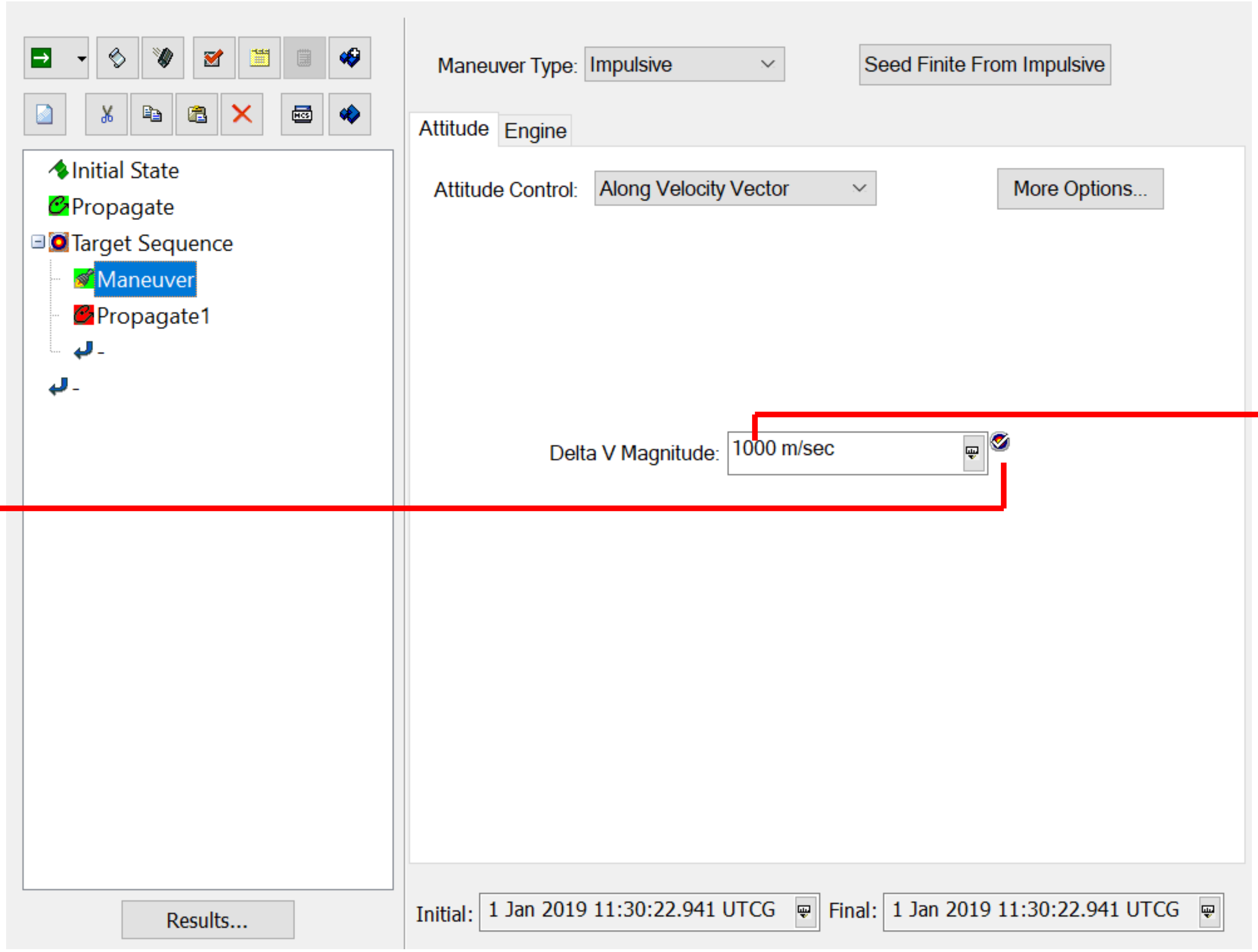
Like a “for” loop in a programming language



Astrogator

Target Sequences – Control Parameters

To define **control parameters** configuration fields from elements within a target sequence can selected and enabled



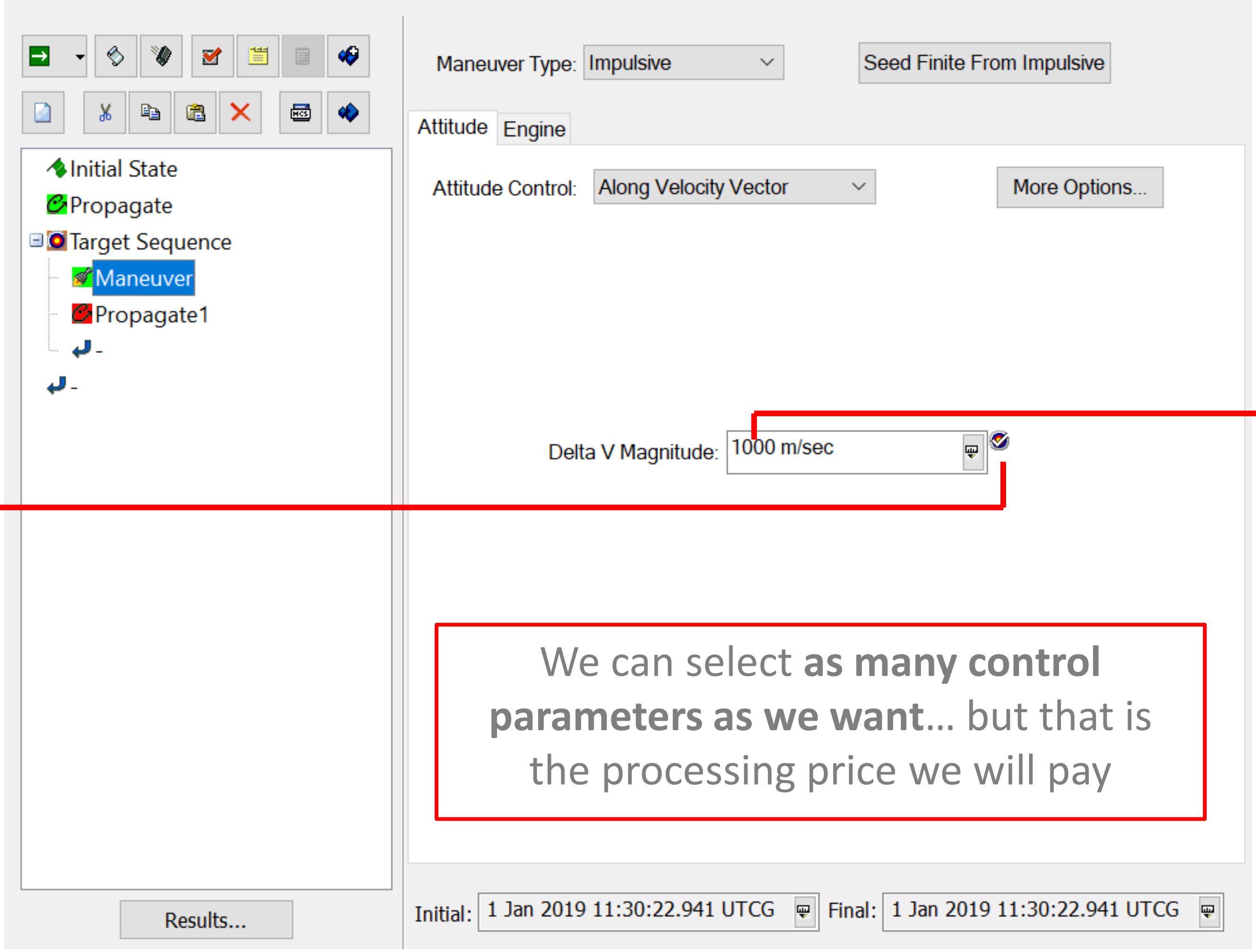
These are the **independent variables**... what we allow the target sequence to **change.**

The configured value is taken as **initial search value**... it is important to help with a good guess!

Astrogator

Target Sequences – Control Parameters

To define **control parameters** configuration fields from elements within a target sequence can selected and enabled



We can select **as many control parameters as we want**… but that is the processing price we will pay

These are the **independent variables**… what we allow the target sequence to **change.**

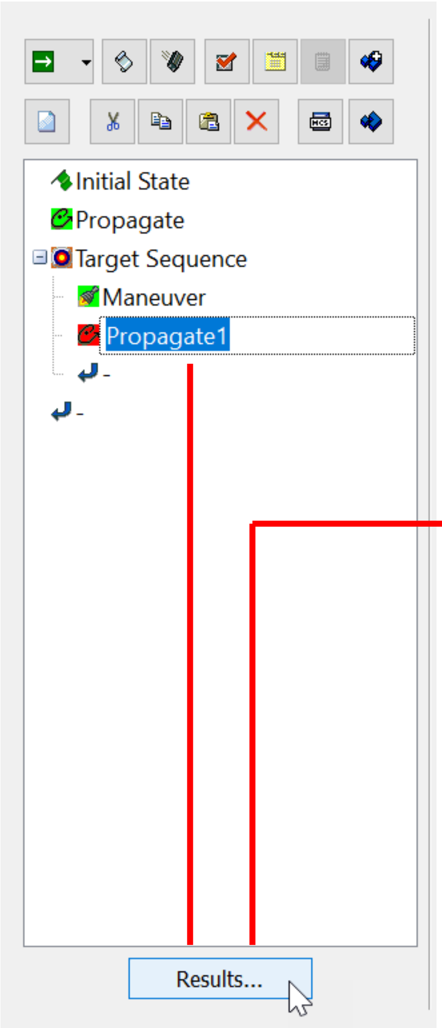
The configured value is taken as **initial search value**… it is important to help with a good guess!

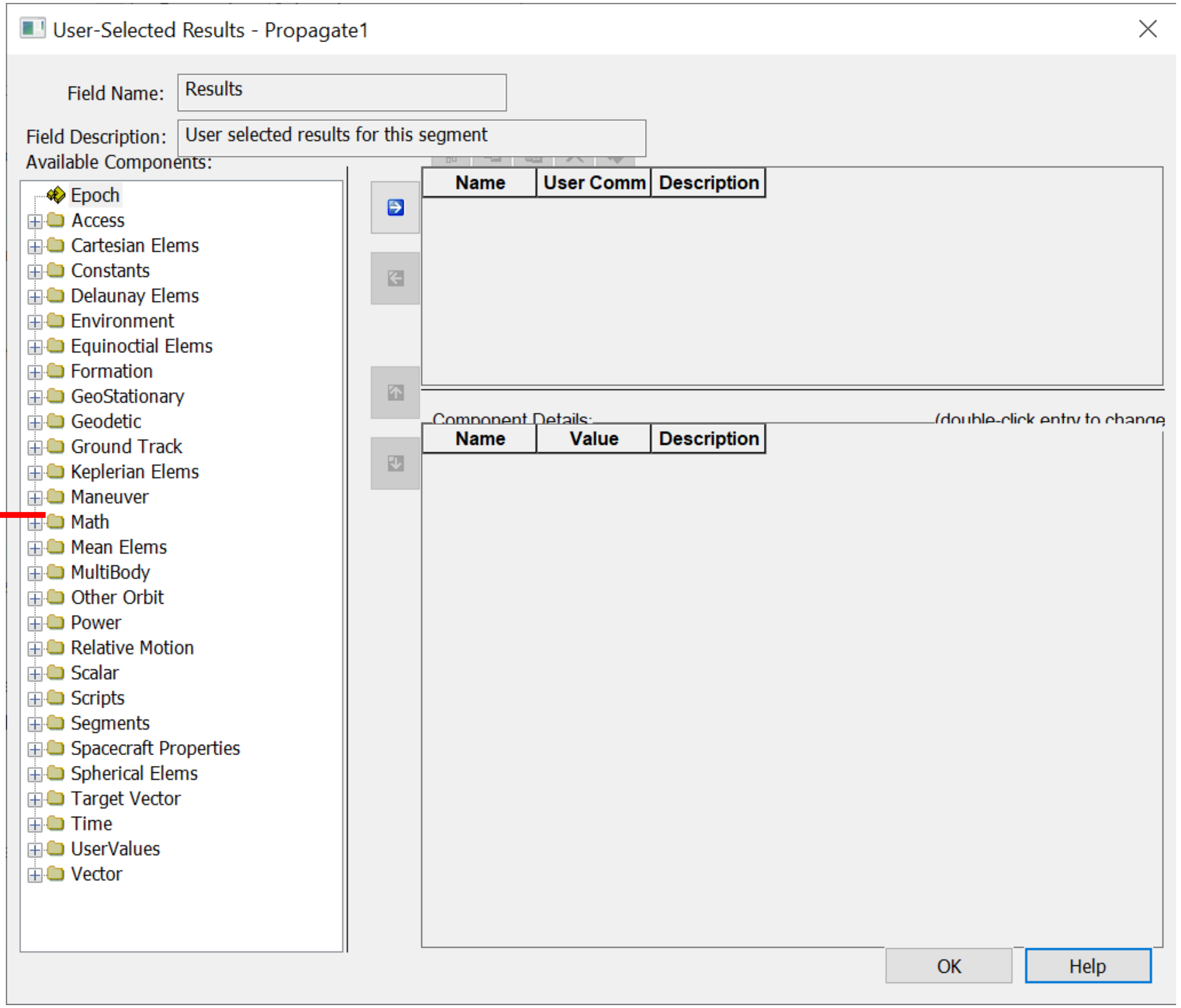
Astrogator

Target Sequences – Results

To define the **target sequence objective**:

1) Select the segment where the objective is expected to happen
2) Open "Results" menu on the bottom





Multi-component select window

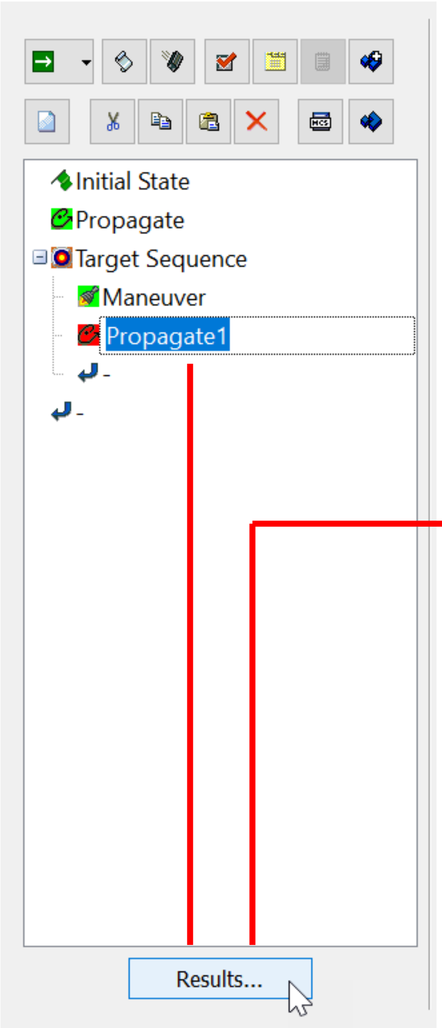
This are our **dependent variables**.

