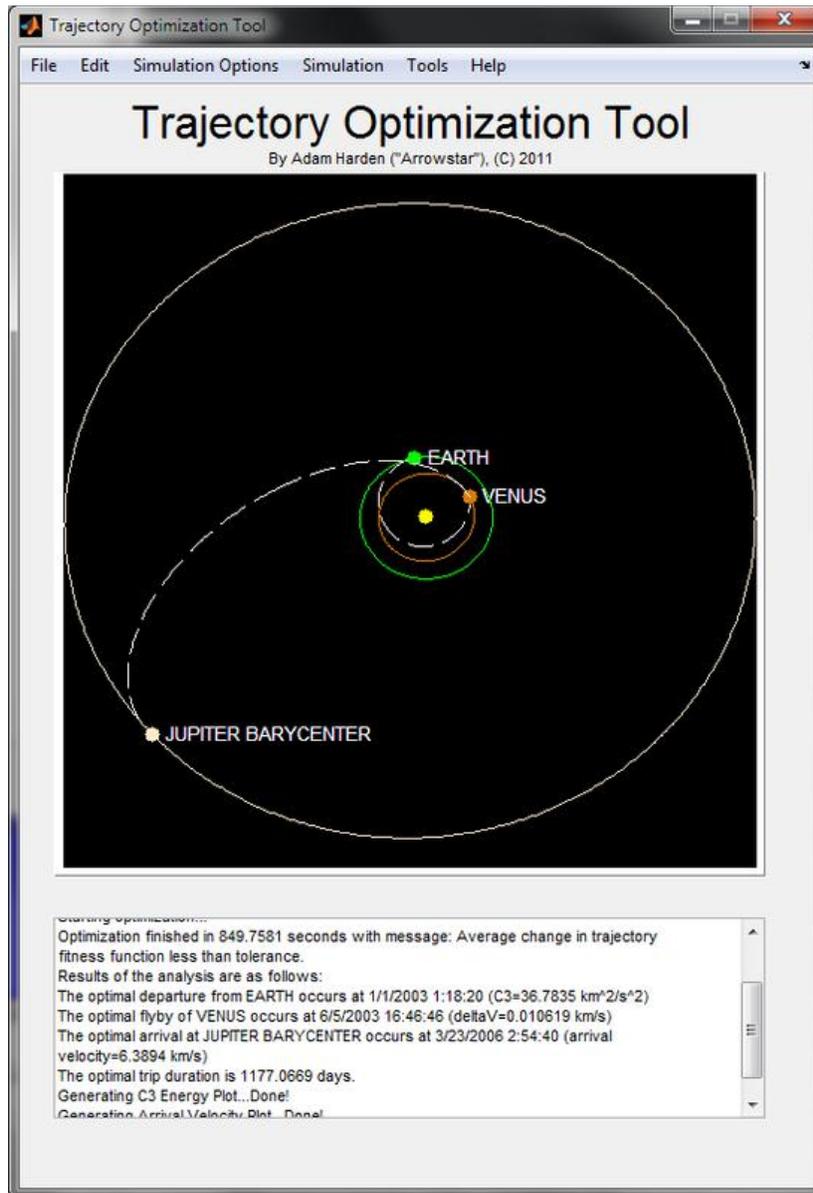


Trajectory Optimization Tool

Ver. 2.1.1



User's Guide

By: Adam Harden ("Arrowstar")

Contents

Overview	1
System Requirements	1
Version History.....	2
Installation	3
Installing the MATLAB Compiler Runtime (MCR)	3
Unpacking the Rest of the Files	3
Adding Additional Ephemerides Files	4
Overview	4
Installation	4
Using Trajectory Optimization Tool	5
User Interface Overview	5
Main User Interface	5
Flight Plan Editor	6
Flight Constraints	8
Optimization Options.....	11
Output Options	15
Other Notable Menu Items.....	18
Optimizing a Trajectory: An Earth-Mars Tutorial.....	18
Setting up the Flight Plan.....	18
Setting the Optimization Options	20
Setting the Output Options.....	21
Running the Optimization	22
Additional Flight Planning Tools	26
Using the C ₃ Energy Plot Generator	26
Using the Arrival Velocity Plot Generator.....	28
Using the Departure Analysis Tool.....	29
Using the Date Conversion Tool.....	30
Theory of Operation.....	31
Lambert's Problem.....	31
Transfer Types.....	33

Trajectory Optimization Tool ver. 2.1.1
User's Guide

Genetic Algorithms 33
Acknowledgements..... 34
Additional Reading..... 35

Overview

One of the most challenging aspects of piloting a spacecraft in Dr. Martin Schweiger's [Orbiter Spaceflight Simulator](#) is completing an interplanetary tour. There are numerous tools available that will assist in the planning of trajectories to accomplish such a feat. For example, the TransX MFD flight planning software is an excellent aid for developing interplanetary flight plans.

The purpose of the Trajectory Optimization Tool is not to replace MFD software available within Orbiter. Instead, the tool aims to be an aid for determining the optimal trajectory for reaching various interplanetary destinations within specified ranges of dates. This information can then be used within software such as TransX to actually carry out the flight plan. Here, "optimal" is defined three ways. First, the "optimal" trajectory away from the launch planet is the one that minimizes the "hyperbolic excess energy." Think of this energy as what the launch vehicle must provide the spacecraft to leave, say, Earth. The lower the hyperbolic excess energy required to reach a destination, the less kick the launch vehicle must give to the spacecraft. Second, when using a planet for a gravity assist maneuver, an "optimal" flyby is one that minimizes the total amount of delta V that must be provided by the spacecraft's engine to achieve the correct trajectory. If planned correctly, a spacecraft may never have to use its engine during these maneuvers, and the goal for planetary flybys is often zero delta V. Third, the slower the spacecraft is traveling when it reaches its destination, the less energy will have to be bled off to put the vehicle into orbit around the target. Thus, arrival speed with respect to the destination body should be minimized to the extent possible.

The Trajectory Optimization Tool uses a patched conics approximation to determine trajectories. Gooding's technique is used to solve Lambert's problem.

Ephemerides for the main solar system bodies are provided with this package, courtesy of JPL NAIF. In particular, the DE421 kernel is provided herein.

The Trajectory Optimization Tool is implemented in MATLAB 2010b. The MATLAB Compiler Runtime is required to use this software. See "Installation," below, for more details.

System Requirements

The Trajectory Optimization Tool is built with the MATLAB Compiler toolbox. As such, it shares many of the system requirements with this software. The system requirements for the Trajectory Optimization Tool are:

1. Any Intel or AMD x86 processor supporting the SSE2 instruction set;
2. 1024 MiB RAM (2048+ MiB recommended);
3. 450 MB of free hard drive space (including the MCR install); and
4. The MATLAB Compiler Runtime version 7.14 OR MATLAB 2010b.

Version History

Version 0.5 – Initial Public Release. Requires MATLAB 2009b to run.

Version 0.7 – Major Revision. Executes as a Windows standalone application. Addition of a GUI for program control. Requires MATLAB Compiler Runtime v7.14 to run. Numerous additional bug fixes in user interface, output code, and trajectory code. NOTE: The size of the download has increased significantly from version 0.5 to version 0.7. This is due to the large (~180 MiB) MATLAB Compiler Runtime that had to be included with the package.

Version 0.8 – Major Revision. Date ranges may now overlap, and dates may be entered as MJD. Text in output is now selectable. Dates on plots are now in MJD. Date step may now be entered as a positive floating point number. Maximum C3 Energy and Delta V added/updated. Central body now saved in Flight Plan file. Updated documentation. New demo flight plan. Other numerous minor bug fixes and alterations.

Version 1.0 – First “Release” Version. Added pause/resume button. Various buttons now go inactive during while calculations are in progress so they cannot be accidentally pressed while the program is running. New B-Plane angle steps added. New B-Plane search modes added. A major bug with the B-plane delta-V calculations was fixed. Other minor bug fixes and tweaks. This documentation updated.

Version 1.1 – Major Revision. The Lambert solver has been changed from p-iteration to a Gooding solver, which is more robust and should also be faster. Multiple revolution transfers are also possible using this solver now. The B-Plane fly-by search code has been heavily optimized for a five-fold increase in speed during this segment of calculations. The ability to output a plot of arrival velocities at the final body on the flight plan has been implemented. An estimated time remaining clock has been added. The legend on plots has been moved to below the abscissa on all plots. Other minor bug fixes and tweaks.