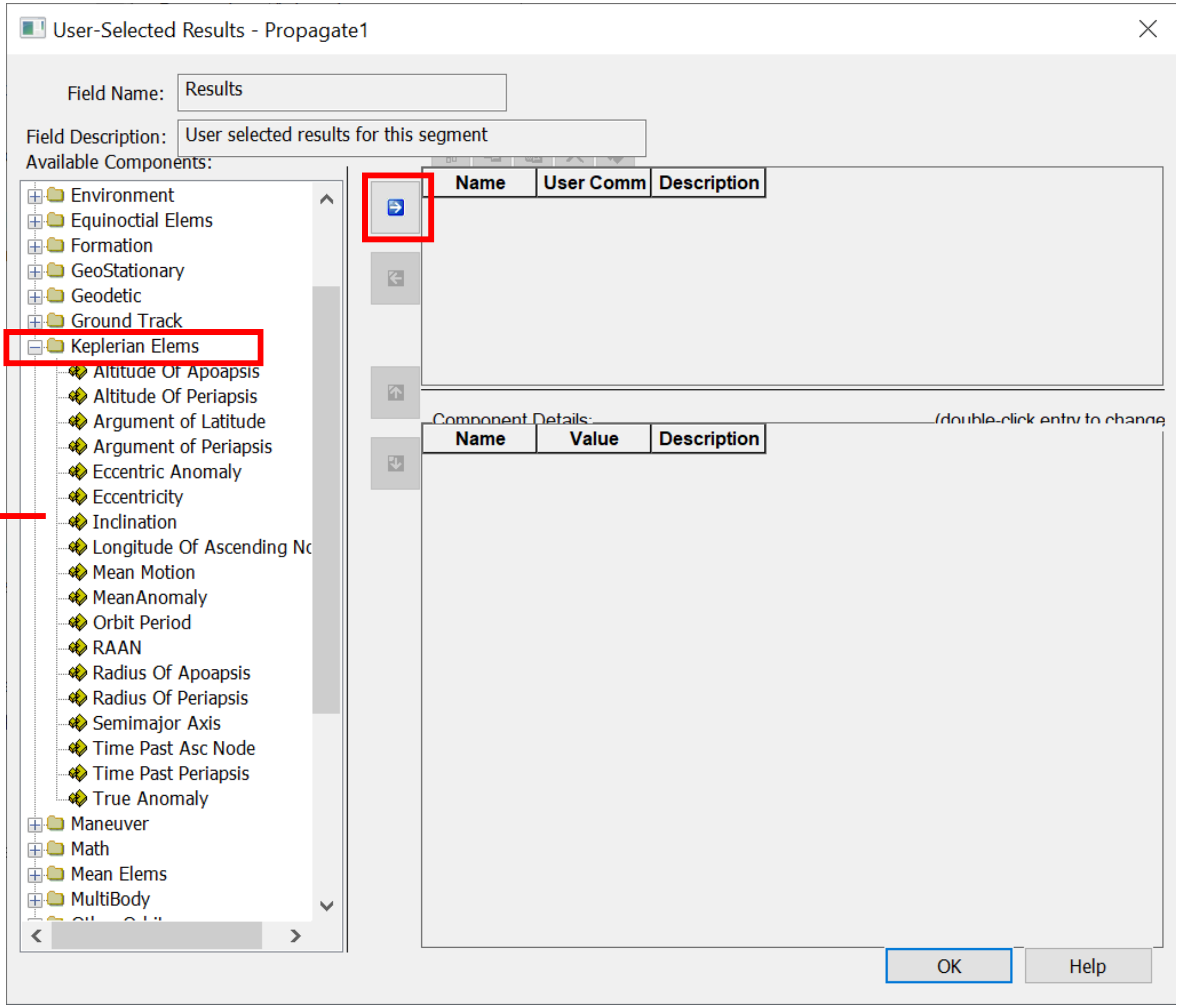
Astrogator

Target Sequences – Results

To define the **target sequence objective**:

1) Select the segment where the objective is expected to happen
2) Open “Results” menu on the bottom





Multi-component select window
- Keplerian elements should be familiar

Select elements which corresponds to our target objective and close.

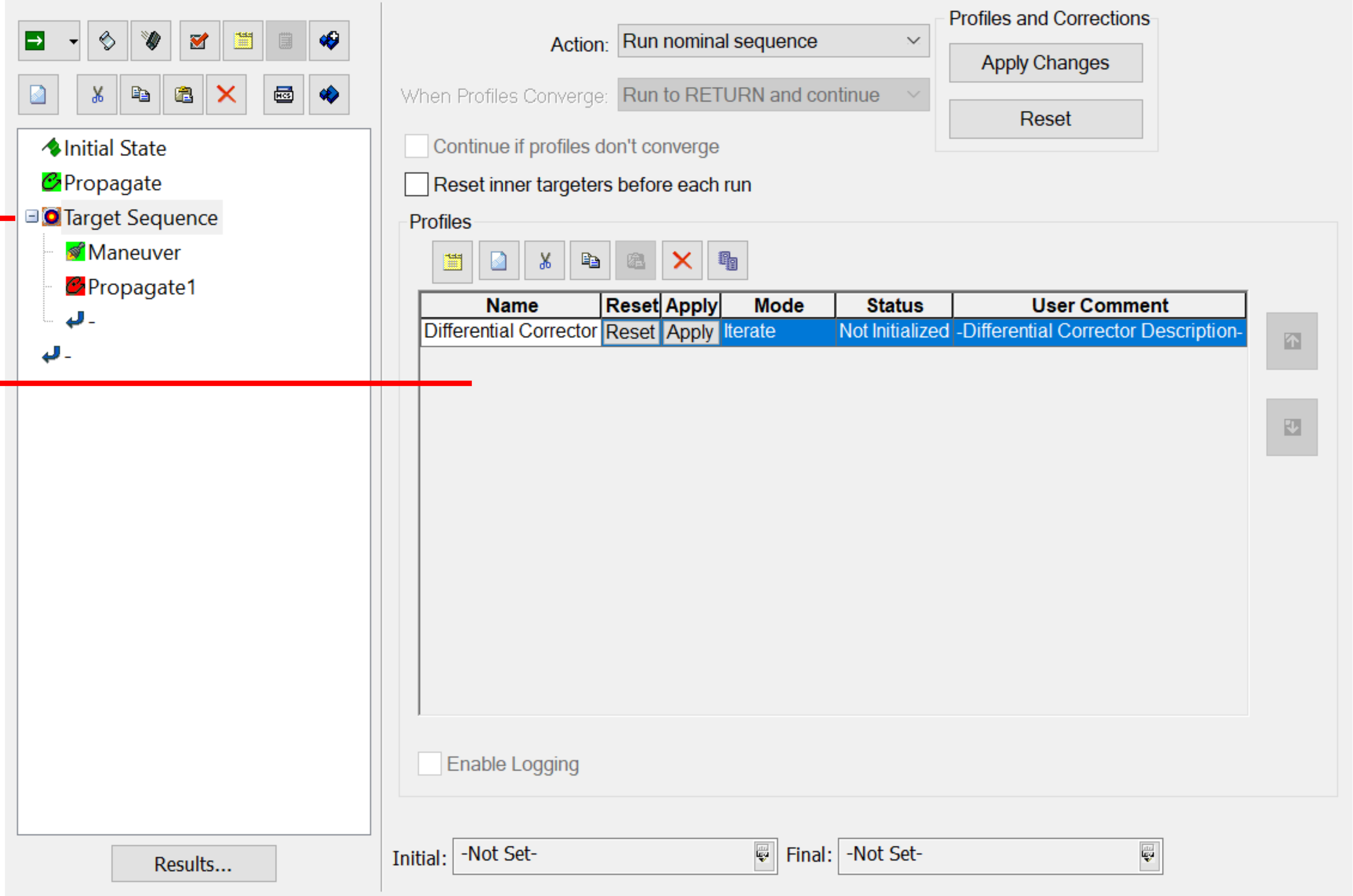
This are our **dependent variables**.

Astrogator

Target Sequences – Profiles

To define the **target sequence behavior** a **profile** need to be defined.

Select the target sequence segment and configure it

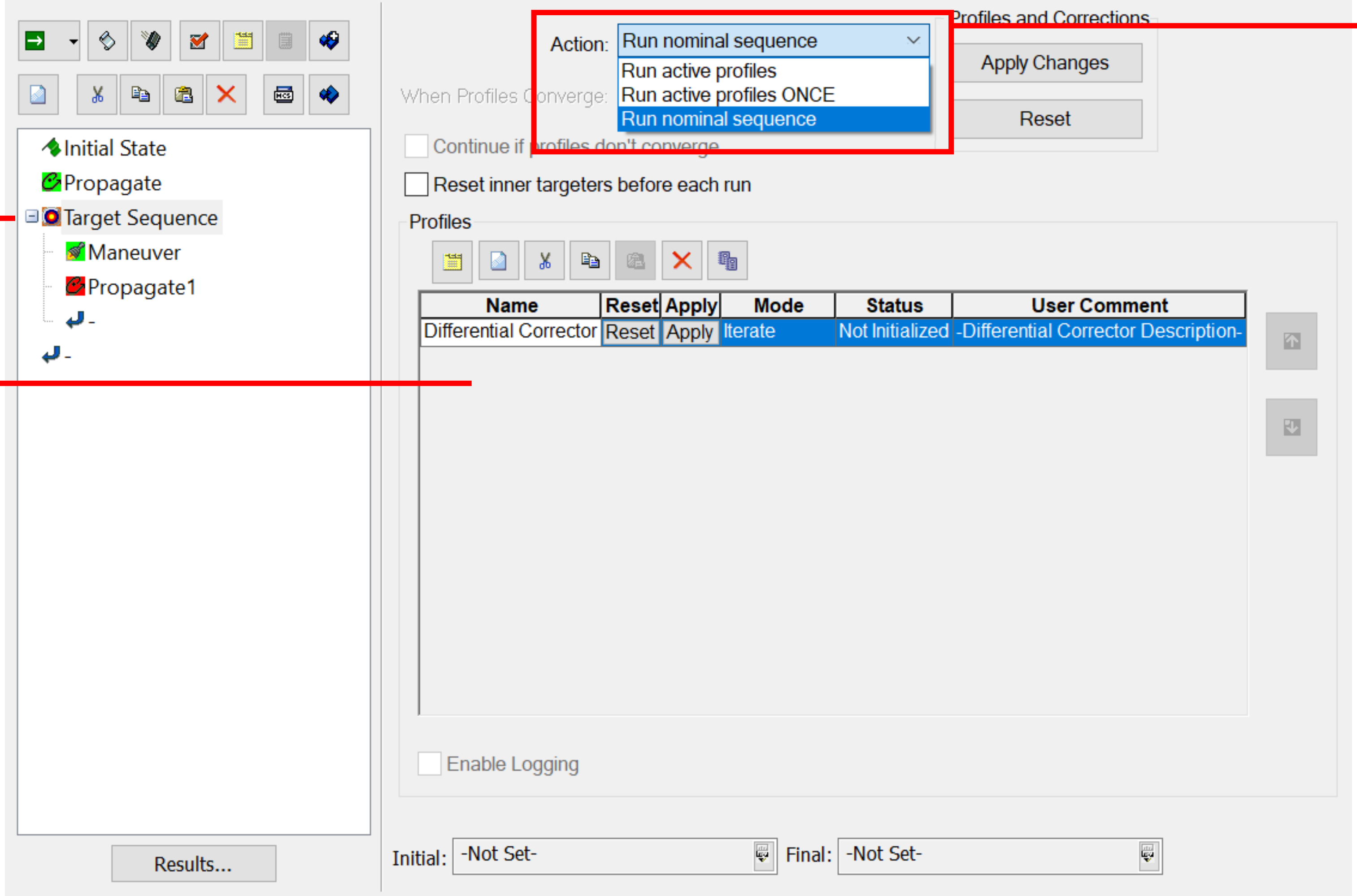


Astrogator

Target Sequences – Profiles

To define the **target sequence behavior** a **profile** need to be defined.

Select the target sequence segment and configure it



Action:

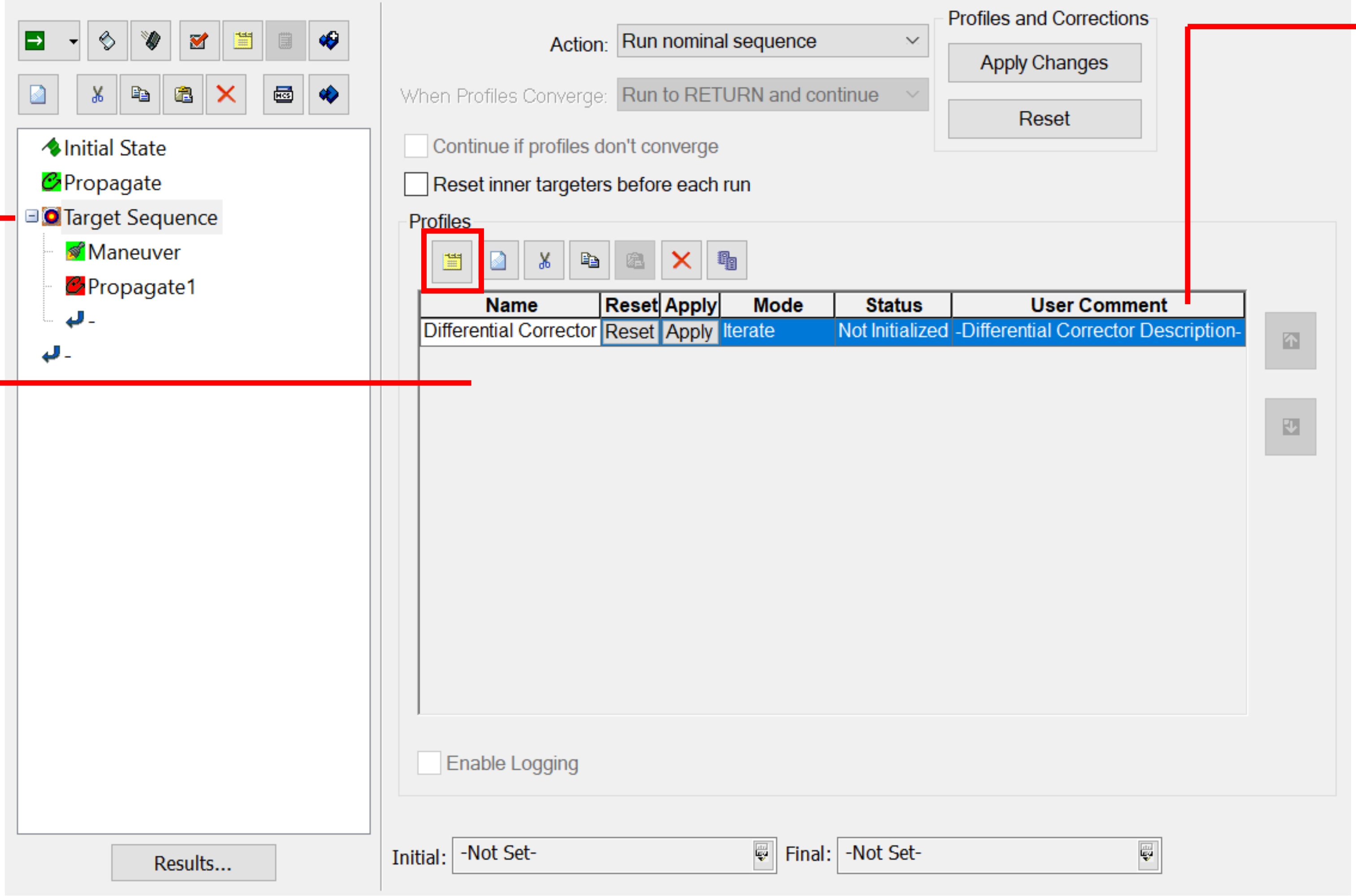
- Run nominal sequence
- Run active profiles
- Run active profiles once

Astrogator

Target Sequences – Profiles

To define the **target sequence behavior** a **profile** need to be defined.

Select the target sequence segment and configure it



By default, a **differential corrector** will be present as a profile.

Astrogator

Target Sequences – Profiles – Differential Corrector

- The differential corrector profile targets specific values defined as **independent variables**
- The target sequence will change the value of independent variables as needed to achieve the goal defined by the **dependent variables**
- STK will utilize a **differential correction algorithm**
  - Numerical search algorithm

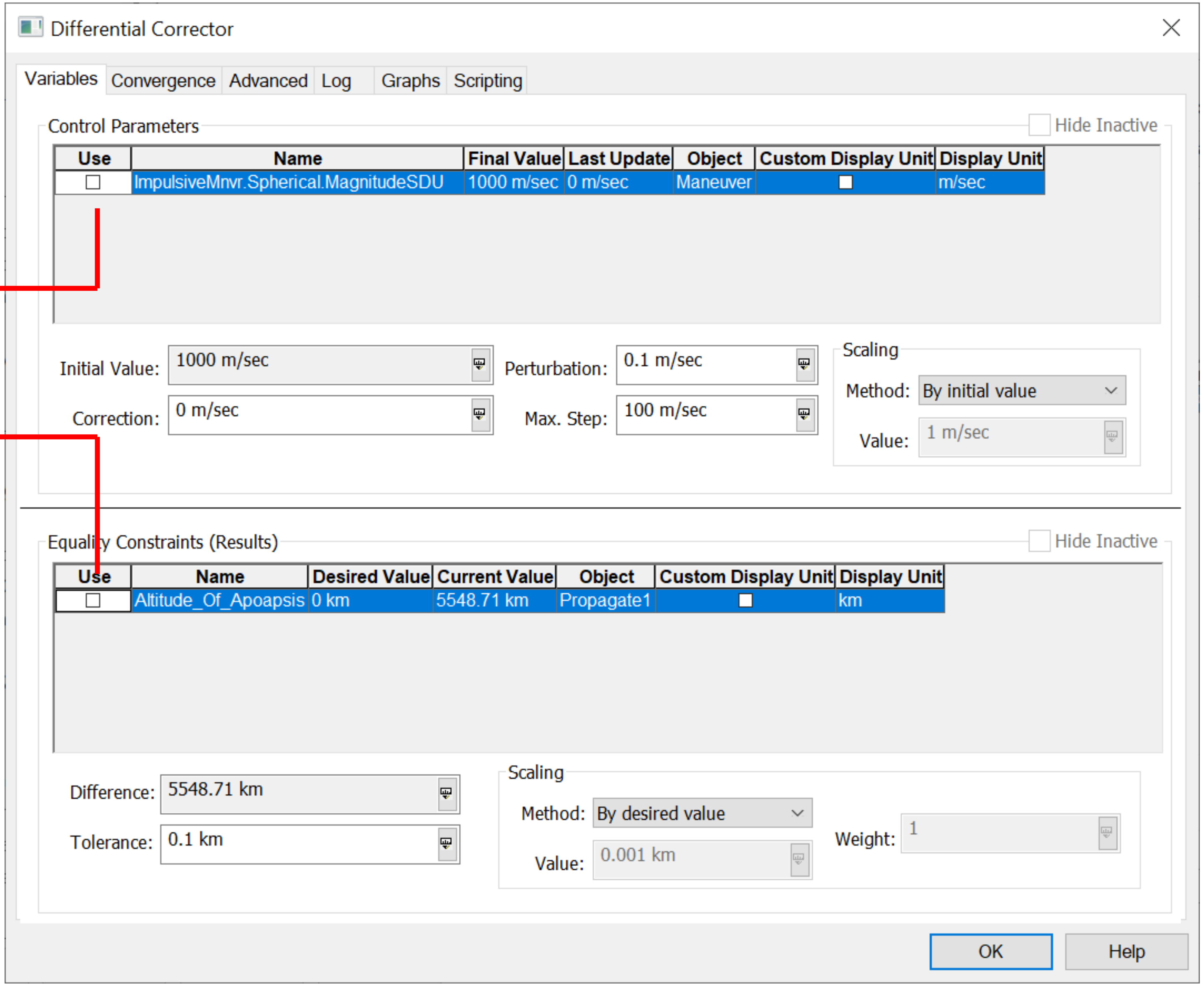
http://help.agi.com/stk/11.2/index.htm#gator/ab-targetprofile.htm

Astrogator

Target Sequences – Profiles – Differential Corrector

Differential corrector config:

- List of **independent variables**
- List of **dependent variables**

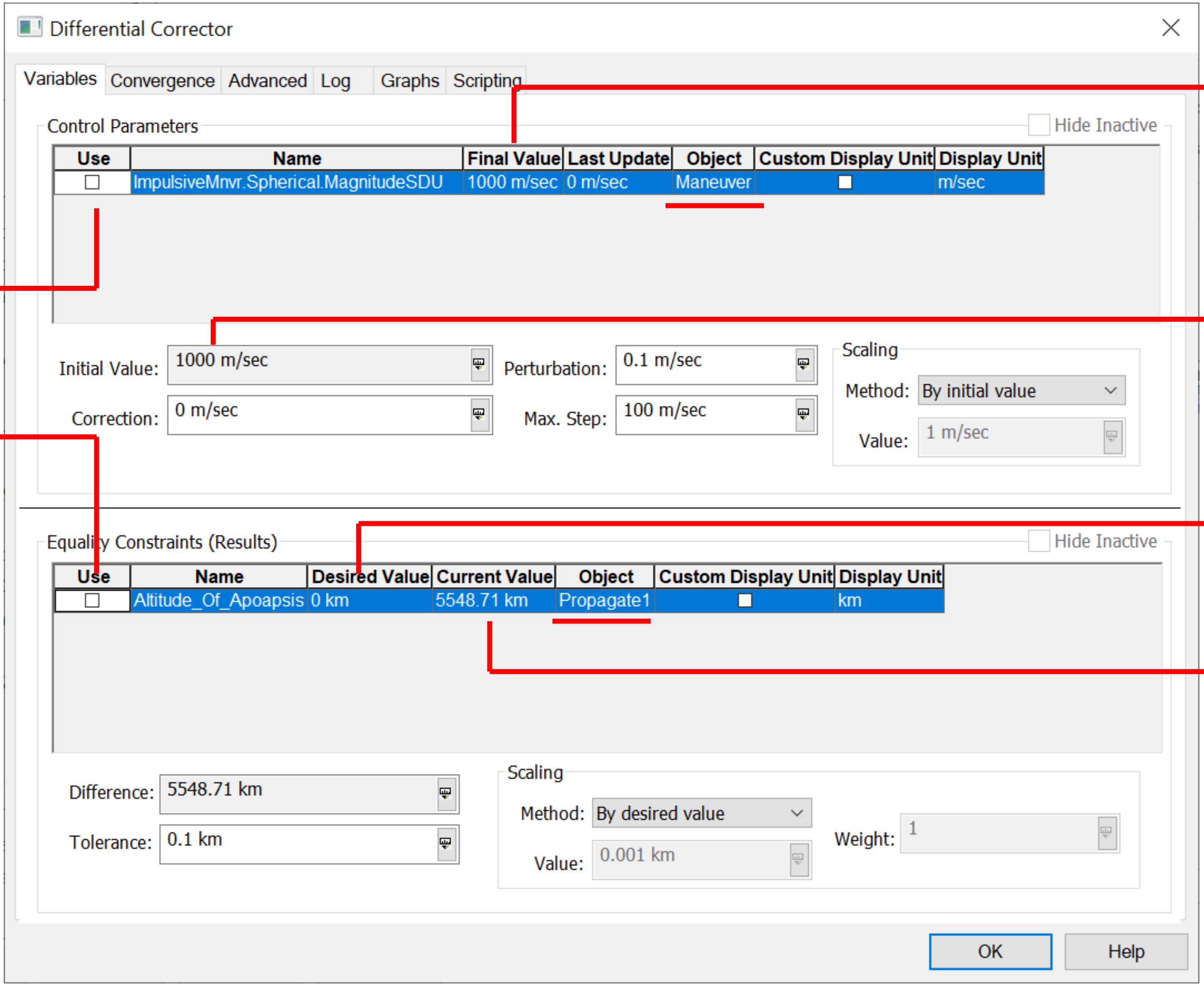


Astrogator

Target Sequences – Profiles – Differential Corrector

Differential corrector config:

- List of **independent variables**
- List of **dependent variables**



The final calculated value of the dependent variable

Initial value

The value we want to achieve

The current value with the final value of the independent variable

Astrogator

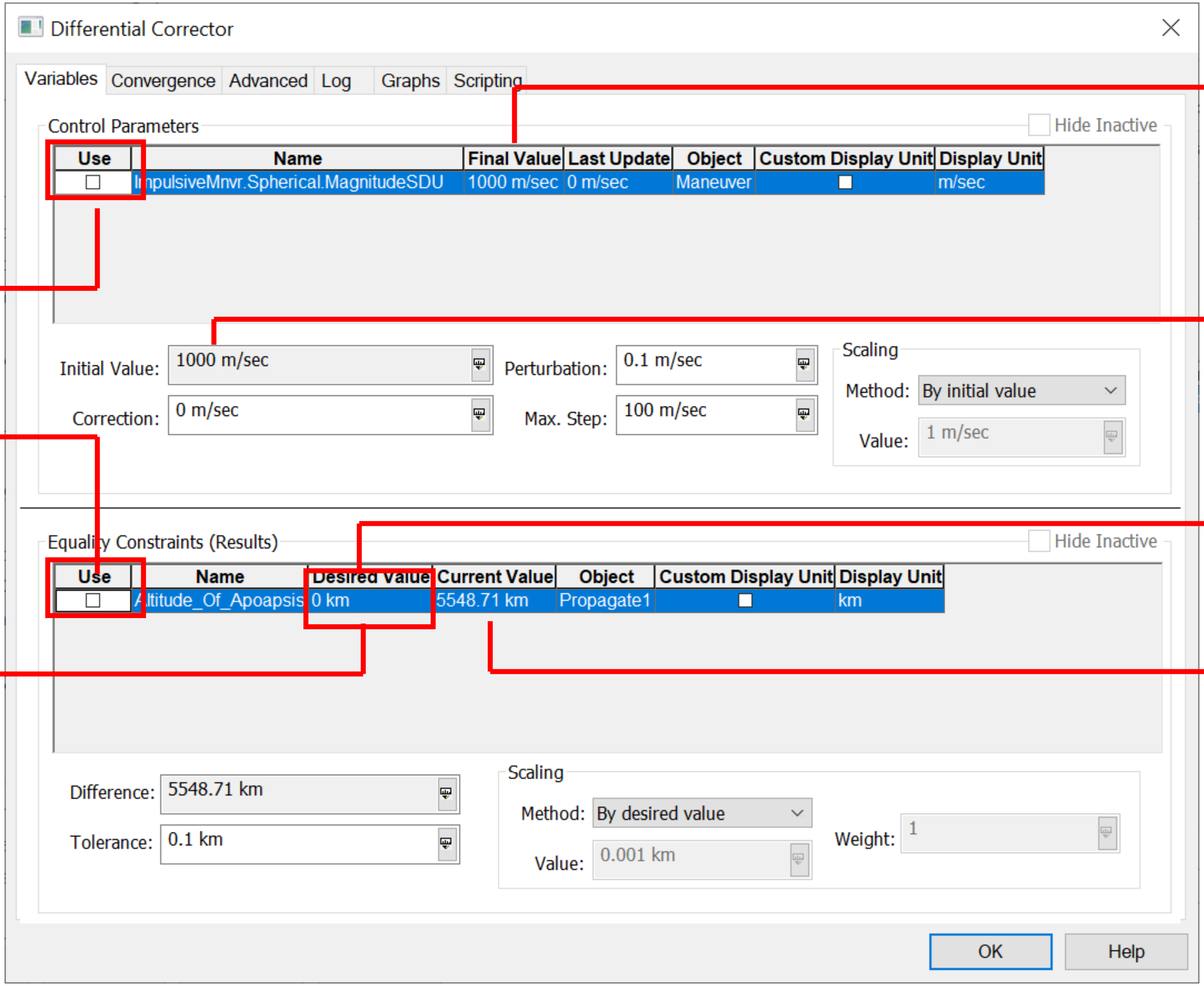
Target Sequences – Profiles – Differential Corrector

Differential corrector config:

- List of **independent variables**
- List of **dependent variables**

USE!

Set to 7500 km and run the sequence…



The final calculated value of the dependent variable

Initial value

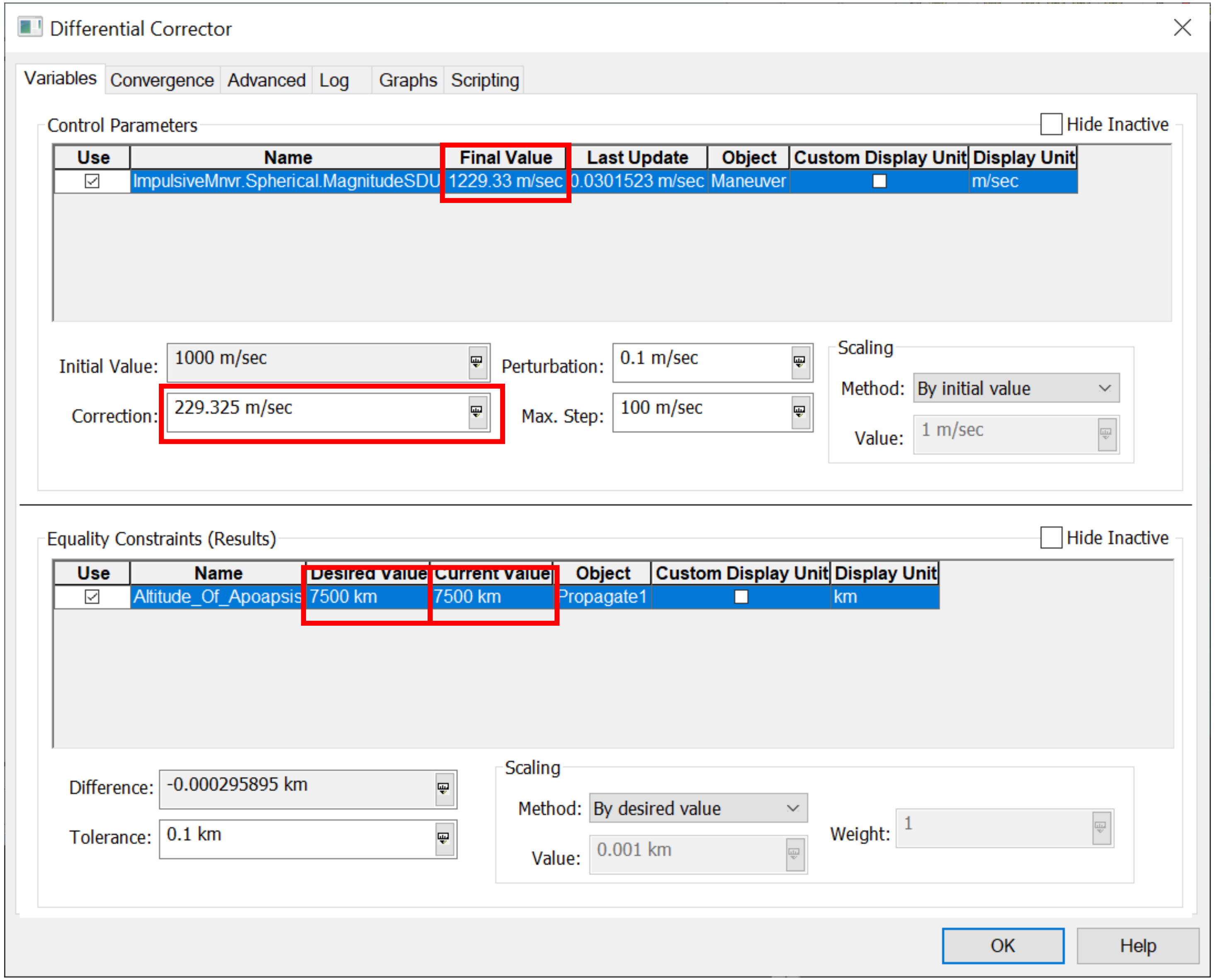
The value we want to achieve

The current value with the final value of the independent variable

Astrogator

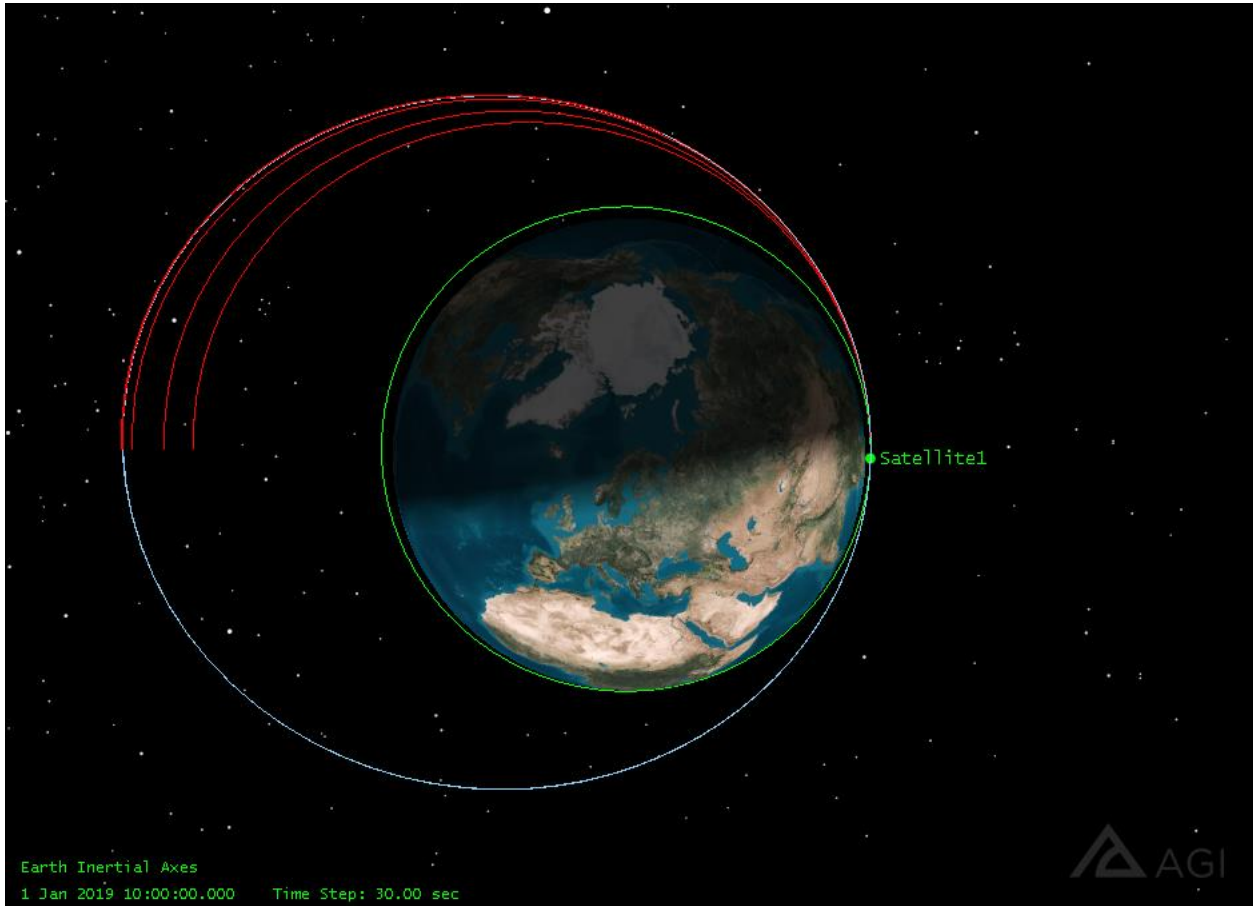
Target Sequences – Profiles – Differential Corrector