Version 1.1.1 – Bug-fix release. The zooming/panning issue with axis tick mark numbers on graphs should be eliminated. The B-Plane plot legend has been moved and updated to show additional information. Other minor tweaks.

Version 1.1.2 – Updated the size of the GUI box so that it can fit on monitors with resolutions less than 900 pixels.

Version 2.0 – Major Revision. Completely rewrote application from scratch. Now includes ability to globally optimize arbitrary trajectory using genetic algorithms. Completely new user interface. Ability to generate individual pork chop plots independent of optimization run. Trajectories are now plotted as a visual aid. Documentation completely rewritten. Numerous other rewrites, updates, and corrections.

Version 2.1 – Major Revision. Added the Departure Analysis Tool, Date Conversion Tool, and Flight Constraints to the software. Numerous bug fixes and updates to the analysis code, menus, and displays. Updated documentation.

Version 2.1.1 – Minor revision. Changed the working reference frame from J2000 (Earth mean equator) to J2000 (ecliptic plane). Added constraint options on C3 energy, fly-by Delta-V, and arrival speed. Changed the “pre-departure” reference frame in the Departure Analysis Tool (DAT) from J200 to the local, non-inertial (landmass-fixed) IAU reference frame for the departure body. Corrected how the DAT computed hyperbolic semi-major axes. Added additional hard constraints to DAT optimization. Revised documentation. Procured new TOT logo.

Installation

There are two steps that must be completed before the Trajectory Optimization Tool can be used. First, the MATLAB Compiler Runtime must be installed. Second, the tool and its accompanying files must be unpacked.

To begin, unpack the ZIP file you downloaded to a new directory and open this new directory.

Installing the MATLAB Compiler Runtime (MCR)

The MATLAB Compiler Runtime is “a standalone set of shared libraries that enable the execution of compiled MATLAB files.” You can think of it as MATLAB’s equivalent to Microsoft’s .NET Framework or Java. Because the Trajectory Optimization Tool was created in MATLAB, the MCR is needed to run it.

Version 7.14 of the MCR is packaged with this tool.

Double-click the file “MCRInstaller.exe” and follow the prompts to install the MCR. During the installation, you may be prompted that your system does not have the .NET Framework installed. If this happens, you may ignore this warning by selecting “Okay”, as the Trajectory Optimization Tool does not require the .NET Framework.

After the MCR software is installed, you will see a screen similar to the image on the right. Tap finish to complete the installation.



Figure 1: MATLAB Compiler Runtime Install

Unpacking the Rest of the Files

If you unpacked everything from the ZIP before installing the MCR, you have completed this step. If you only unpacked the MCR package before installing it, unpack the rest of the files in the ZIP file now. Once that has been accomplished, installation is complete. Test your installation of the MCR and the Trajectory Optimization Tool by starting trajOptTool2.exe and seeing if it loads without errors. If it does,

congratulations, you are done! If it does not, consider reinstalling the MCR and re-downloading the full ZIP file.

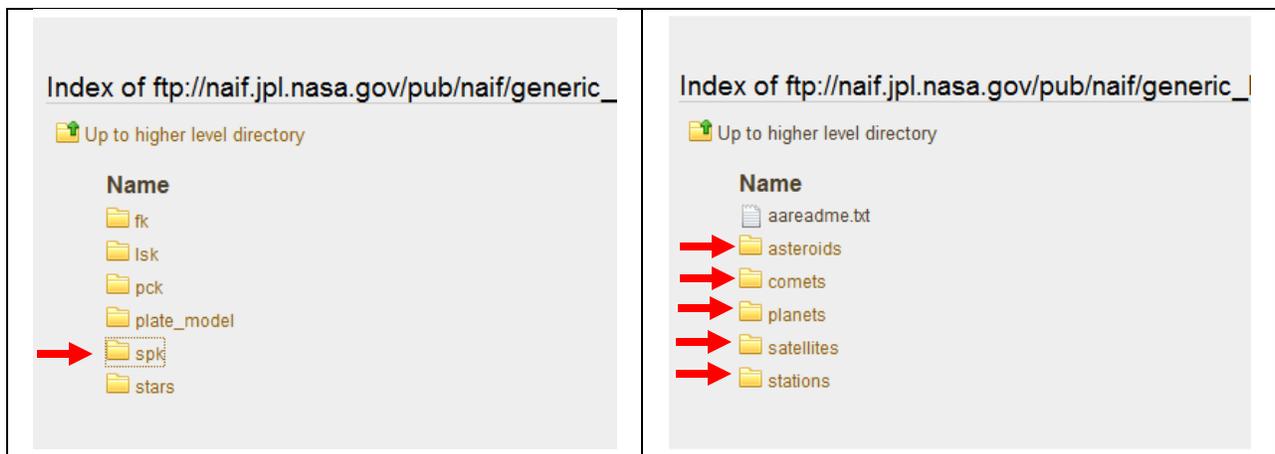
Adding Additional Ephemerides Files

Overview

The only bodies in the DE421 ephemerides that are loaded by default are the Sun, the eight planets in the solar system, the Moon, and Pluto. However, it is certainly conceivable that a trajectory directly to another solar system body, such as a comet or asteroid, might be desired. In order to accomplish this, additional ephemerides will have to be downloaded from the JPL NAIF website and loaded with MATLAB.

Note that, at this time, only Sun-centered trajectories are possible. Therefore, moons of major solar system bodies (such as Europa and Io) should not be selected as destinations, even if the Trajectory Optimization Tool loads their ephemerides. Instead, select the body those moons orbit around as the destination.

Additional ephemerides kernels can be downloaded from the JPL NAIF website at: ftp://naif.jpl.nasa.gov/pub/naif/generic_kernels/. Select the SPK folder to access the ephemerides. Select the category of body you'd like to add (such as "satellites" for a moon of Jupiter). Ephemerides files have the file extension ".bsp" and information regarding what data is in each file can be found in the folder's summaries text file.



Installation

Installation of additional SPICE kernels is quite easy. Simply download the desired file and place it in the "SPICE_Libs" directory where the main program executable is. The code will automatically detect and load the new file on program start or clicking "Reload SPICE Kernels" in the toolbar's Edit menu.

Using Trajectory Optimization Tool

User Interface Overview

The following is an overview of the user interface with explanation of the major features thereof.

 **Warning!** When using the Trajectory Optimization Tool for the first time (or after a new installation/upgrade), you must open and then close each item in the Simulation Options menu: Flight Plan, Flight Constraints, Optimization Options, and Output Options. This will create the necessary data structures in the TOTdata.mat file.

Main User Interface

The main user interface of the Trajectory Optimization Tool contains three main items: a toolbar for accessing application controls, a window for viewing solar system body orbits and spacecraft trajectories, and an area for text output from the application.

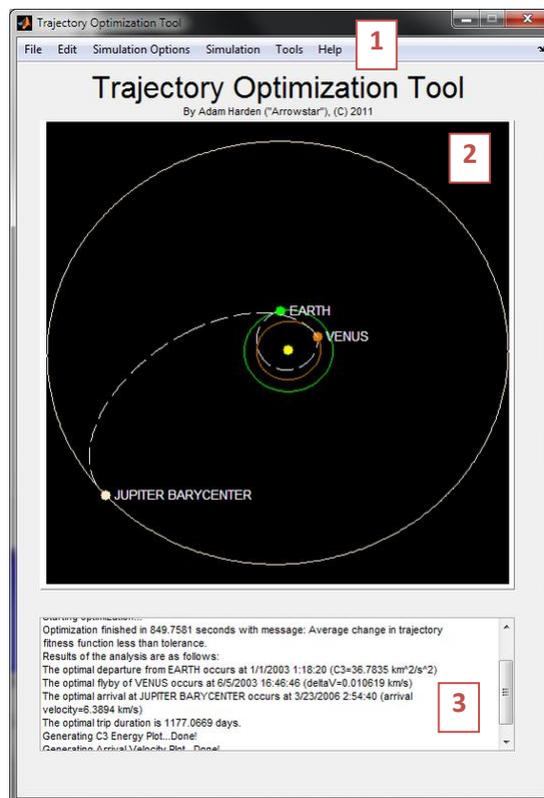


Figure 2: Main User Interface Window

1 - Toolbar

The toolbar contains options, settings, and commands for using the Trajectory Optimization Tool. More information on the noteworthy toolbar entries is below.

2 - Trajectory Visualization Window

This graph will show the orbit of body you have elected to display, the position of that body at some relevant time (either the time your spacecraft is present at that body or the present time if it does not

directly participate in the flight plan), and the computed trajectory the optimization routine has generated, if one exists.

The view shown in the window uses the J2000 reference frame and projects three-dimensional orbits onto the J2000 X-Y plane.

Right-clicking the window will give the option to save the current trajectory plot as a PNG image at a location of your choosing. Options for zooming and panning the graph can also be found by right-clicking on the window. Zooming and panning can be turned off by disabling the appropriate action under the toolbar's Edit menu.

3 - Text Output

Textual output from the application will be displayed here. Major statistics of each optimization run will be shown, as well as error messages and other text describing the activities of the application. Error messages will be accompanied by this box turning red. Text may be cleared by via the toolbar through **Edit > Clear Output Text** or tapping Ctrl-C.

Flight Plan Editor

Flight plans are the means by which you set the target destination of your trajectory, your departure point, and any bodies to swing by. The dates you wish to set as boundaries for the optimization routine are also applied here. Much of the information set in the Editor is passed directly to the Lambert solver.

A flight plan must be created before the optimization algorithm is run.

The flight plan editor may be accessed from the toolbar through Simulation Options > Flight Plan Editor or via Ctrl-F.

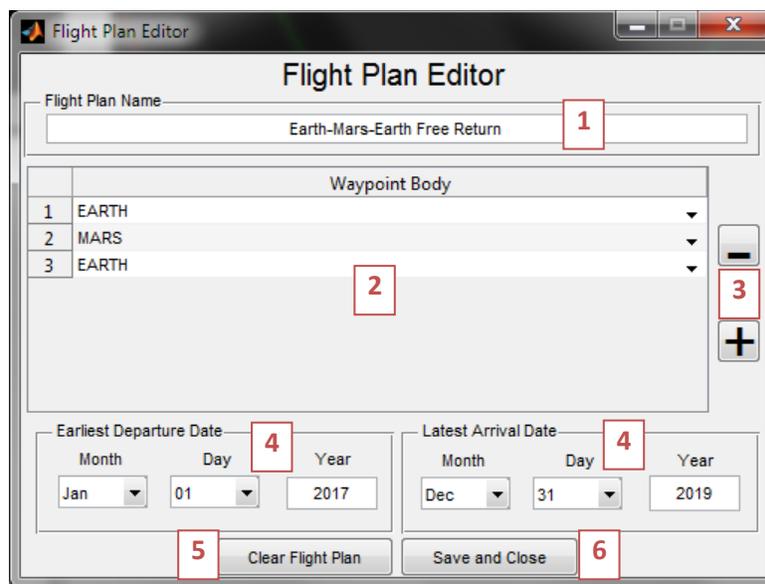


Figure 3: Flight Plan Editor GUI

1 – Flight Plan Name

Enter a name for your flight plan here. A name is required data. The name is used when generating optimization reports and displaying previous trajectories.

2 – Flight Plan Waypoint Information

Specific information about the trajectory being planned is entered here. At least two bodies must be in this list. The first body entered will be the body the spacecraft departs from. The final body entered is the ultimate arrival destination. Any other body entered between the first and last body will be swung by in a gravity assist maneuver. The optimization routine will not select bodies for gravity assist maneuvers, you must do so here.



Warning! At this time, only solar system bodies that primarily orbit the Sun are supported as waypoints. This includes all eight planets, Pluto, asteroids, comets, and like bodies. It does *not* include moons or other satellites of these bodies. For example, Jupiter is a valid waypoint, but Europa (a moon of Jupiter) is not.

3 – Add/Remove Waypoint Buttons

These buttons add or remove lines in the flight plan table. Clicking the “+” button without any selected pre-existing rows will add a row to the end of the table. Clicking the “-” button without any selected pre-existing rows will delete the last row in the table. If a row, or part of a row, is selected when one of these buttons is selected, that row will either be deleted (by clicking the “-” button) or a row will be added after the selection (by clicking the “+” button). Multiple rows may be selected for these buttons.

There is no undo button and all deletions are irreversible!

4 – Earliest Departure & Latest Arrival Dates

Setting these dates provides boundaries for the trajectory optimization routine to search within. Dates are checked for validity before being saved. Initial departure, all gravity assists, and final arrival may occur at any point within the boundaries set, subject to the constraint that the spacecraft must arrive at waypoint n before it arrives at waypoint $n+1$ (where n is any positive integer). Obviously, the earliest departure date set here must occur before the latest arrival date.

Because of the typically long duration of spaceflight, be sure to provide for at least a year or two between these dates. More time may be necessary for flights into the outer solar system. Consult the timelines of previous missions to the final destination body to get a rough estimate of the minimum spacing required between earliest departure and latest arrival.

5 – Clear Flight Plan Button

This button clears the flight plan name, all waypoints and associated information, and resets the earliest departure and latest arrival dates. **This action is irreversible and may not be undone.**

6 – Save Flight Plan and Close Button

The “Save and Close” button checks the as-entered flight plan for errors and displays a notice in the main user interface text output area if issues are found. If no problems are discovered, the flight plan is saved to memory and the editor window is closed.

Flight Constraints

Flight constraints are mathematical restrictions on the optimized trajectory solution. The Trajectory Optimization Tool allows users the ability to set minimum and maximum flight times for a given trajectory, as well as minimum or maximum flight times between any two waypoints in the flight plan. Flight constraints are not required to optimize a trajectory, but they may be used to restrict a solution to within some user-defined concept of feasibility.

The flight constraints may be accessed from the toolbar through Simulation **Options > Flight Constraints** or via Ctrl-G.

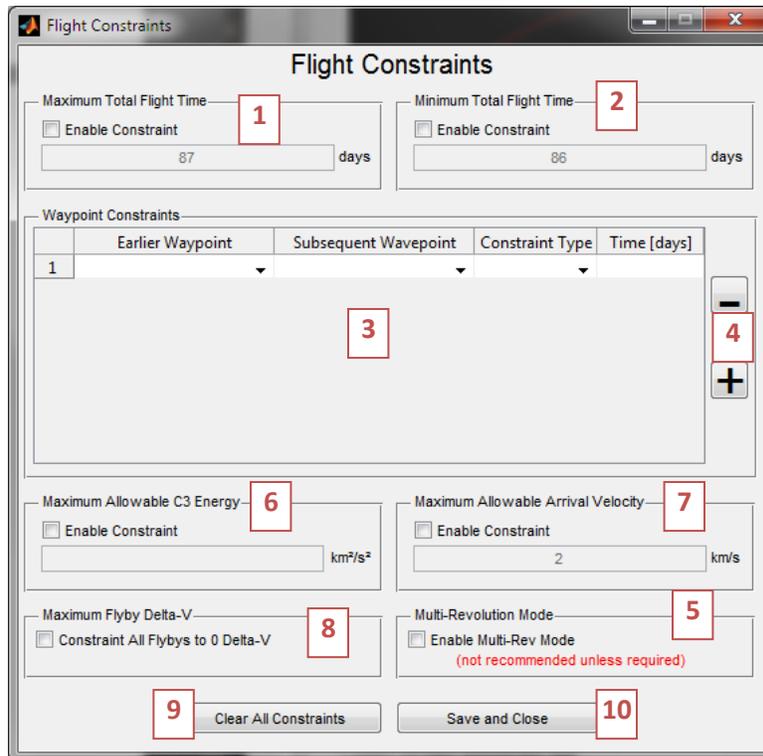


Figure 4: Flight Constraints User Interface

1 – Maximum Total Flight Time Constraint

This option will set a hard constraint within the optimizer, limiting solutions to those trajectories that can be fully accomplished in under the maximum time set. To enable to constraint, check the “Enable Constraint” box. If the box is checked, the text field underneath it must be filled in with a positive number of days. The number entered here is the maximum flight time allowed.

This constraint is optional and does not need to be set.

2 – Minimum Total Flight Time Constraint

This option will set a hard constraint within the optimizer, limiting solutions to those trajectories that can be fully accomplished in more time than set here. To enable to constraint, check the “Enable Constraint” box. If the box is checked, the text field underneath it must be filled in with a positive number of days. The number entered here is the minimum flight time allowed.