After running the corrector…



Astrogator

Target Sequences – Profiles – Differential Corrector

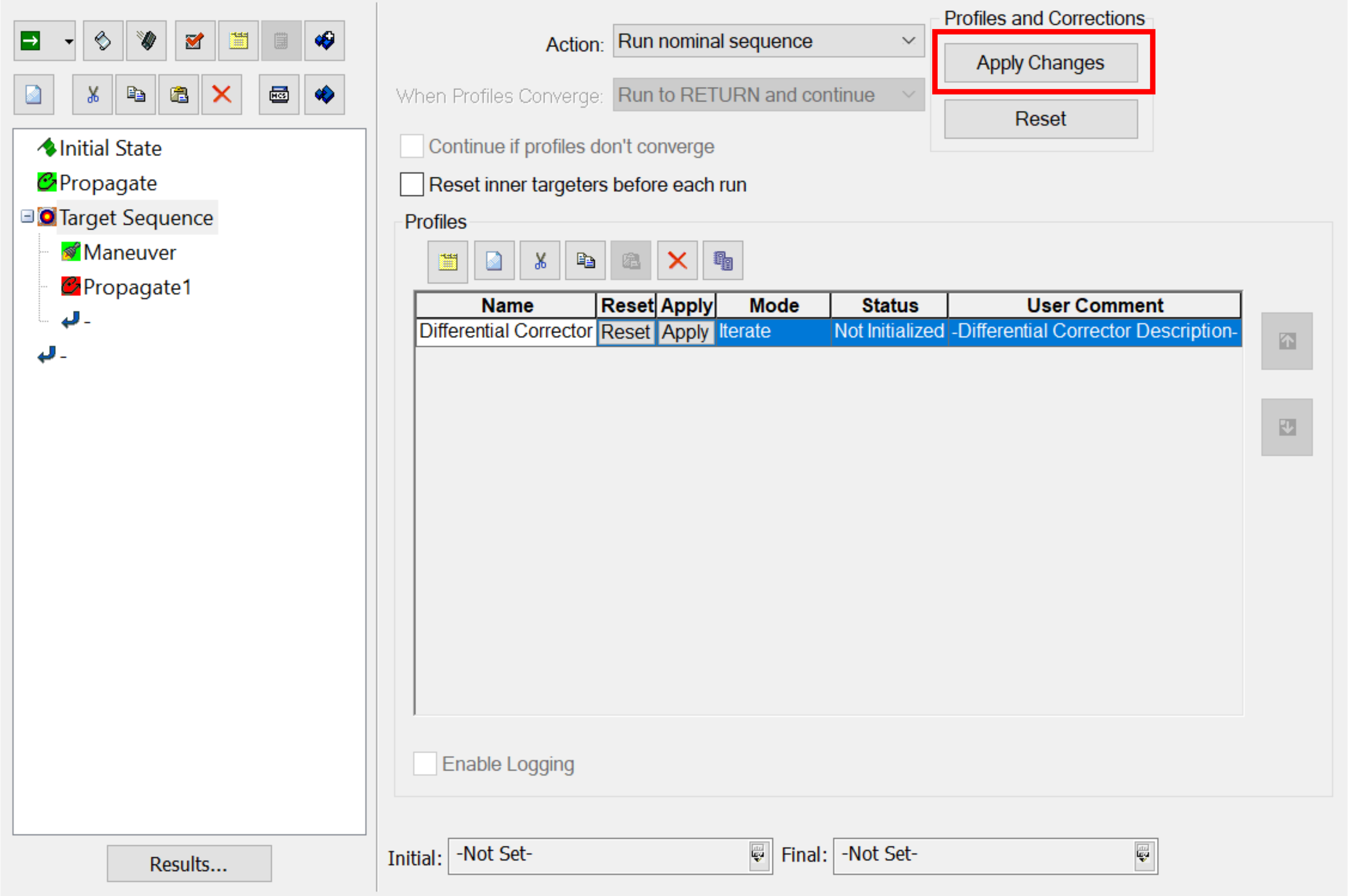


Astrogator

Target Sequences – Profiles – Differential Corrector

Happy with the result? **Apply**!

- Otherwise, reset and restart
- Set action to **Run nominal sequence** and proceed with the MCS…



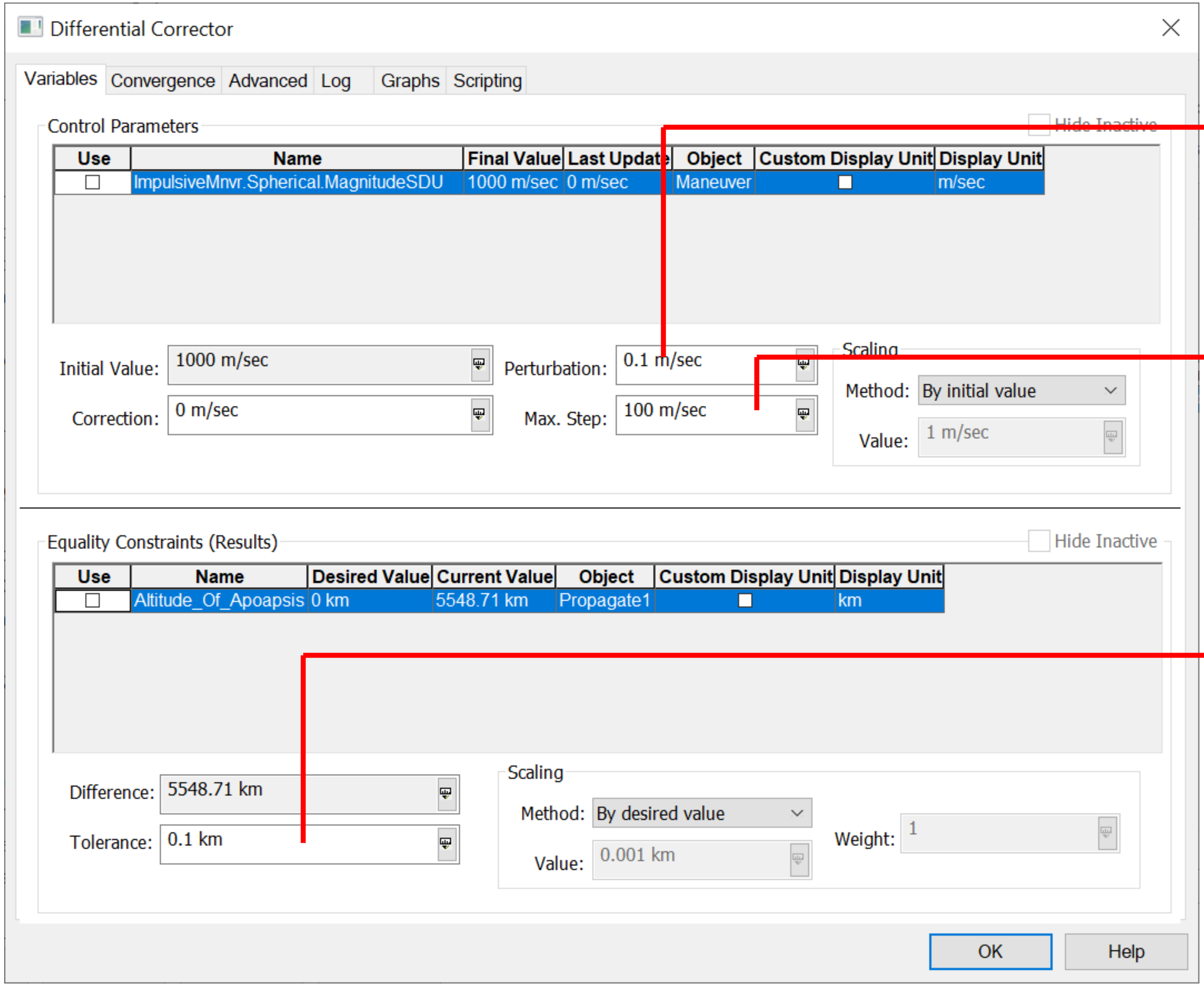
The target sequence will remember the last found value.

Unless a reset is applied, the target will converge immediately

Astrogator

Target Sequences – Profiles – Differential Corrector - Details

Some details of the Differential Corrector configuration



Value to be used in calculating numerical derivatives

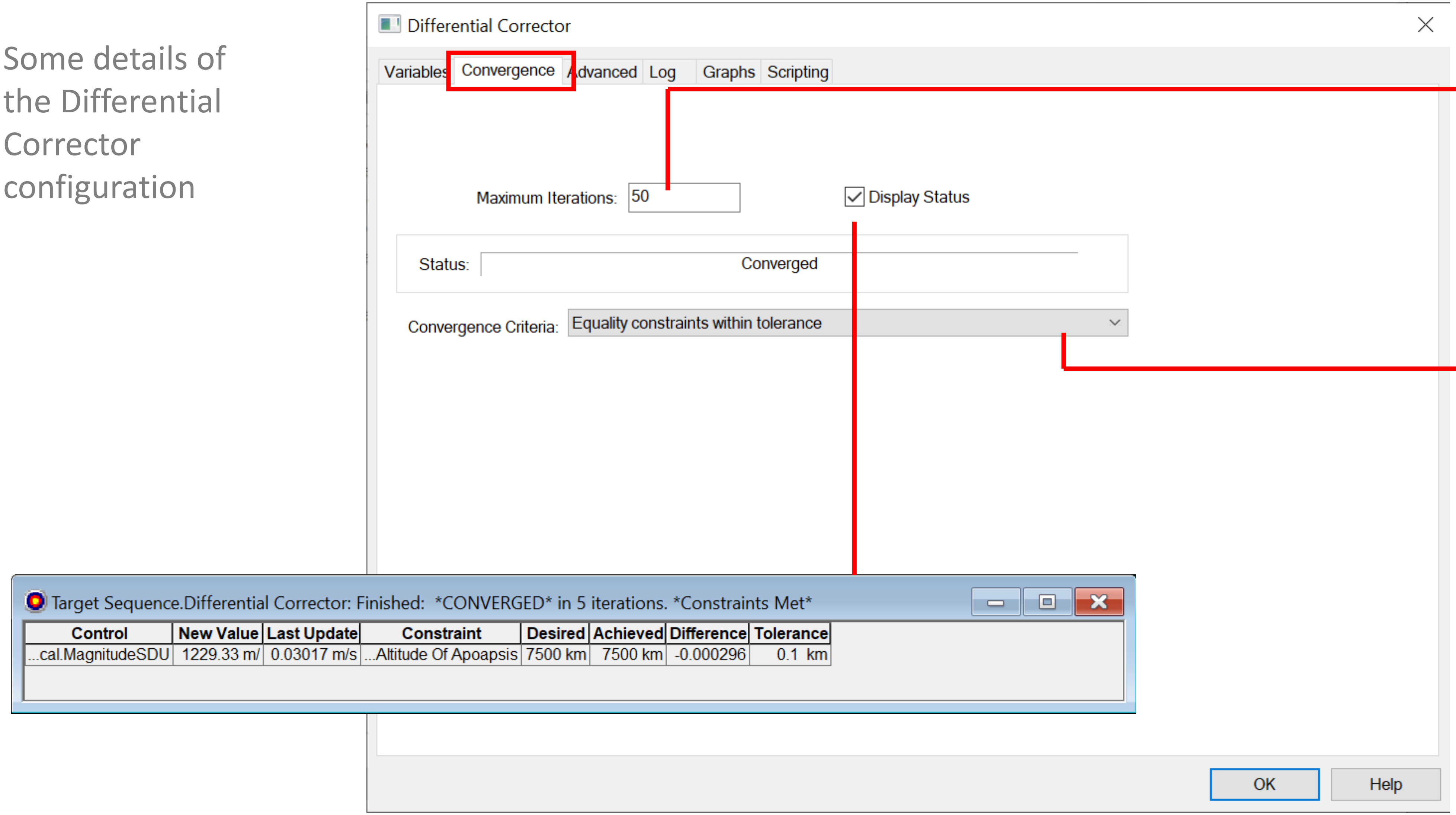
Maximum increment to make to the value of the parameter in any one step

The smallest update to the parameter to be made before the targeter stops

Astrogator

Target Sequences – Profiles – Differential Corrector - Details

Some details of the Differential Corrector configuration



After this, the search stops

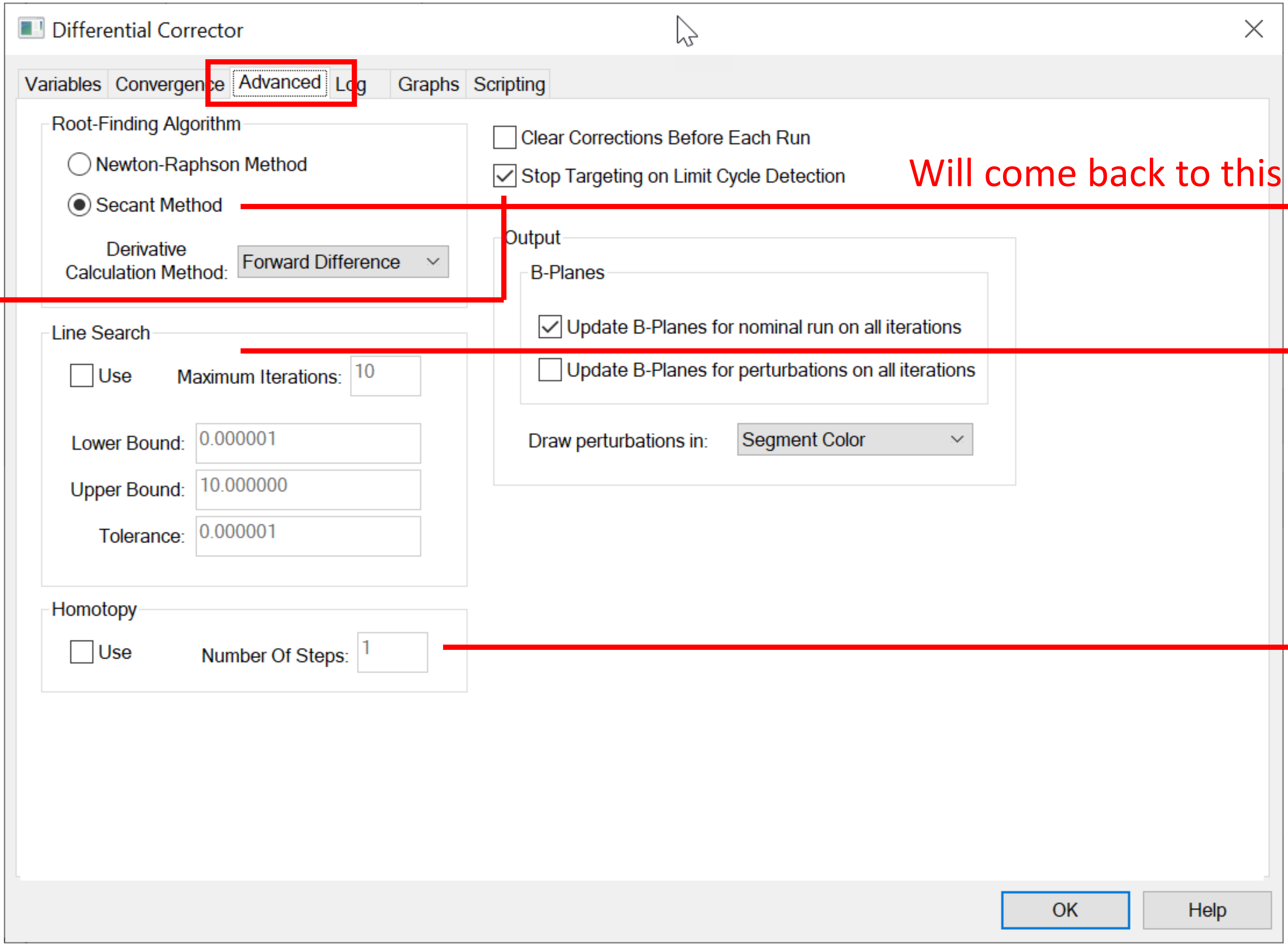
Result is within tolerance bounds
and
Last update within tolerance for convergence

Astrogator

Target Sequences – Profiles – Differential Corrector - Details

Some details of the Differential Corrector configuration

Limit cycles: independent variables are the same as two iteration earlier



Will come back to this later

Root-Finding algorithm – secant is known to converge faster

On each step, explore the line of the step (use when no converging)

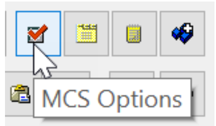
Attempt to reach the target in stages (recommended for non lineal searches)

Astrogator

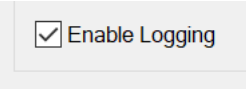
Target Sequences – Profiles – Differential Corrector - Details

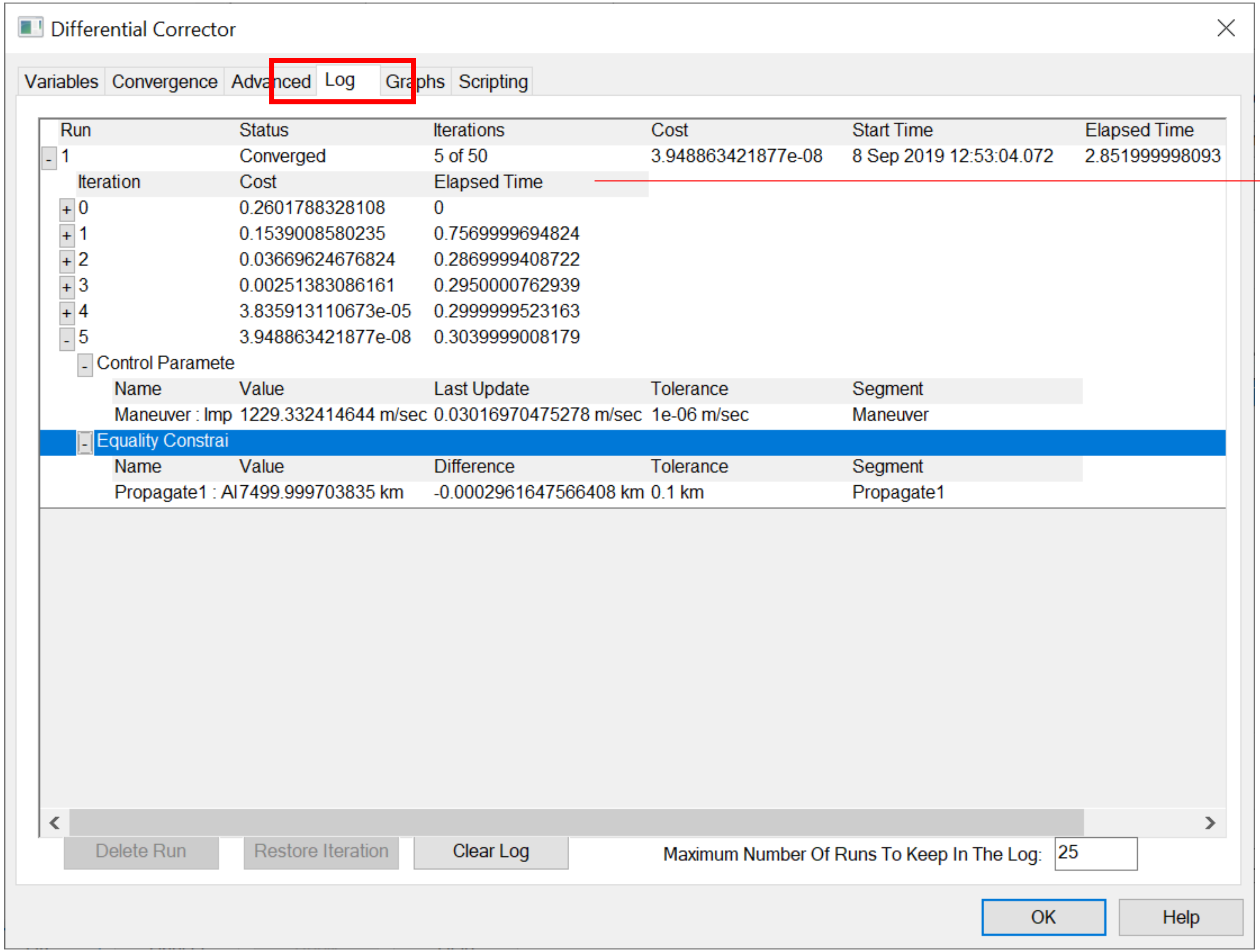
Some details of the Differential Corrector configuration

Enable logging 1) In the MCS options



2) In the target sequence config window



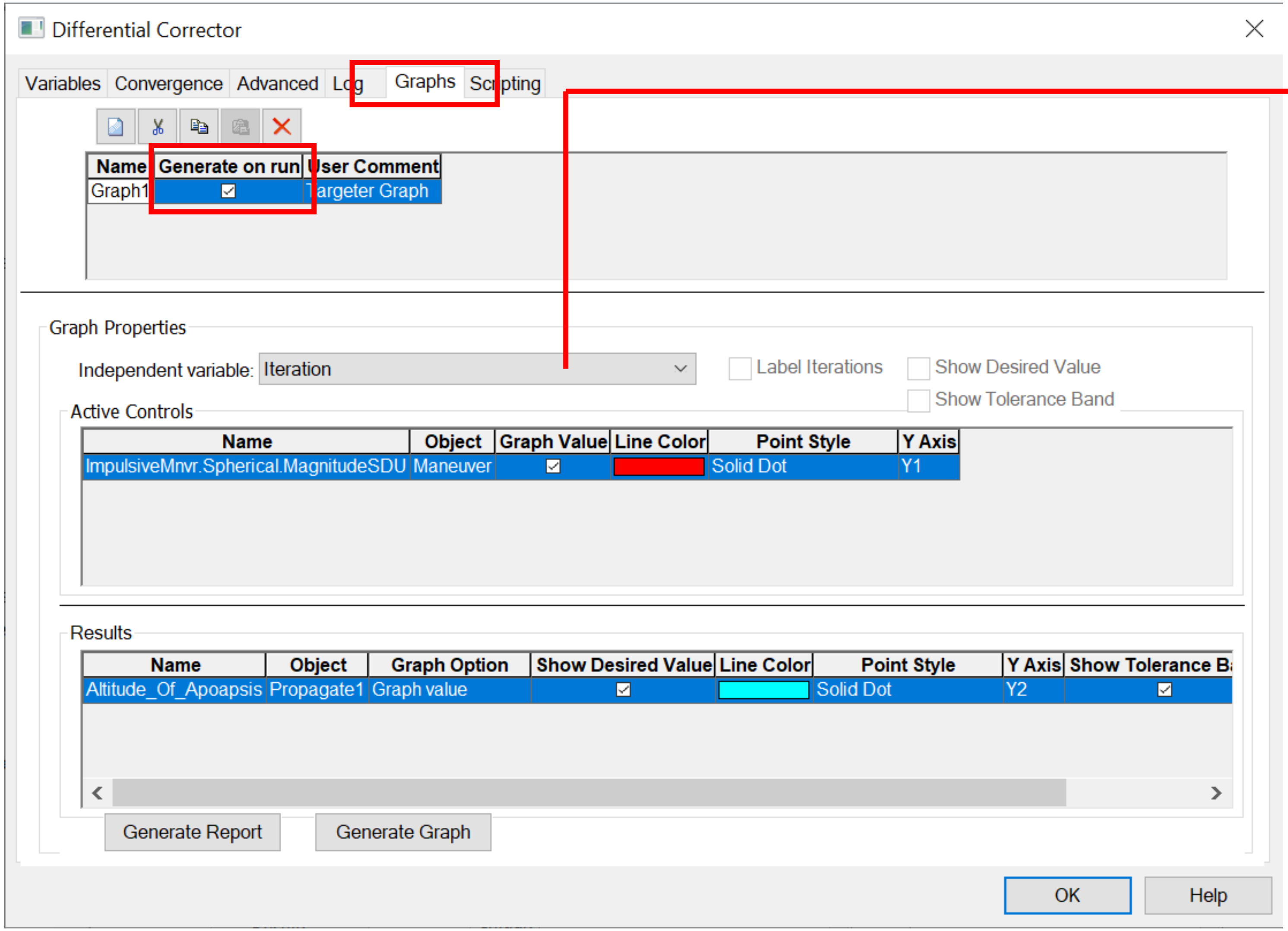


Cost is a simple function computed as the **square root of the sum of the squares of the scaled differences** between the achieved and desired values of the equality constraints.

Astrogator

Target Sequences – Profiles – Differential Corrector - Details

Some details of the Differential Corrector configuration



Plot the values vs

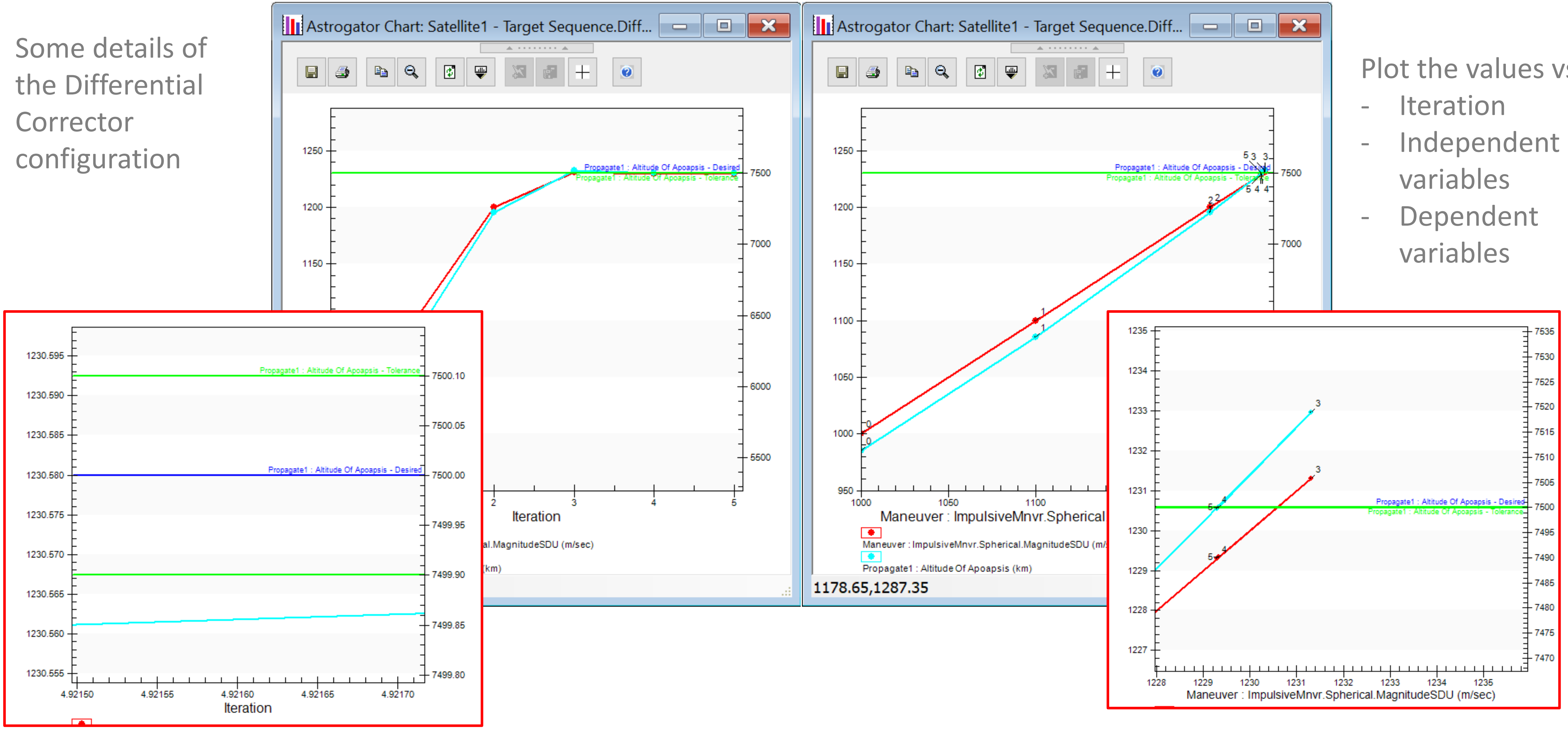
- Iteration
- Independent variables
- Dependent variables

In real time as astrogator calculates!

Astrogator

Target Sequences – Profiles – Differential Corrector - Details

Some details of the Differential Corrector configuration



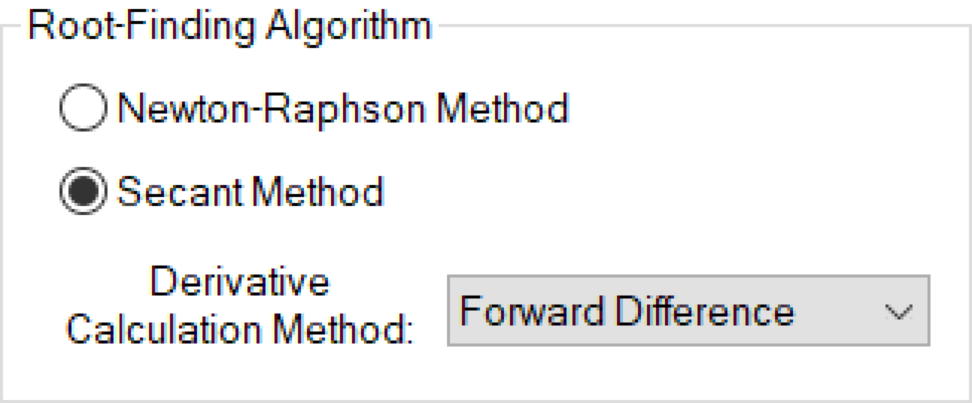
Plot the values vs

- Iteration
- Independent variables
- Dependent variables

Astrogator

Target Sequences – Profiles – Differential Corrector - Under the Hood

Root-Finding Algorithms

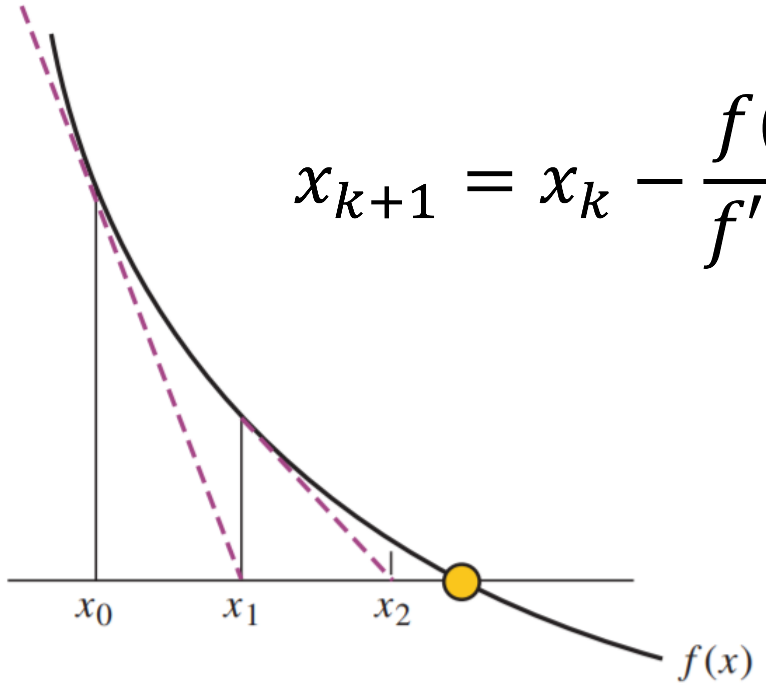


- A root of a function $f\colon \mathbb{R}^n \rightarrow \mathbb{R}^m$, is a vector of n **independent** variables $\boldsymbol{x}$ such that $\mathbf{y} = f(\boldsymbol{x}) = 0$, where $\mathbf{y}$ is a vector of m **dependent** variables.
- In most cases, $\boldsymbol{x}$ cannot be computed exactly, but approximated typically starting from some **initial guess** $\boldsymbol{x}_0$ and iterating to hopefully converge in the root
- On each cycle a step is computed for $\boldsymbol{x}$ and evaluated at least once