This constraint is optional and does not need to be set.

3 - Waypoint Constraints

The waypoint constraints options allow for minimum or maximum durations to be set between any two waypoints in the flight plan. To use, first select the "Earlier Waypoint." Doing so will populate the next column, "Subsequent Waypoint," with all waypoints that occur after the waypoint in "Earlier Waypoint." Select a waypoint from the "Subsequent Waypoint" column.

Next, select the constraint type. Two options exist, "less than or equal to" (" \leq ") and "greater than or equal to" (" \geq "). If "less than or equal to" is selected, the amount of time between the two selected waypoints will be less than the amount of time entered into the "Time" column. Likewise, if "greater than or equal to" is selected, the amount of time between the two selected waypoints will be greater than the amount of time entered into the "Time" column. The amount of time entered in the fourth column should be entered in units of days.

There is no limit to the number of waypoint constraints that may be created. As with the other constraints, waypoint constraints are optional.

4 - Add/Remove Waypoint Buttons

These buttons add or remove lines in the waypoint constraints table. Clicking the "+" button without any selected pre-existing rows will add a row to the end of the table. Clicking the "-" button without any selected pre-existing rows will delete the last row in the table. If a row, or part of a row, is selected when one of these buttons is selected, that row will either be deleted (by clicking the "-" button) or a row will be added after the selection (by clicking the "+" button). Multiple rows may be selected for these buttons. **There is no undo button and all deletions are irreversible!**

5 - Multi-Revolution Mode

Multi-revolution mode allows the Lambert Solver to find trajectories that require the spacecraft to make a full revolution around the central body. These solutions tend to require more flight time and may be more difficult to compute, but also open up trajectory options that would otherwise be unavailable. For example, it is possible to create trajectories that fly-by the same planet multiple times in a row using Multi-Revolution Mode.

It is recommended that Multi-Revolution Mode is kept disabled (by un-checking the box) unless such a trajectory is explicitly desired. Currently, the maximum number of full revolutions around the central body is limited to 1 using this mode. This may be increased to 2 in a future version.



Warning! Use of Multi-Revolution Mode may cause the optimizer to fail to find a valid solution. Do not enable this mode unless you are explicitly interested in finding trajectories that require at least one full revolution around the Sun.

6 – Maximum Allowable C3 Energy

Using this constraint will force the trajectory optimizer algorithm to only consider trajectories with departure C3 energies below the amount specified in the text box. To use this constraint, check the box next to the words “Enable Constraint” and enter a positive number in the text box.

	Warning! Use of this constraint may cause the optimizer to fail to find a valid solution. Do not set this constraint unrealistically low. For best results, initially select a high value and slowly reduce over multiple optimization runs.
	Warning! Use of this constraint may significantly increase the amount of time necessary to run the optimization algorithm. Only use when necessary.

7 – Maximum Allowable Arrival Velocity

Using this constraint will force the trajectory optimizer algorithm to only consider trajectories with arrival speeds below the amount specified in the text box. To use this constraint, check the box next to the words “Enable Constraint” and enter a positive number in the text box.

	Warning! Use of this constraint may cause the optimizer to fail to find a valid solution. Do not set this constraint unrealistically low. For best results, initially select a high value and slowly reduce over multiple optimization runs.
	Warning! Use of this constraint may significantly increase the amount of time necessary to run the optimization algorithm. Only use when necessary.

8 – Maximum Flyby Delta-V

Using this constraint will force the trajectory optimization algorithm to only consider trajectories in which the sum of the flyby Delta-V is 0 km/s (or very nearly so). To use this constraint, check the box next to the words “Enable Constraint” and enter a positive number in the text box.

	Warning! Use of this constraint may cause the optimizer to fail to find a valid solution. For best results, initially run the optimizer such that the weights on C3 energy and arrival velocity are 0 and leave this constraint disabled. If a 0 flyby Delta-V trajectory can be found in such a manner, then restore the C3 and arrival velocity weights, set the fly-by weight to 0, and enable this constraint.
	Warning! Use of this constraint may significantly increase the amount of time necessary to run the optimization algorithm. Only use when necessary.

9 – Clear All Constraints Button

Pushing this button will reset all entered constraints, but it will not disable any constraints that have been enabled. **This action is irreversible and may not be undone.**

10 – Save and Close

The “Save and Close” button checks the constraints for errors and displays a notice in the main user interface text output area if issues are found. If no problems are discovered, the constraints are saved to memory and the window is closed.

Optimization Options

The Optimization Options provide a means for the user to access some of the basic options MATLAB provides for the genetic algorithm that attempts to optimize trajectories in the Trajectory Optimization Tool. While not every option possible has been made front-facing, the six most useful have been given user-controllable settings. Be sure to read the chapter on genetic algorithms in the Theory of Operation section for more insight into genetic algorithms and how these settings affect operation.

Much of the information shown below will be excerpted from the MATLAB documentation on genetic algorithms, which provides a clear, concise explanation of these options. Those excerpts will be provided in light blue boxes with dashed outlines.

Optimization options must be set before the optimization algorithm is run for the first time.

The optimization options may be accessed via the toolbar through Simulation Options > Optimization Options or by tapping Ctrl-T.

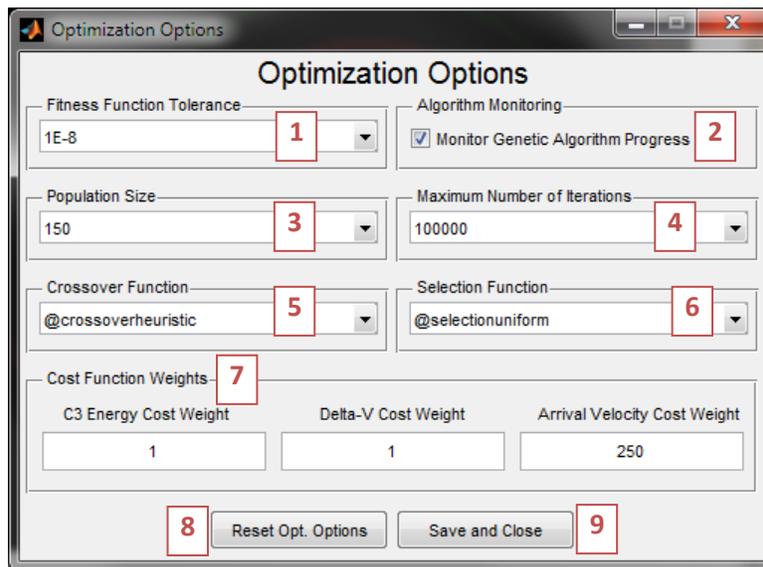


Figure 5: Optimization Options User Interface

1 - Fitness Function Tolerance

Function tolerance (`TolFun`) — The genetic algorithm runs until the cumulative change in the fitness function value over a set number of generations is less than or equal to **Function Tolerance**.

2 - Algorithm Monitoring

Allows for the user to monitor the output of the genetic algorithm. Monitoring may be used to determine how close the optimization run is to completing, how well the algorithm is doing, and may provide insight into setting the algorithm options. Two plots are provided that update at every iteration of the genetic algorithm: the current best “individual” in the “population” for each “generation”; and the stopping criteria levels.

Use of algorithm monitoring is **highly** recommended.

3 – Population Size

Population size (`PopulationSize`) specifies how many individuals there are in each generation. With a large population size, the genetic algorithm searches the solution space more thoroughly, thereby reducing the chance that the algorithm will return a local minimum that is not a global minimum. However, a large population size also causes the algorithm to run more slowly.



Warning! Setting the population size too low will severely limit the effectiveness of the optimizer. Likewise, setting the population size too high will use significantly more computation time and system resources. A population size of 100-500 tends to be sufficient for most problems.

4 – Maximum Number of Iterations

Also known as maximum number of generations.

Generations (`Generations`) — Specifies the maximum number of iterations for the genetic algorithm to perform.

5 – Crossover Function

Crossover options specify how the genetic algorithm combines two individuals, or parents, to form a crossover child for the next generation.

The following provides basic information concerning each of the crossover functions provided with the Trajectory Optimization Tool.