Newton-Raphson Method

STK Method 1)

- Uses the **first derivative** of f to determine the step to take in $\boldsymbol{x}$
- At each iteration, the $\boldsymbol{x}$ values for the next iteration are chosen such that the root would be reached if $f(\boldsymbol{x})$ were linear



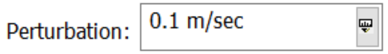
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Single-variable

Newton-Raphson Method

STK Method 1)

- Uses the **first derivative** of f to determine the step to take in $\boldsymbol{x}$
- Since the first derivative cannot be computed analytically, it can be obtained numerically by using a **perturbation** δx



$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k + \delta x) - f(x_k)} \delta x$$

Single-variable

Newton-Raphson Method

STK Method 1)

- In **multivariable problems**, the Newton-Raphson method is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \mathbf{J}_n^{-1}(\boldsymbol{f}(\boldsymbol{x}_k))$$

(Pseudo-inverse if **J** not squared)

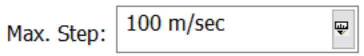
- To compute **J** numerically $\boldsymbol{x}$ variables are **perturbed** by δx. Every i column of **J** is given by

Column vector with δx_i in i[th] column, zero on the rest

$$\mathbf{J}_i = \frac{1}{\delta x_i}\left(\boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta x}_i) - \boldsymbol{f}(\boldsymbol{x})\right)$$

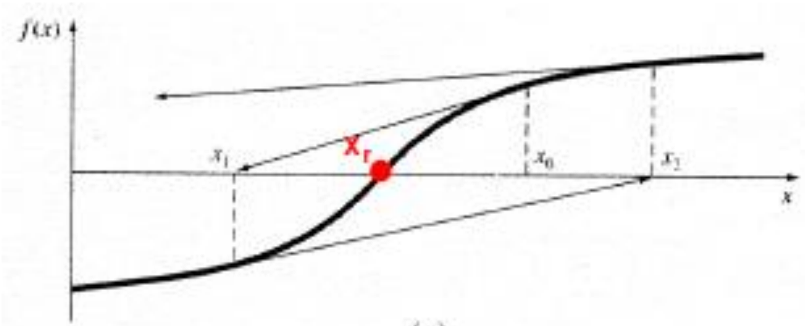
meaning that n evaluations of $\boldsymbol{f}$ are required per iteration

- A **maximum step** size Δx_i can be configured to avoid overshooting



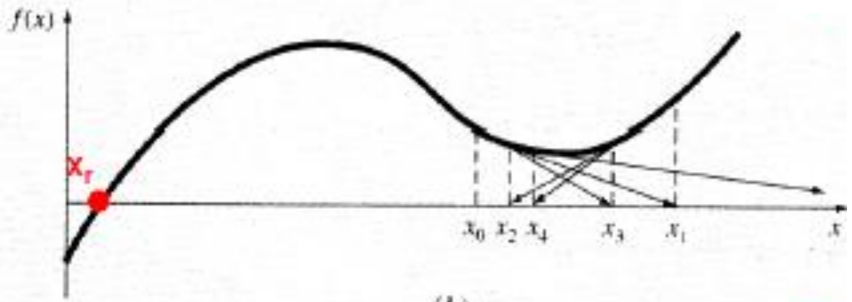
Newton-Raphson Method

STK Method 1) - Problems



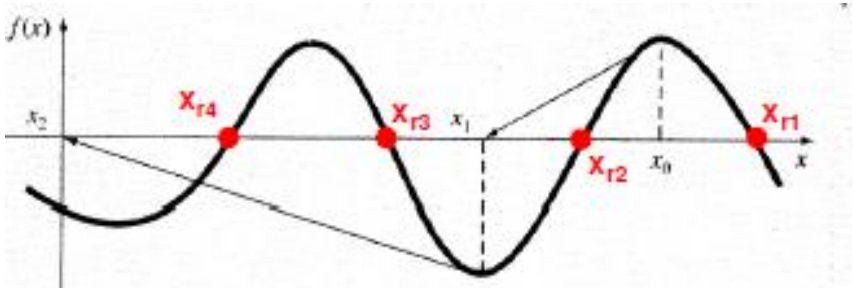
(a)

Case-a: Case where an inflection point ($f''(x)=0$) occurs in the vicinity of a root. Iterations begins at x_0 progressively diverge from the root.



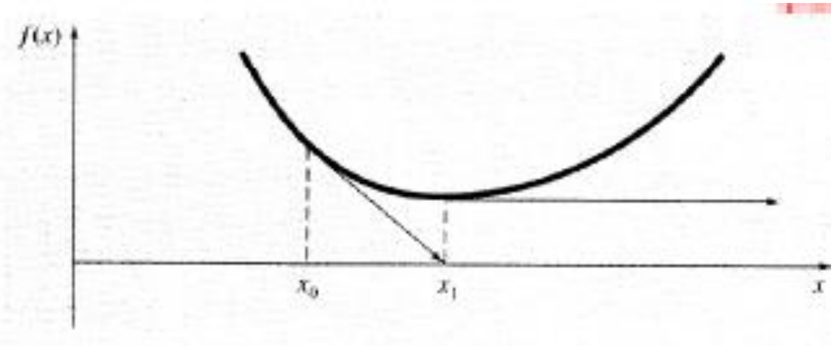
(b)

Case-b: Case shows the tendency of the Newton's method to oscillate around a local maximum or minimum.



(c)

Case-c: Case shows how an initial guess that is close to one root can jump to a location several roots away. This is due to near zero slope.



(d)

Case-d: Case shows an encounter of zero slope ($f'(x)=0$). In this case the solution shoots off horizontally and never hits the x-axis. (diverge- a disaster)

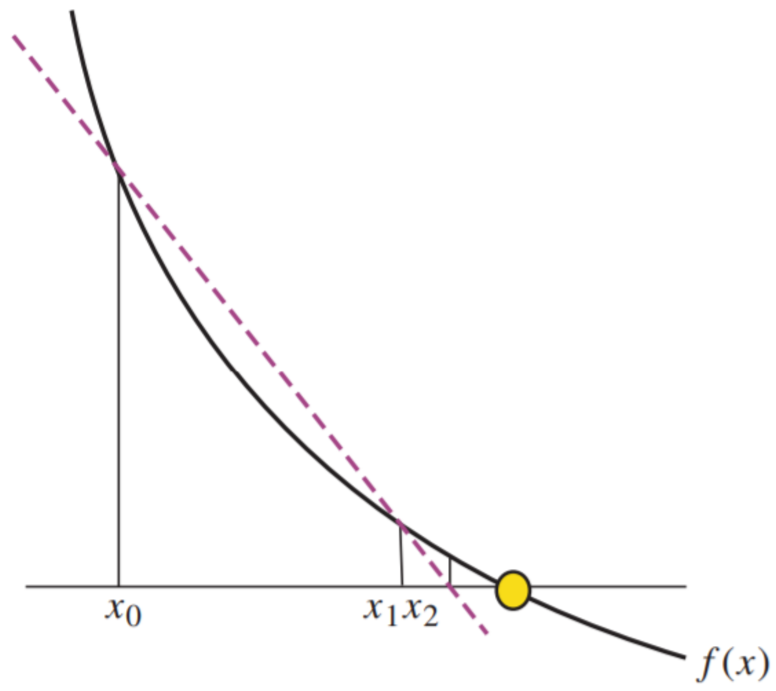
Secant Method (Broyden's Method)

STK Method 2)

- Uses **successive values** of x (wider range than with perturbations)

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

Single-variable



Secant Method (Broyden’s Method)

STK Method 2)

- **Broyden’s** method is a generalization of the secant method for multi-variable problems.
- Instead of evaluating **J** on every iteration, it uses an **estimate**

$$\mathbf{J}_{k} = \mathbf{J}_{k-1} + \frac{\boldsymbol{f}(\boldsymbol{x}_{k}) - \boldsymbol{f}(\boldsymbol{x}_{k-1}) - \mathbf{J}_{k-1}(\boldsymbol{x}_{k} - \boldsymbol{x}_{k-1})}{\|\boldsymbol{x}_{k} - \boldsymbol{x}_{k-1}\|^{2}}$$

meaning that only 1 evaluation of $\boldsymbol{f}$ is required per iteration, but more iterations might be required to converge

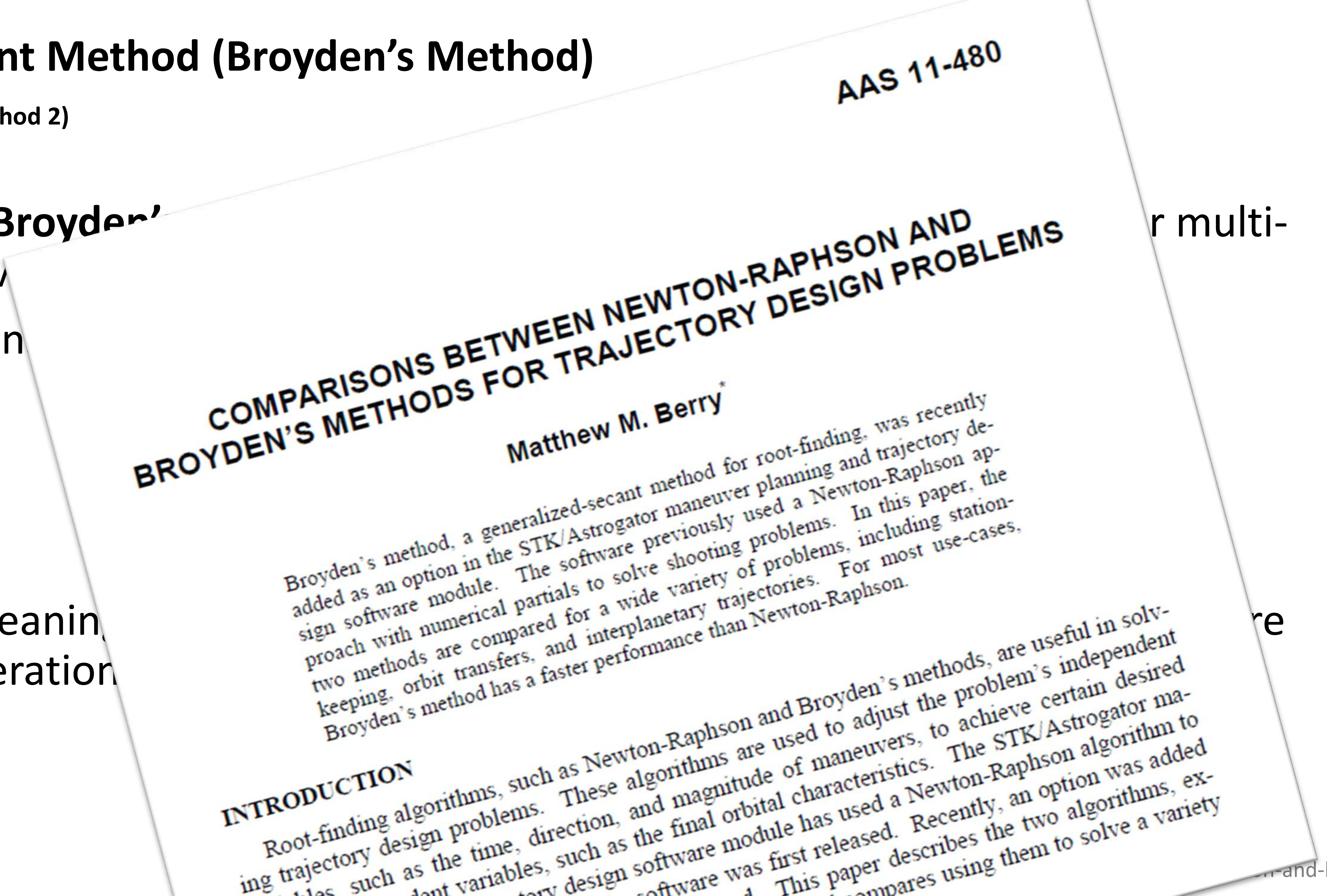
https://www.agi.com/resources/white-papers/comparisons-between-newton-raphson-and-broyden-s-m

Secant Method (Broyden’s Method)

STK Method 2)

- **Broyden’** r multi-
  v
- In

meanin re
iteration



AAS 11-480

COMPARISONS BETWEEN NEWTON-RAPHSON AND BROYDEN’S METHODS FOR TRAJECTORY DESIGN PROBLEMS

Matthew M. Berry*

Broyden’s method, a generalized-secant method for root-finding, was recently added as an option in the STK/Astrogator maneuver planning and trajectory design software module. The software previously used a Newton-Raphson approach with numerical partials to solve shooting problems. In this paper, the two methods are compared for a wide variety of problems, including station-keeping, orbit transfers, and interplanetary trajectories. For most use-cases, Broyden’s method has a faster performance than Newton-Raphson.

INTRODUCTION

Root-finding algorithms, such as Newton-Raphson and Broyden’s methods, are useful in solving trajectory design problems. These algorithms are used to adjust the problem’s independent variables, such as the time, direction, and magnitude of maneuvers, to achieve certain desired dependent variables, such as the final orbital characteristics. The STK/Astrogator maneuver design software module has used a Newton-Raphson algorithm to ... software was first released. Recently, an option was added ... This paper describes the two algorithms, ex- ... compares using them to solve a variety

Methods Comparison

In Depth Analysis

- The paper shows that in several test cases Broyden's method uses fewer evaluations in multiple different cases

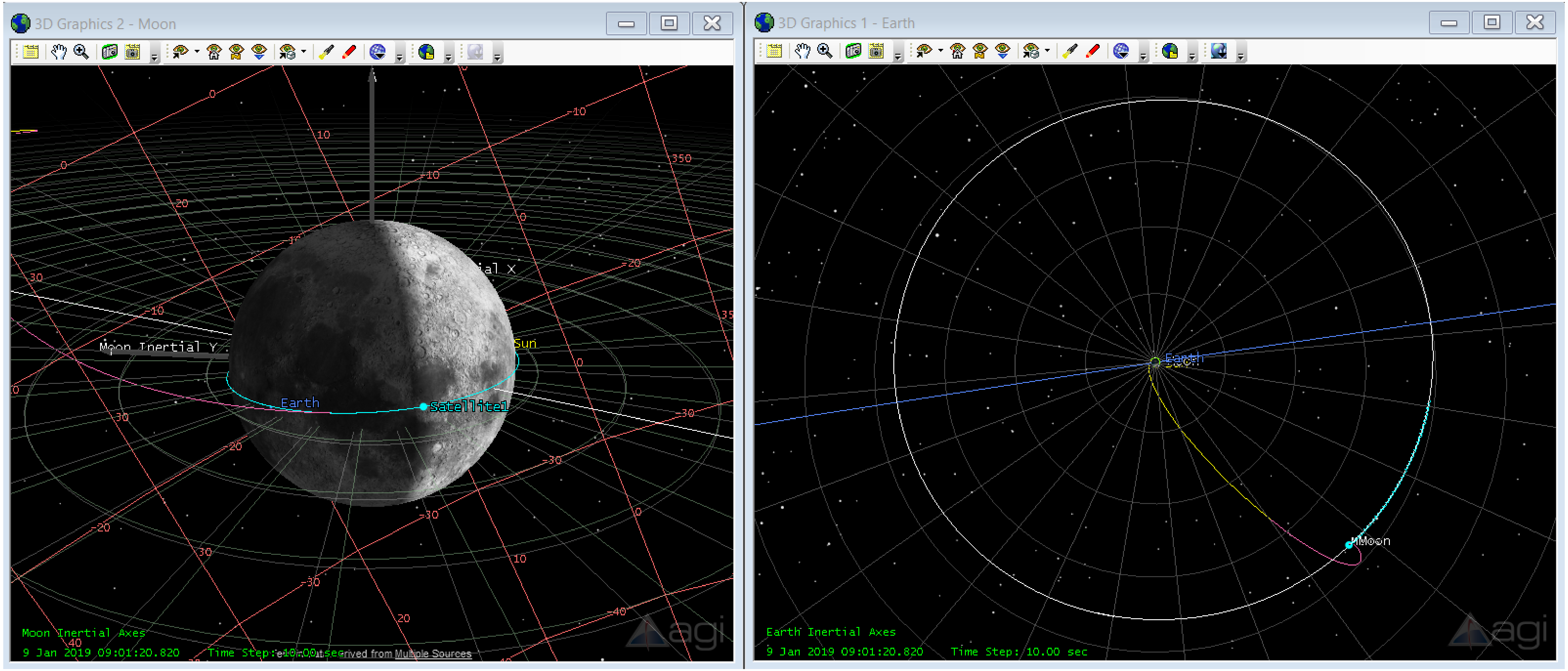
Table 3: Iterations and evaluations needed in Earth to Moon example