Heuristic (`@crossoverheuristic`) returns a child that lies on the line containing the two parents, a small distance away from the parent with the better fitness value in the direction away from the parent with the worse fitness value.

Scattered (`@crossoverscattered`), the default crossover function, creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child.

Intermediate (`@crossoverintermediate`) creates children by taking a weighted average of the parents.

Single point (`@crossoversinglepoint`) chooses a random integer n between 1 and **Number of waypoints** and then

- Selects vector entries numbered less than or equal to n from the first parent.
- Selects vector entries numbered greater than n from the second parent.
- Concatenates these entries to form a child vector.

`Two point (@crossovertwopoint)` selects two random integers m and n between 1 and **Number of waypoints**. The function selects

- Vector entries numbered less than or equal to m from the first parent
- Vector entries numbered from $m+1$ to n , inclusive, from the second parent
- Vector entries numbered greater than n from the first parent.

The algorithm then concatenates these genes to form a single gene.

`Arithmetic (@crossoverarithmetic)` creates children that are the weighted arithmetic mean of two parents. Children are always feasible with respect to linear constraints and bounds.

The suggested crossover function is the heuristic option. Tests have shown that the others often do not provide the same ability to minimize trajectory cost.

6 - Selection Function

Selection options specify how the genetic algorithm chooses parents for the next generation.

The following provides basic information concerning each of the selection functions provided with the Trajectory Optimization Tool.

`Uniform (@selectionuniform)` — Uniform selection chooses parents using the expectations and number of parents. Uniform selection is useful for debugging and testing, but is not a very effective search strategy.

`Remainder (@selectionremainder)` — Remainder selection assigns parents deterministically from the integer part of each individual's scaled value and then uses roulette selection on the remaining fractional part. For example, if the scaled value of an individual is 2.3, that individual is listed twice as a parent because the integer part is 2. After parents have been assigned according to the integer parts of the scaled values, the rest of the parents are chosen stochastically. The probability that a parent is chosen in this step is proportional to the fractional part of its scaled value.

`Stochastic uniform (@selectionstochunif)` — `Stochastic uniform` lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.

`Roulette (@selectionroulette)` — Roulette selection chooses parents by simulating a roulette wheel, in which the area of the section of the wheel corresponding to an individual is proportional to the individual's expectation. The algorithm uses a random number to select one of the sections with a probability equal to its area.

Tournament (@selectiontournament) — Tournament selection chooses each parent by choosing 4 players at random and then choosing the best individual out of that set to be a parent.

There is no suggested default selection function, though useful results have been generated with the Uniform and Stochastic Uniform methods. Some trials to determine the best selection function for your particular flight plan may be necessary.

7 – Cost Function Weights

The optimization algorithm calculates the cost of any particular trajectory as a function of its required C3 energy, its arrival velocity, any powered ΔV required for gravity assists, and the pass distance of each gravity assist. Mathematically, the cost function is:

$$cost = W_{C_3} (C_3)_{computed} + W_{V_{arrive}} (V_{arrive})_{computed} + \sum_{i=1}^n [W_{\Delta V} (\Delta V_i)_{computed}] + \sum_{i=1}^n [-(r_{swingby_i} - 1.1r_{body_i})]$$

Where:

cost = Total trajectory cost

$(C_3)_{computed}$ = Trajectory's computed C₃ energy

$(V_{arrive})_{computed}$ = Trajectory's computed final destination arrival speed

$(\Delta V)_{computed}$ = Computed powered ΔV needed for the i^{th} gravity assist

$r_{swingby_i}$ = Computed pass radius for the i^{th} gravity assist

r_{body_i} = Radius of i^{th} gravity assist body

W = Weight factor for subscripted quantity

The quantities entered in this box are the W values in the mathematical expression above. Assume that the units of each weight factor are the reciprocal of the actual computed quantity it multiplies such that each term in the equation, and thus the cost, are dimensionless. Weights may be any positive decimal number or 0. Setting a weight to zero effectively eliminates it from consideration when optimizing a trajectory.

Note that the costs entered here describe the *relative importance* of each quantity to you, the analyst. Thus, setting the arrival velocity weight to 3 while leaving the remaining two weights at 1 suggests (in simple terms) that the arrival velocity is three times more important than either the ΔV or the C₃ energy.

The recommended method of setting each weight is to assign weights such that the sum of the weights equals 10. This will mitigate extrema bias in your optimization results.

A note concerning the “pass distance cost,” the last term in the above equation. For each gravity assist, the expression in the parenthesis is evaluated. If the expression is positive and adds cost to the trajectory, the cost is returned as-computed. However, if the expression is negative, meaning that the

trajectory passes above $1.1r_{body}$, then the cost is set to 0. (The 1.1 is a safety factor to account for planetary atmosphere and other such effects.) This is to prevent exceedingly high swing-by maneuvers from canceling the other costs and forcing the genetic algorithm into non-optimal trajectories. The weight of 1 of these pass distance costs cannot be altered at the present time.

The genetic algorithm will attempt to come up with a trajectory that minimizes total cost. Thus, weighting some quantities higher will, in essence, tell the optimizer how to trade off one quantity for another when minimizing. Thus, weighting one quantity more than the others will force the algorithm to find a solution that more aggressively minimizes that quantity. The opposite is true for setting some weight factors lower than the others.

Given that acceptable C_3 energy values can be in the tens of km^2/sec^2 and acceptable ΔV and arrival velocities tend to be either be near zero or under 1 km/sec respectively, reducing the C_3 weight slightly may be necessary to find acceptable solutions. Adjusting weights by factors of 2 or 3 appears to be acceptable for first trials.

8 – Reset Opt. Options Button

Pushing this button will reset the optimization options to their defaults. **This action is irreversible and may not be undone.**

9 – Save and Close

Pushing this button saves the as-set optimization options to memory and closes the window.

Output Options

The Output Options dialog box provides for various ways of displaying and extracting results from the Trajectory Optimization Tool. These methods are primarily visual with an option for textual output, as well.

The output options may be accessed from the toolbar through Simulation Options > Output Options or via Ctrl-P.

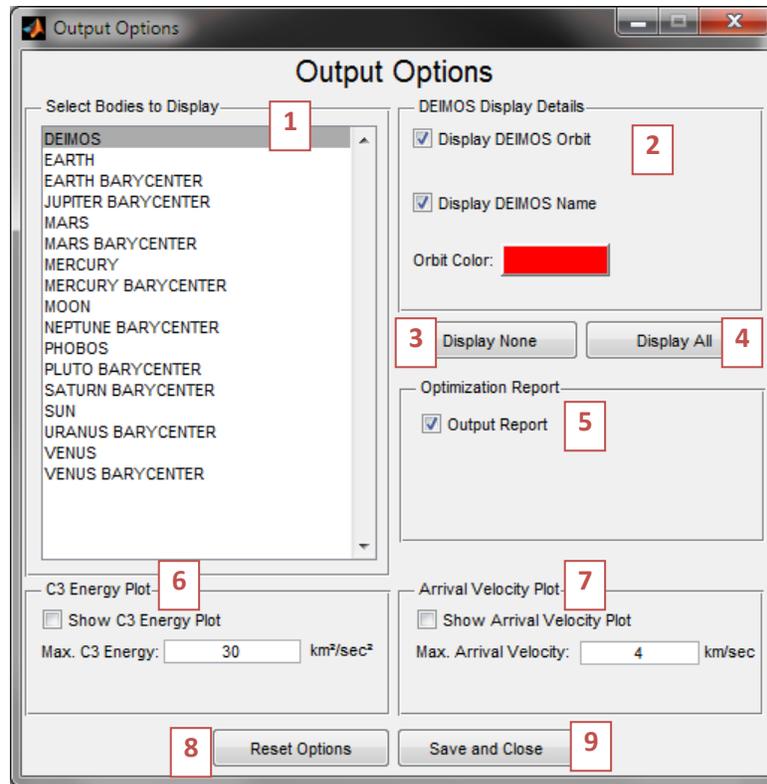


Figure 6: Output Options User Interface

1 - Display Bodies Selection

This section displays a single selection list of all the bodies that have valid ephemerides loaded into memory. Selecting a body in the list brings up the display details for that body in the area denoted by the boxed 2 in Figure 6.

Note that **there is a difference between a body and that body's barycenter!** The barycenter of a body is the center of mass of that body's local system, including the moons of that body. The gravitational parameter and trajectory of a barycenter are appropriately adjusted to account for this. As a simplification, the Trajectory Optimization Tool treats a body and its barycenter as distinct, separate bodies. It is recommended that flight plans and the corresponding display options use the body itself, not the barycenter, when available. Download additional ephemerides from the JPL website if you wish to use a body where barycenter information is only available by default.

2 - Body Display Details

Checking the "Display BODY Orbit" checkbox will cause that body's orbit to be plotted on the trajectory visualization window in the main user interface. To do so, the body's orbital elements are computed from its state vector at a particular time and those elements are used to draw the orbit. In addition to the orbit path, the location of the body is indicated as a colored dot on the orbit track. The date/time used to compute the location of the body is either the present time (if no optimized trajectory is being displayed or an optimized trajectory does not encounter that body) or the date/time of encounter with the body in an optimized trajectory.

If the “Display BODY Name” box is checked, the name of the body will be displayed next to its location.

The “Orbit Color” box displays the color that the orbit will be displayed in (text is always displayed white). Clicking the box will bring up a color selector that may be used to change the color of the orbit.

3 – Display None Button

Pushing this button unchecks the “Display BODY Orbit” checkbox for all bodies in the list on the left. No bodies will be displayed in the trajectory view.

4 – Display All Button

Pushing this button checks the “Display BODY Orbit” checkbox for all bodies in the list on the left. All the bodies loaded will be displayed in the trajectory view.

5 – Output Optimization Report

Checking this box outputs text reports that provide details and results concerning optimization runs. Data provided includes the dates and times of encounters with bodies, the cost of each leg of the trajectory, orbital elements for each leg, and data on the genetic algorithm.

Producing a report after each optimization run is suggested as there is no other way to recover cost data after a new run has been started or the text output display has been cleared. Reports are stored in the /Reports/ folder in the directory that you run the Trajectory Optimization Tool from and have the flight plan name and a timestamp in the file name.

6 – Generate C3 Energy Plot

Checking the “Show C3 Energy Plot” will output a plot of C3 energy (a “pork chop plot”) with respect to departure and arrival date for the first two bodies on the flight plan waypoint list. The maximum C3 energy is the C3 energy at which the computed C3 energy is no longer plotted above. Setting this option lower will provide higher resolution in the lower C3 energies. If the C3 energy plot comes out empty, set this value higher.

7 – Generate Arrival Velocity Plot

Checking the “Show Arrival Velocity Plot” will output a plot of arrival velocity with respect to departure and arrival date for the last two bodies on the flight plan waypoint list. The maximum arrival velocity is the arrival velocity at which the computed arrival velocity is no longer plotted above. Setting this option lower will provide higher resolution in the lower arrival velocities. If the arrival velocity plot comes out empty, set this value higher.

8 – Reset Options Button

Pushing this button will reset the output options to their defaults. **This action is irreversible and may not be undone.**

9 – Save and Close Button

Pushing this button saves the as-set output options to memory and closes the window.

Other Notable Menu Items

Other notable menu items in the toolbar include:

File > Open Flight Plan: Opens a previously saved flight plan and its associated options and stored results.

File > Save Flight Plan: Saves the current flight plan, its associated options, and any stored results. This is useful for storing results or a particular flight plan. Note that the currently stored options are stored in a permanent file in the same directory as the Trajectory Optimization Tool executable, totData.mat.

Simulation > Optimize Trajectory...: This menu item actually runs the optimization that has been setup in the previously described options.

 **Warning!** The optimizer should be run at least three times on the same problem setup to determine if the optimized solution is in fact optimal. The genetic algorithm is a heuristic brute-force technique and optimality of the returned solution is not guaranteed.

Help > User's Manual (F1): Opens this manual. Tapping F1 on the keyboard also opens this manual.

Optimizing a Trajectory: An Earth-Mars Tutorial

This section will serve as a tutorial for setting up a basic flight plan and determining a trajectory that optimally accomplishes that flight plan. The simple case used will be an Earth to Mars trajectory that departs in 2003. This tutorial can be a jumping off point for more advanced flight plans.

Setting up the Flight Plan

The first step in generating a trajectory is to setup the flight plan. In this case, our objective is to leave Earth and arrive at Mars with the minimum amount of C3 energy required. Open the Flight Plan Editor in the toolbar via **Simulation Options > Flight Plan**. If the Flight Plan Editor has options currently set, click "Clear Flight Plan" to reset the flight plan editor.

Let's name the flight plan "Earth to Mars 2003." Enter this into the Flight Plan Name box near the top of the editor. See Figure 7.



Figure 7: Flight Plan Name "Earth to Mars 2003"

Now we get to the meat of the flight plan. On line 1 of the flight plan waypoint list, under the Body column, drop down the menu and find "EARTH." Select "EARTH" as the body on line 1. Add a new waypoint by selecting the large "+" to the right. On line 2, set the Body to "MARS."

After these steps, your Flight Plan Editor should look similar to Figure 8.

Trajectory Optimization Tool ver. 2.1.1
User's Guide

Waypoint Body	
1	EARTH
2	MARS

Figure 8: Flight Plan Editor Waypoints - Earth to Mars

Now set the earliest departure date and the latest arrival date. Because this is a tutorial, we know in advance that there is a suitable transfer window to Mars from Earth in 2003. If this knowledge is not known in advance, then the Generate C3 Energy Plot and the Generate Arrival Velocity Plot tools under the **Tools** menu should be run in advance to determine when optimal transfers might take place. In this case, however, set the earliest departure date to January 1, 2003 and the latest arrival date to December 31, 2004.

Once the dates have been entered, your flight plan editor should look like Figure 9.

The screenshot shows the 'Flight Plan Editor' window. At the top, the title bar reads 'Flight Plan Editor'. Below the title bar, the main window has a title 'Flight Plan Editor' and a text field for 'Flight Plan Name' containing 'Earth to Mars 2003'. Below this is a table with the following data:

Waypoint Body	
1	EARTH
2	MARS

To the right of the table are minus and plus buttons. Below the table are two date selection sections. The 'Earliest Departure Date' section has dropdowns for 'Month' (Jan), 'Day' (01), and 'Year' (2003). The 'Latest Arrival Date' section has dropdowns for 'Month' (Dec), 'Day' (31), and 'Year' (2004). At the bottom are two buttons: 'Clear Flight Plan' and 'Save and Close'.

Figure 9: Tutorial Flight Plan Editor – Earth to Mars 2003

Tap the “Save and Close” button to save the flight plan and close the window.

Setting the Optimization Options

With the flight plan set, we can move on to setting the optimization options. Open the Optimization Options window by going to **Simulation Options > Optimization Options**. Tap the Reset Opt. Options button to set everything back to the default setting.

First, let's set the Fitness Function Tolerance to "1E-6". This is scientific notation for 1×10^{-6} , and it should provide a tight enough tolerance on our cost function that we get close to the true minimum while not drawing out the optimization calculations longer than they need to be. Also, check the "Monitor Genetic Algorithm Progress" box so we can watch the algorithm home in the optimal trajectory.

Setting the Population Size and Maximum Number of Iterations is next. Since this is a simple problem, we can probably get away with a smaller population size. Set the Population Size to 100. If the problem was more complex, a greater population size would be more efficient in searching the range of trajectory possibilities. Set the Maximum Number of Iterations to Inf (meaning "Infinity"). I like to set this value high so that the maximum number of iterations is never exceeded, as the results returned from that optimization run will not be optimal.

Now we select the genetic algorithm sub-functions. Leave the crossover function at "@crossoverheuristic" in order to use the heuristic crossover function. This option seems to work best with trajectory optimization and I use it for all of my optimization runs. The selection function is a bit looser, but let's go with the "@selectionstochunif" setting, which calls the Stochastic Uniform selection function. As an exercise, you can later try the "@selectionuniform" option to see if it changes the results.

Finally, let's set the cost function weights. Leave the C3 energy cost weight at 1. This Earth-Mars window is particularly cheap on fuel and there won't be much need to minimize C3 energy here. Of course, this is a priori knowledge, but in general I find setting the C3 cost weight at 1 or below (0.5, 0.25, etc) is more useful than setting it greater than 1.

Set the Arrival Velocity Cost Weight to 2. Since we're going to Mars, which has a thin atmosphere, it behooves us as mission planners to minimize the amount of energy we need to expend to stop at the planet. You might also later try setting this to 3, 5, or 10 and comparing the results to this run.