Profile	Newton-Raphson method		Broyden's method	
	Iterations	Evaluations	Iterations	Evaluations
RA Dec	5	16	7	10
B-plane	3	13	5	9
Altitude and inclination	3	13	2	6
Total		42		25

https://www.agi.com/resources/white-papers/comparisons-between-newton-raphson-and-broyden-s-m

Astrogator

Lunar Mission Tutorial (STK Moon Mission with B-Plane Targeting Tutorial)



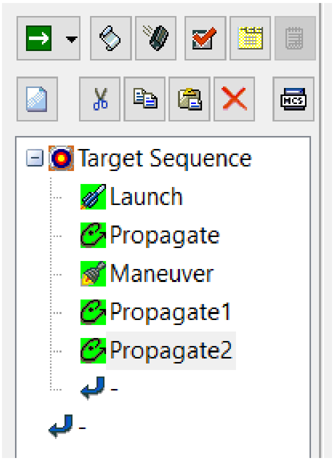
Astrogator

Lunar Mission Tutorial (STK Moon Mission with B-Plane Targeting Tutorial)

- Create scenario **1st Jan 2018 + 30 days** (link to tutorial)
- Scenario properties
  - Basic, time: **Step size 3 min**
  - 2D graphics, global: show **planet orbit**, not show **ground tracks** and **markers**
- Add planet, default planet, repeat 3 times
  - Set them to **Sun**, **Earth** and **Moon**
- 3D window
  - Grids, view **ECI coordinates**
  - Advanced, **view distance** 1e+10 km
- View, duplicate 3D graphics window, center on moon

Astrogator

Lunar Mission Tutorial – Setting the Context

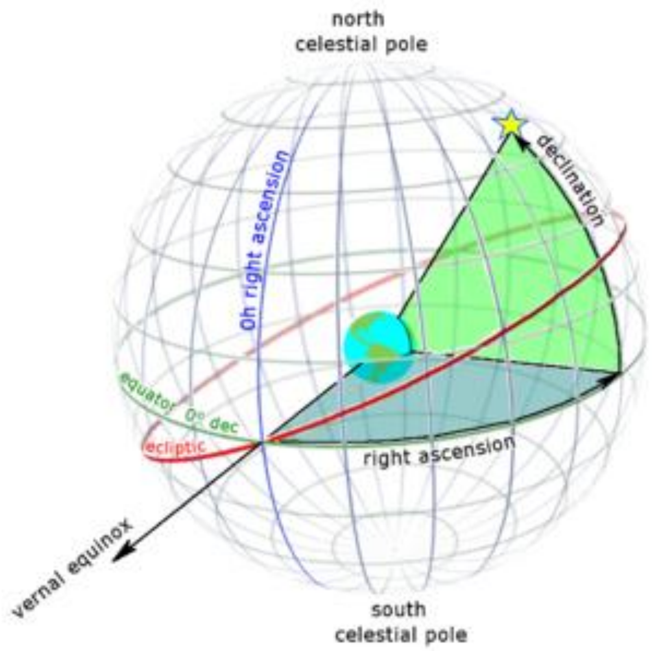


- Add default **satellite** (Lunar Probe), set Astrogator
- In Astrogator
  - Delete all segments
  - Add **target sequence**
    - **Launch** (set to **initial analysis time**)
    - Propagate (**Coast**): **90 min** trip
    - Maneuver (**TransLunarInjection**): set to **Thrust vector**, **VNC Earth**, **X=3120 m/sec**
    - Propagate (**ToSwingBy**): set propagator to **Cis-Lunar**, **R Mag** stopping condition, **trip=300000 km**
    - Propagate (**ToPeriselene**): set propagator to **Cis-Lunar**, Stop condition 1 duration to **10 days**, Stop condition 2 **altitude trip 0km**, moon central, Stop condition 3 **periapsis**, moon central
- **Run** the sequence to see how it looks like

Astrogator

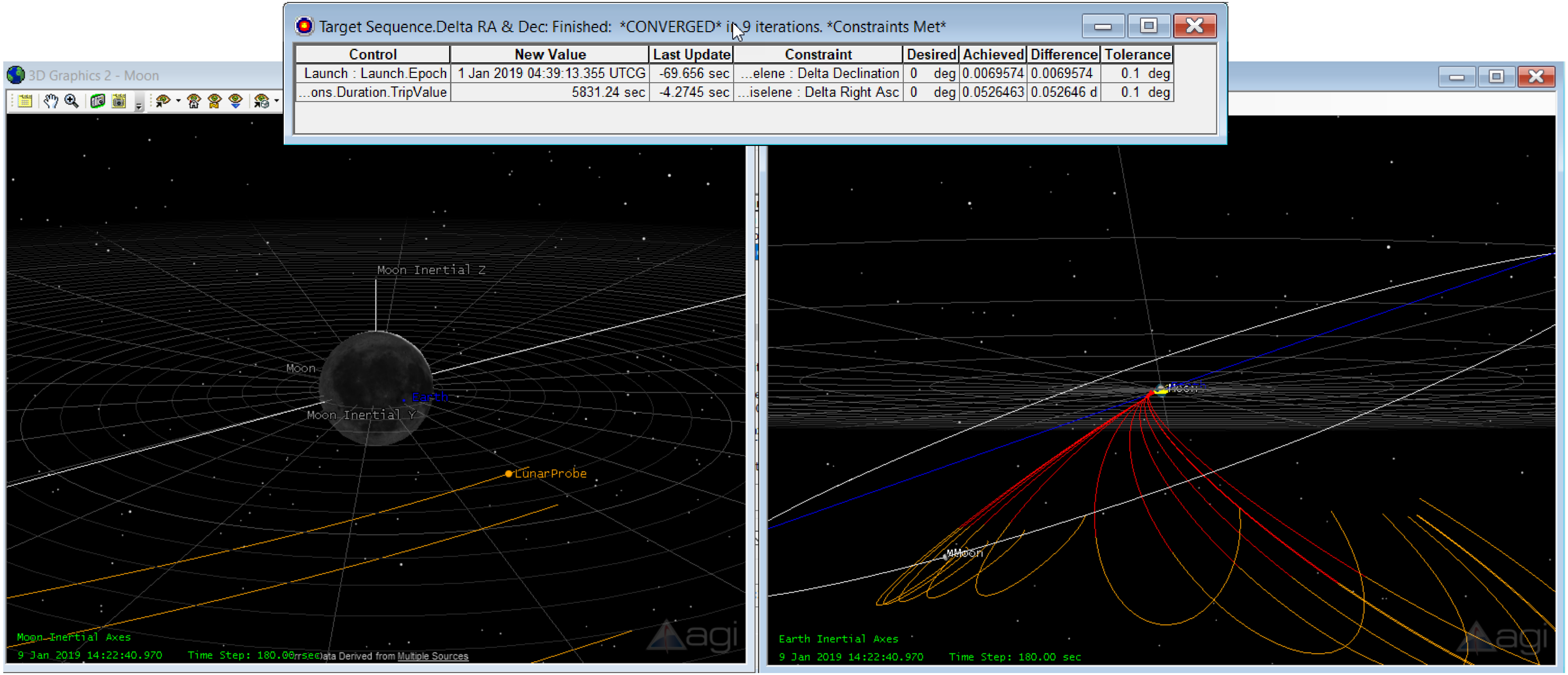
Lunar Mission Tutorial – 1) Launch and Coast Times

- Set **independent** variables
    - Launch: **Epoch (set time)**
    - Propagate (Coast): **Trip time**
- Set **dependent** variables (results)
    - Propagate (ToPeriselene): multibody, **delta right ascension** and **delta declination**, verify central body is the Moon
- Name this profile as **Delta RA & Dec** and set
    - Enable **both independent** variables
        - Launch: perturbation=1 min, Max step=1 hr
        - StopCond.Durantion.Trip: perturbation=1 min, Max step=5 min
    - Enable both **dependent** variables (results)
    - Set iterations to 100
    - Target sequence: Run Active Profiles



Astrogator

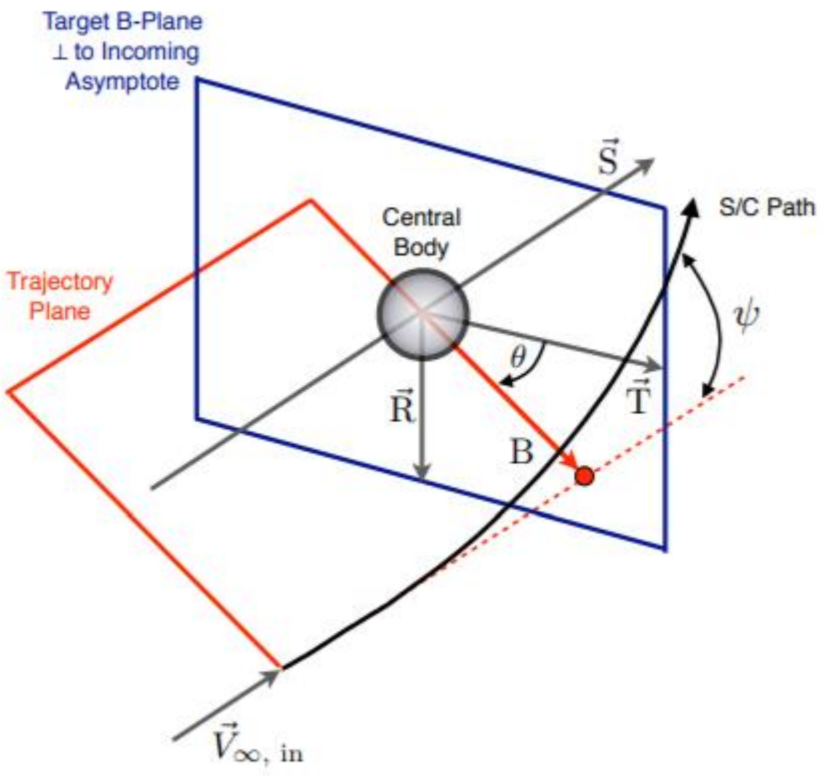
Lunar Mission Tutorial – 1) Launch and Coast Times



Astrogator

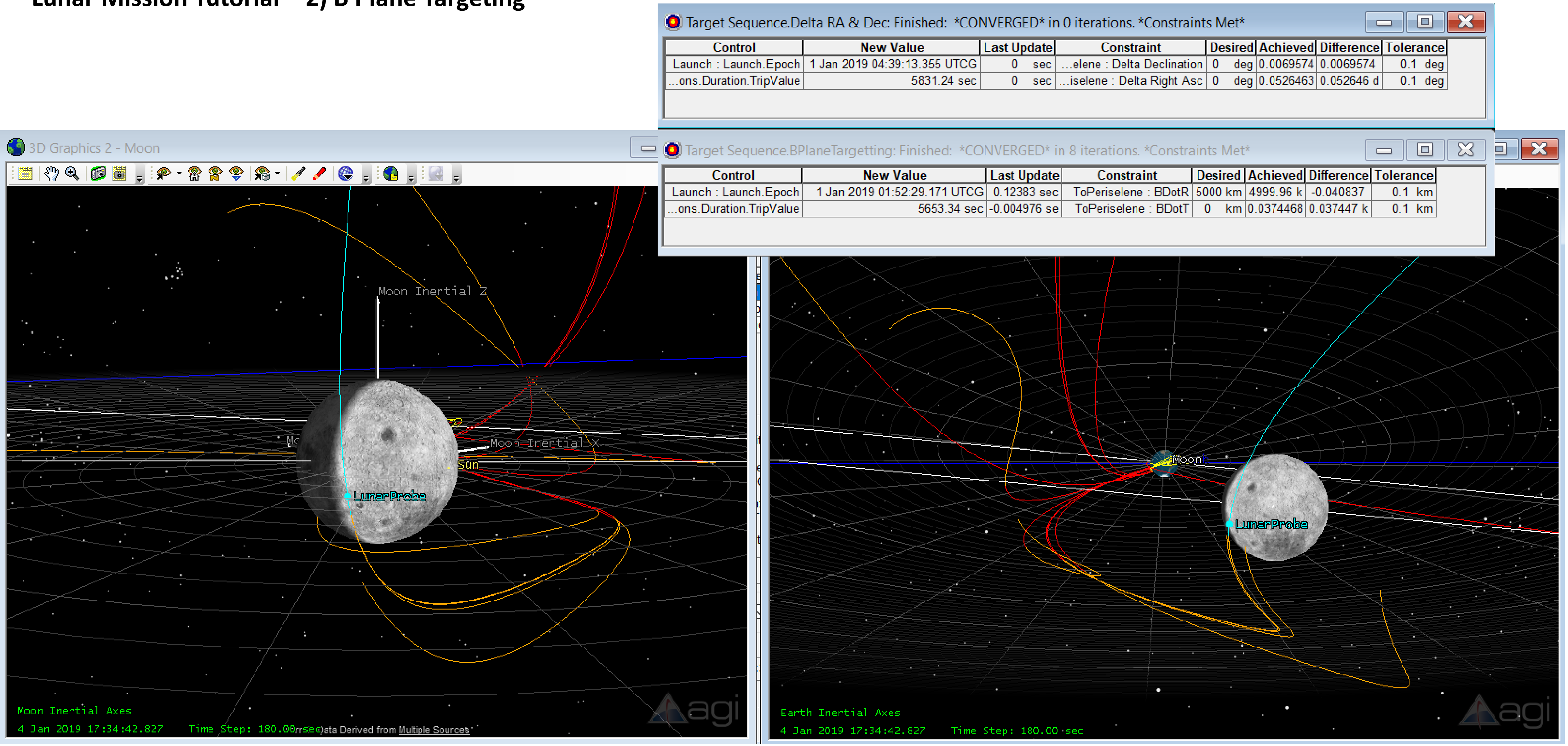
Lunar Mission Tutorial – 2) B Plane Targeting

- Duplicate differential corrector
- Set new **dependent** variables (results)
  - Propagate (ToPeriselene): multibody, **BdotT** and **BdotR**, verify central body is the Moon
- Name this profile as **B-PlaneTargeting** and go to properties
  - Enable **BdotT** and **BdotR** as **dependent** variables
  - Set BdotR desired value to 5000 km
- Add new Propagator (Prop3Days) after target sequence
- Target sequence: Run Active Profiles



Astrogator

Lunar Mission Tutorial – 2) B Plane Targeting



Target Sequence.Delta RA & Dec: Finished: *CONVERGED* in 0 iterations. *Constraints Met*

Control	New Value	Last Update	Constraint	Desired	Achieved	Difference	Tolerance
Launch : Launch.Epoch	1 Jan 2019 04:39:13.355 UTCG	0 sec	...elene : Delta Declination	0 deg	0.0069574	0.0069574	0.1 deg
...ons.Duration.TripValue	5831.24 sec	0 sec	...iselene : Delta Right Asc	0 deg	0.0526463	0.052646 d	0.1 deg

Target Sequence.BPlaneTargetting: Finished: *CONVERGED* in 8 iterations. *Constraints Met*

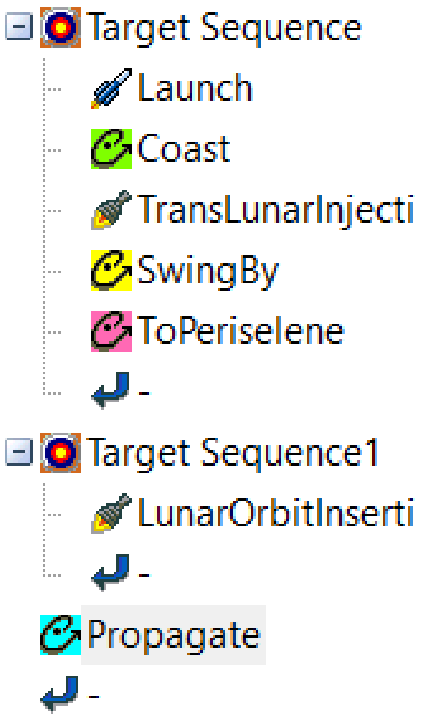
Control	New Value	Last Update	Constraint	Desired	Achieved	Difference	Tolerance
Launch : Launch.Epoch	1 Jan 2019 01:52:29.171 UTCG	0.12383 sec	ToPeriselene : BDotR	5000 km	4999.96 k	-0.040837	0.1 km
...ons.Duration.TripValue	5653.34 sec	-0.004976 se	ToPeriselene : BDotT	0 km	0.0374468	0.037447 k	0.1 km