Since we have no gravity assists, the Delta-V cost weight will not be used: set it as you will. I prefer to leave it at 0.

Once your optimization options are set, tap the "Save and Close" button. Your Optimization Options window should look like Figure 10.

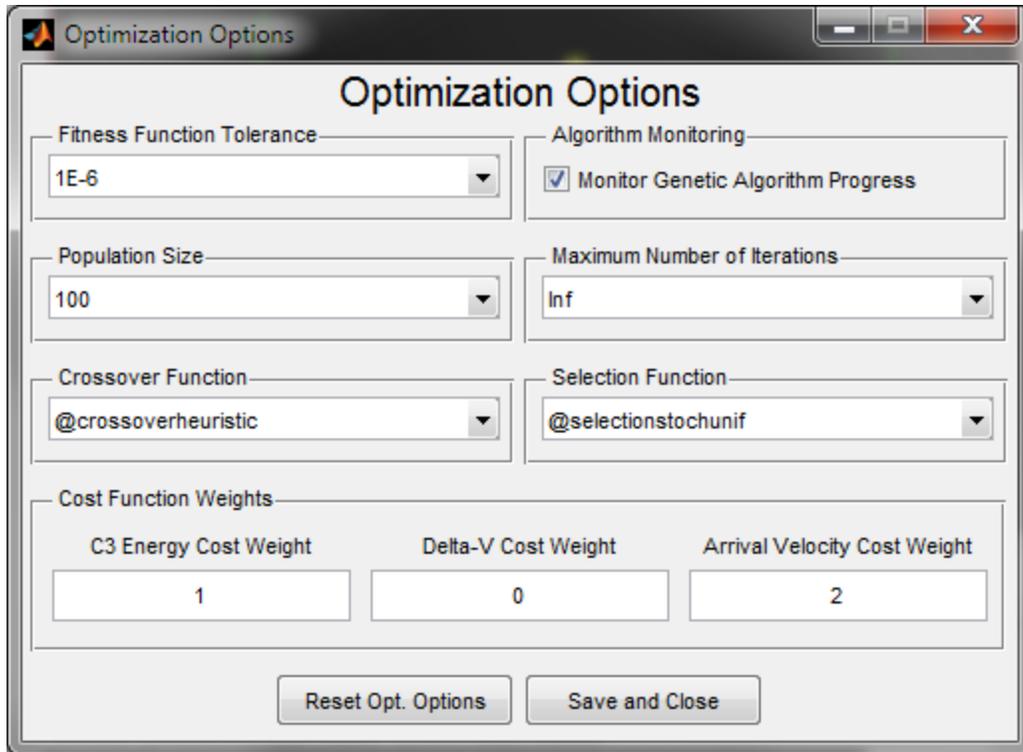


Figure 10: Tutorial Optimization Options - Earth to Mars 2003

Setting the Output Options

Finally, we need to set the output options. Let's start with the trajectory display window and the bodies to be displayed there. Since this is a simple Earth to Mars transfer, we'll likely just want those two bodies. We might also include the Sun as a point of reference. To start, click the "Display None" button to reset all the bodies to an inactive display state.

Find "EARTH" in the bodies list and select it. Check the "Display EARTH Orbit" box and the "Display Earth Name" box. I like to set the orbit color for Earth to green, but you may select the color you find most suitable. Either way, change the color from the default red.

Now find "MARS" in the bodies list and select it. Again, check both checkboxes to display the orbit and name. You may change the orbit color if you desire. I typically leave the default for Mars, given that it is the "Red Planet."

Let's add the Sun to the center of our solar system. Find "SUN" in the list and select it. Check the "Display SUN Orbit" box but leave the name display box empty. We really don't need to label the Sun, as it's the only body in the center of the solar system. Change the orbit color to a nice yellow for aesthetics' sake. That completes the visual portion of the output options.

Under the Optimization Report box, check the box to generate a report. This will be useful for review later.

Finally, check the Show C3 Energy Plot box and set the Max C3 to 30. Do the same for the Arrival Velocity plot, but set the Max Arrival Velocity to 4.

Once you've completed this step, your output options window should look like Figure 11.

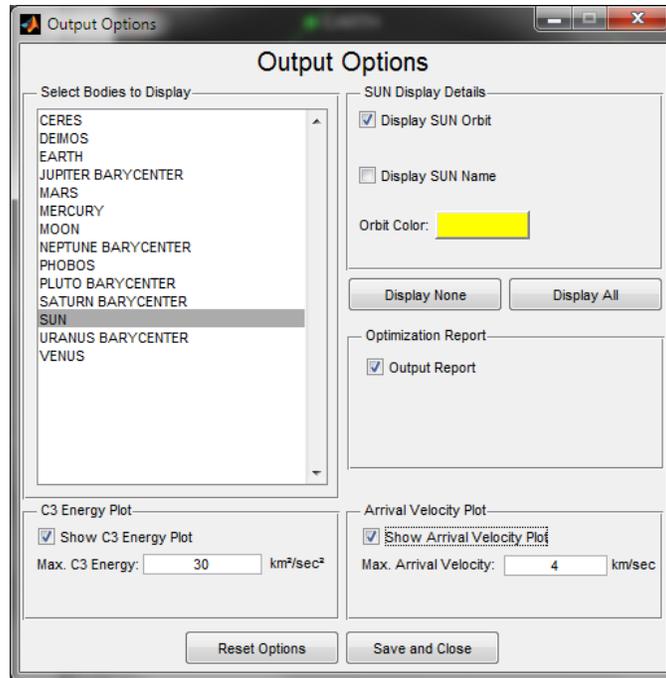


Figure 11: Tutorial Output Options - Earth to Mars 2003

Tap the “Save and Close” button to store the settings.

Flight Constraints

For this trajectory, it is not necessary to impose any additional constraints on the optimization. You may, however, try various constraints after successfully completing this optimization run without them. It will be beneficial to understand how each type of constraint affects the behavior of the solver.

Running the Optimization

With all the options set as described above, let's start optimizing the trajectory. Remember, all we've told the genetic algorithm is where we want to go and when we want to do it, not how to execute the flight plan. The algorithm will be coming up with this on its own.

Feel free to double check the settings we put in place in the previous pages now. If you're satisfied with the way things are, go up to the toolbar and tap **Simulation > Optimize Trajectory...** or alternatively tap Ctrl-1.

Since we've selected to monitor the genetic algorithm, a window such as the following should pop up (Figure 12):