Astrogator

Lunar Mission Tutorial – 3) Lunar Orbit Insertion

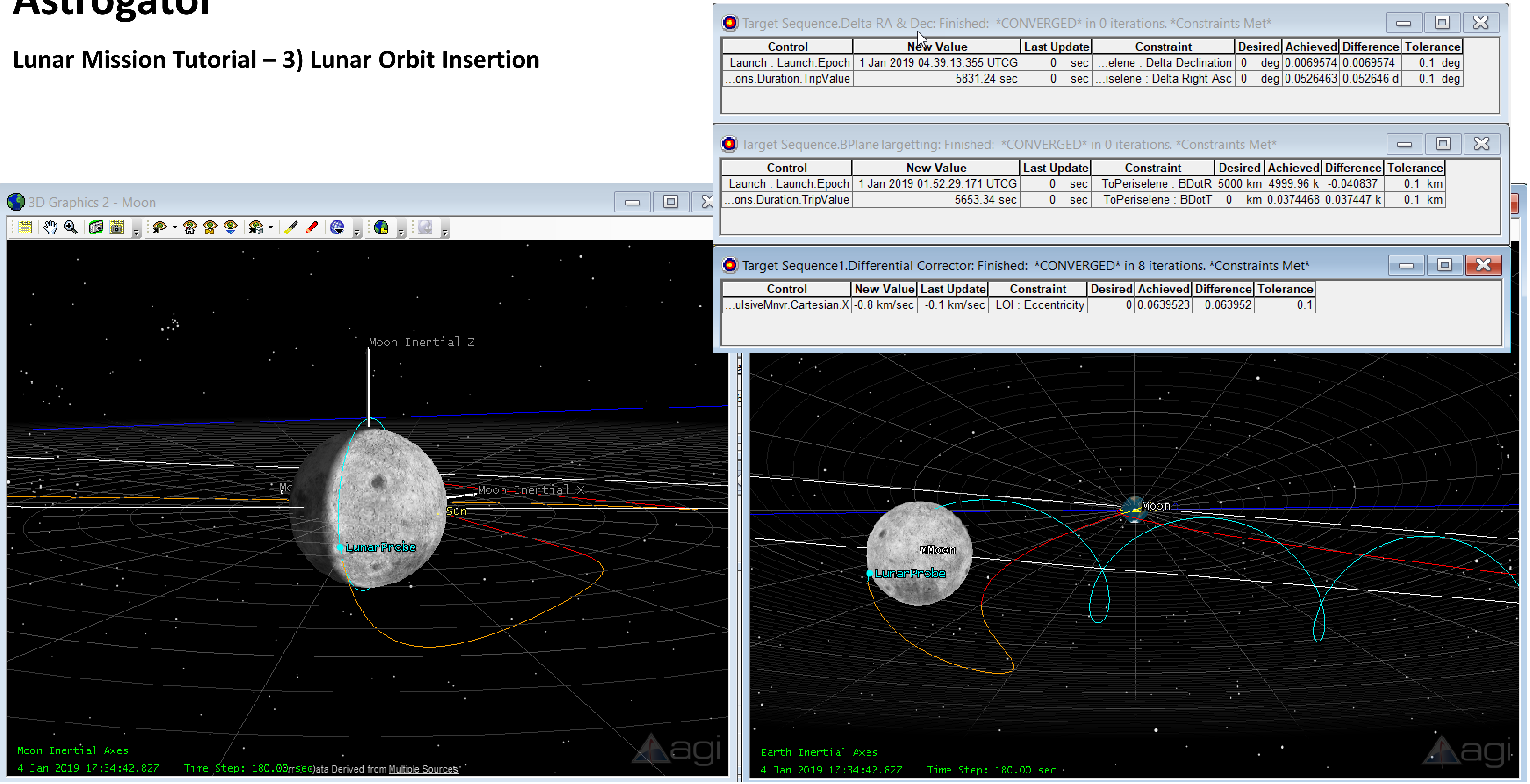
V The X axis is along the velocity vector $\vec{V}$
N The Y axis is along the orbit normal ()
C The Z axis completes the orthogonal triad ()

- Add new target sequence
  - Add **maneuver LunarOrbitInsertion** (LOI), set **Thrust vector:** **VNC (Moon)**
- Set new **independent** variables
  - **X Velocity**: leave at 0 to let Astrogator calculate the delta-V
- Set new **dependent** variables (results)
  - Maneuver: Keplerian elements, **eccentricity** (central body: Moon)
- Name this profile as **LOI** and go to properties
  - Enable **X Velocity** as **independent** and **eccentricity** as **dependent** variable
- Target sequence: Run Active Profiles



Astrogator

Lunar Mission Tutorial – 3) Lunar Orbit Insertion



Astrogator

Astrogator Guild - https://astrogatorsguild.com/

SEE Home

The Astrogator's Guild

LUNAR MISSIONS VIDEO BLOG

by Astrogator John | May 16, 2019 | General Comments

Mike Loucks and I were at agi.com last week in their new studio and we recorded this video blog on YouTube with Josh Poley. We're talking about some Lunar missions we worked on:

Ep6 – Lunar Missions

Recent Posts

- Lunar Missions Video Blog
- Great Article about LADEE that somehow didn't get posted here 3 years ago
- "Back of the Envelope" Astrogation: Re-creating the New Horizons Trajectory
- Candidate Object on Asteroid Zoo could be "Snoopy" from Apollo 10
- Could the Meteorite crater in Nicaragua be related to Asteroid 2014 RC?

Meta

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

Astrogator Training Scenarios

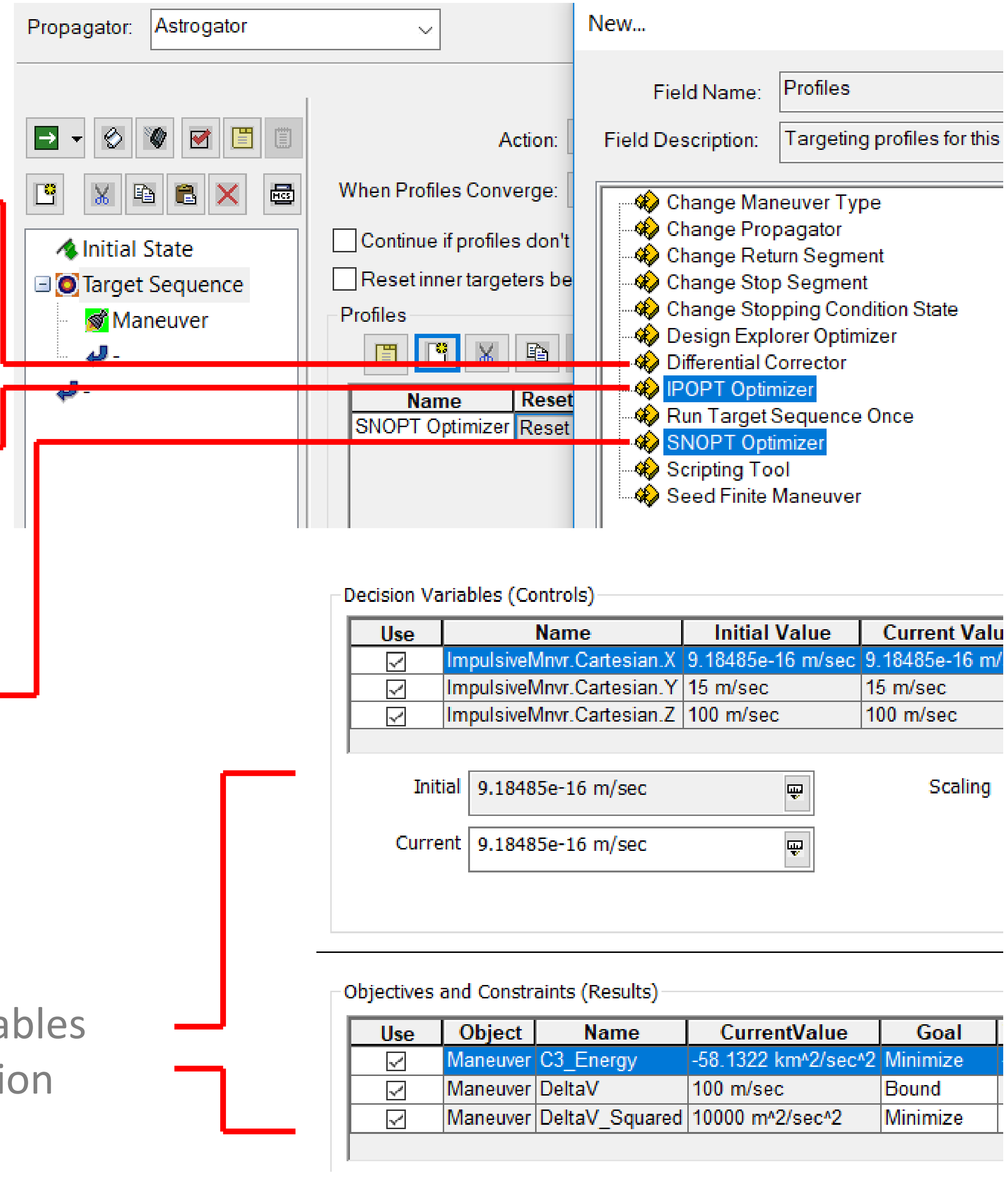
Astrogator

Target Sequence - Optimizers

- Differential **Corrector**
  - Root-finding
    (good to approximate a feasible solution)
- Interior Point **Optimizer** (IPOPT)
  - [Interior-point method](#)
    (good to obtain a local optimal)
- Sparse Nonlinear **Optimizer** (SNOPT)
  - Sequential quadratic programming
    (good to obtain a local optimal)

Non linear

Non linear



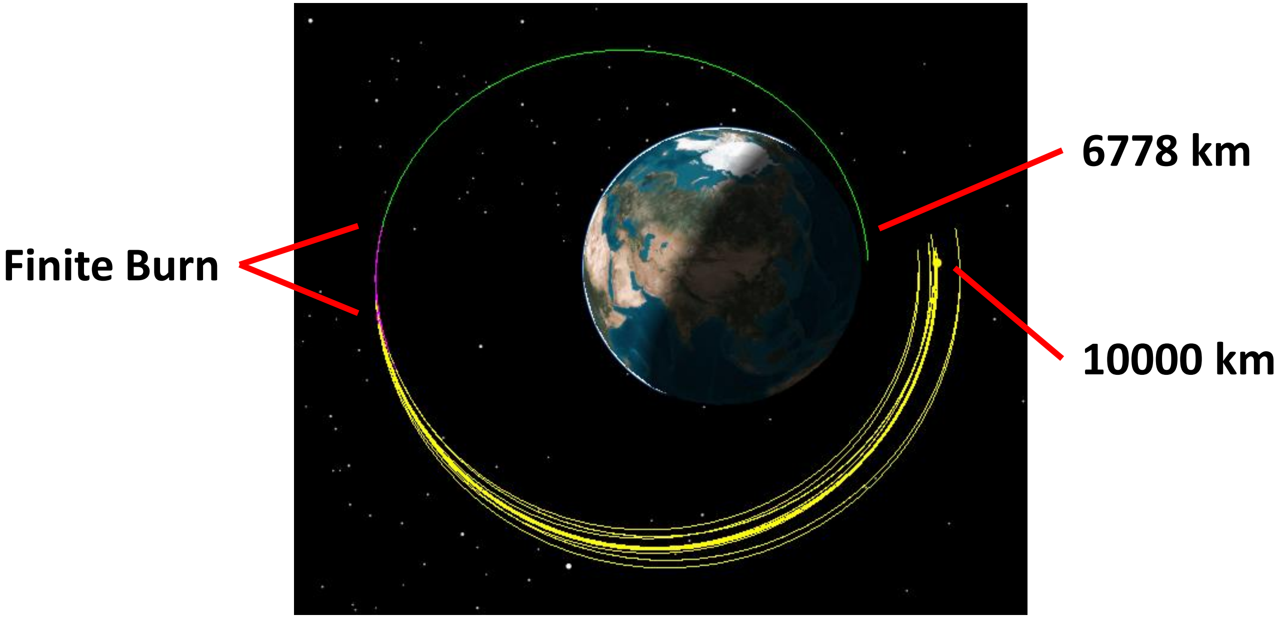
Independent variables (control) -> Decision variables

Dependent variables (results) -> Objective function

Astrogator

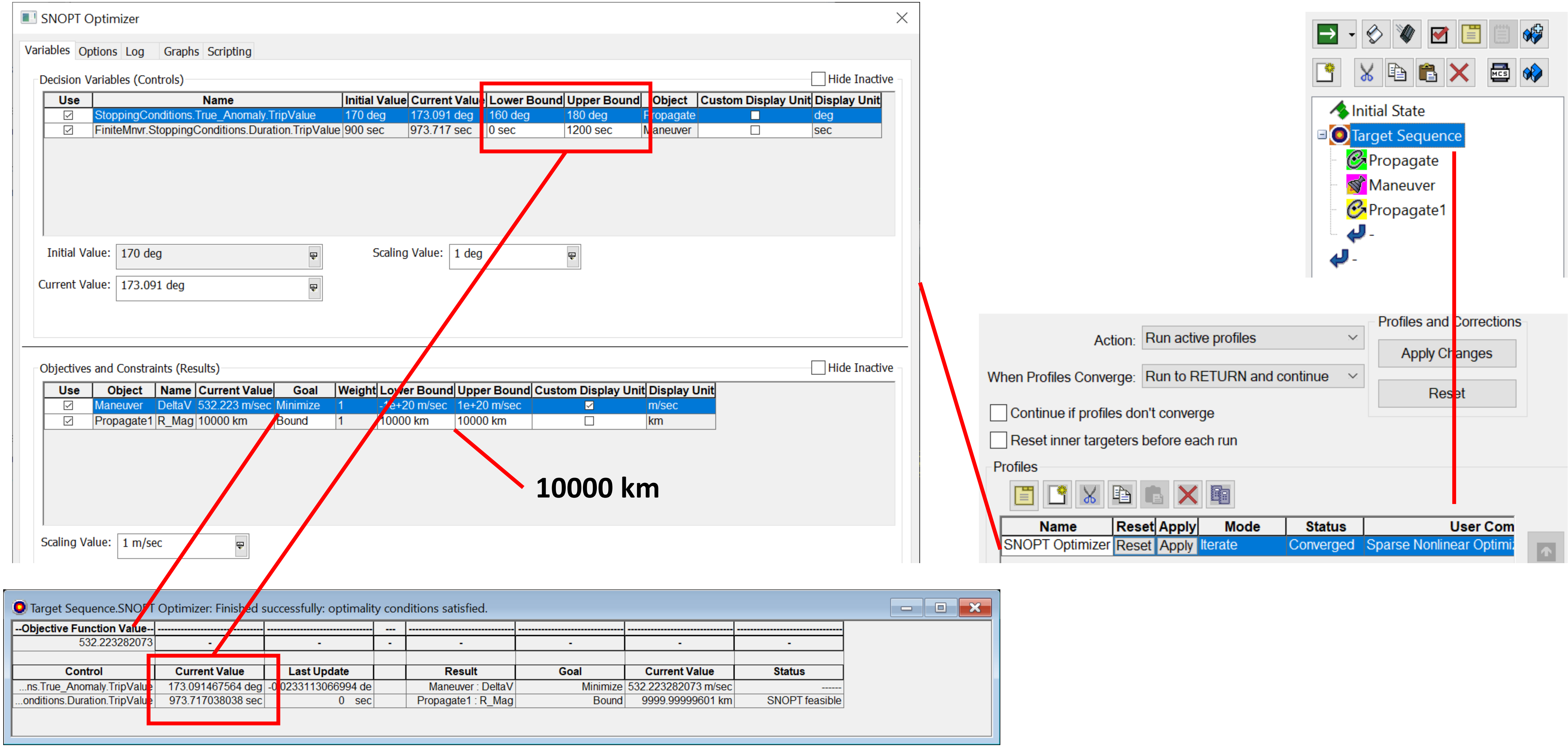
Target Sequence – Optimizers: Trajectory Design with SNOPT [Tutorial]

- Initial state (RAAN, Argument of Perigee, and True Anomaly = 0)
  - Radius of Perigee: **6778 km**, Eccentricity: 0.4 km, Inclination: 45 degree
- Objective
  - Propagate the orbit **until** a certain true anomaly (guess = 170 deg)
  - Use **finite** burn with **min delta V** | radius of perigee = **10000 km**



Astrogator

Target Sequence – Optimizers: Trajectory Design with SNOPT Tutorial



Questions?