Trajectory Optimization Tool ver. 2.1.1 User's Guide

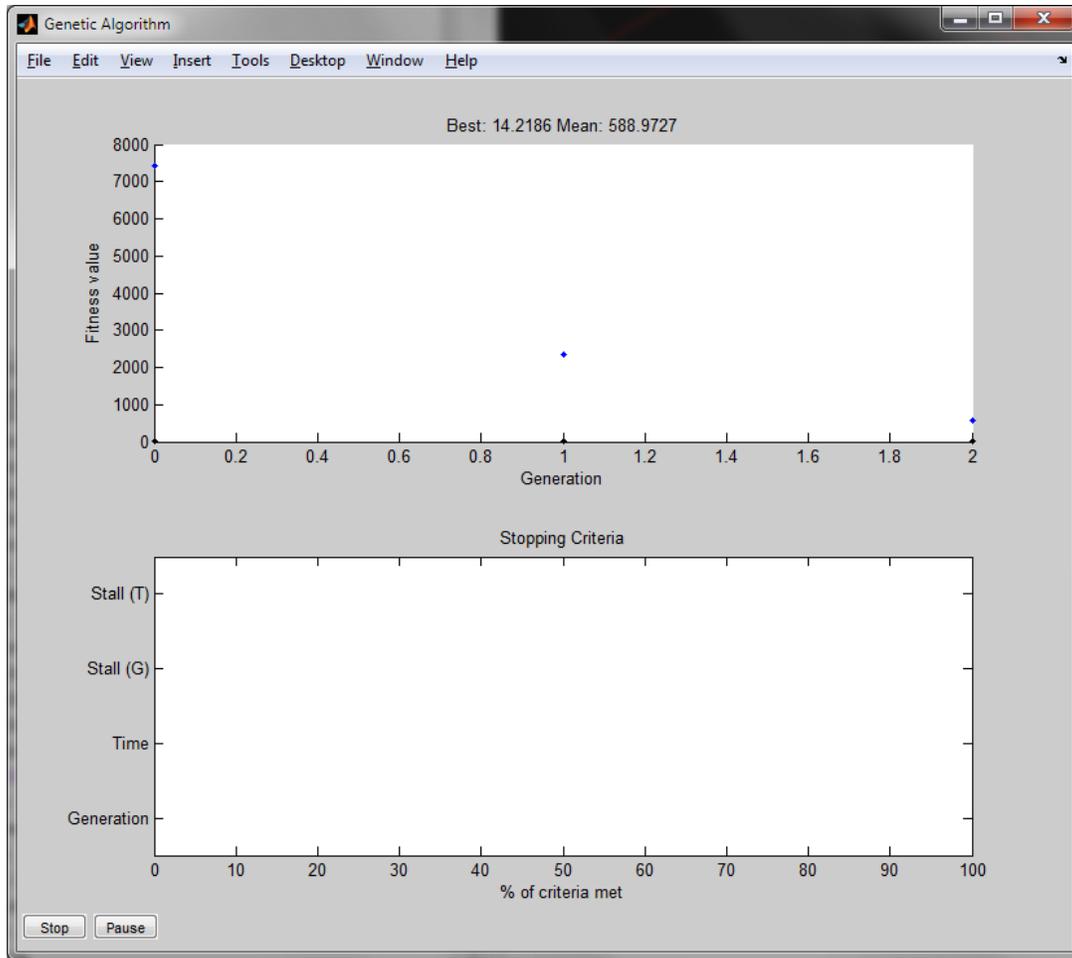


Figure 12: Tutorial Genetic Algorithm Monitoring at Start - Earth to Mars 2003

As the algorithm works, you should see the value of the right-most black point decrease. (The black points represent that generation's best trajectory cost.) Continue watching the plot until it ceases changing and text appears in the main user interface output area. This signifies that the trajectory has finished. At this point, your monitoring plot should appear similar to Figure 13.

Trajectory Optimization Tool ver. 2.1.1
User's Guide

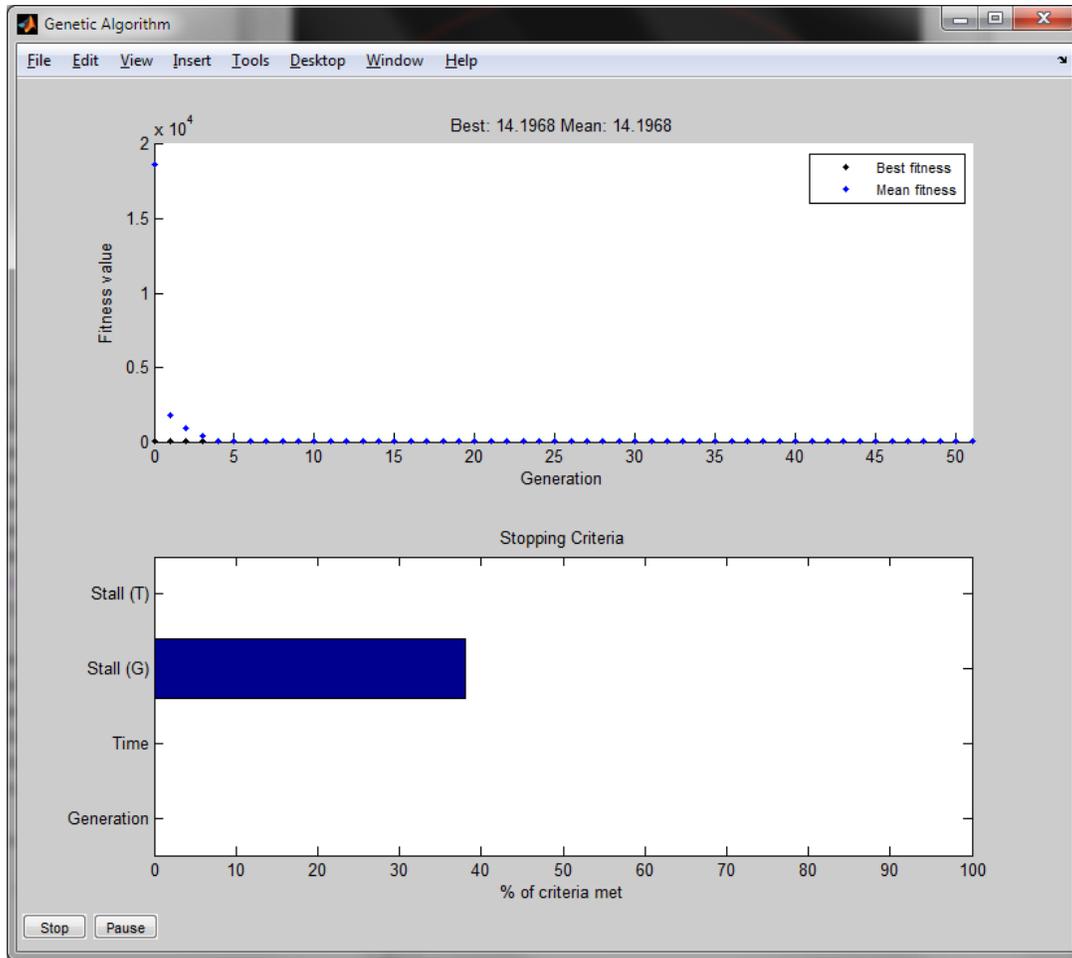


Figure 13: Tutorial Genetic Algorithm Monitoring at Completion - Earth to Mars 2003

Check out the text output in the main user interface. You should now have some figures there. Your text readout should be similar to this:

```
Starting optimization...
Optimization finished in 168.1093 seconds with message: Average change in trajectory fitness
function less than tolerance.
Results of the analysis are as follows:
The optimal departure from EARTH occurs at 6/5/2003 22:30:40 (C3=8.7903 km^2/s^2)
The optimal arrival at MARS occurs at 12/26/2003 14:59:11 (arrival velocity=2.7032 km/s)
The optimal trip duration is 203.6865 days.
Generating report file...Done!
```

I bolded the important lines in the above output. A value of $8.8 \text{ km}^2/\text{sec}^2$ for the C_3 energy is very good, and the arrival velocity of 2.7 km/sec is not terrible, either. Remember, this is arrival velocity **with respect to Mars**, not the Sun.

You should also have a plot of the optimal trajectory that appears very similar to Figure 14.

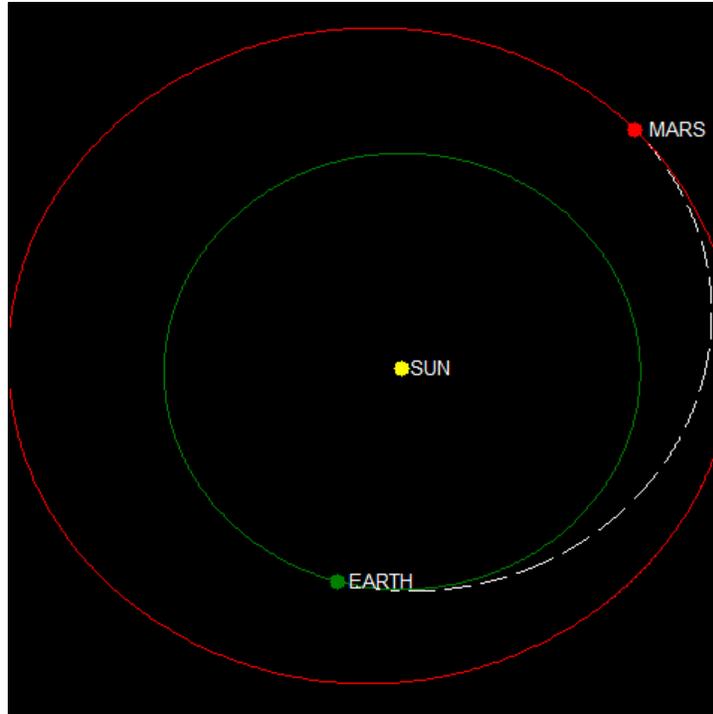


Figure 14: Tutorial Trajectory Plot Output - Earth to Mars 2003

The white dashed line is the trajectory that we had computed for us. Notice that the trajectory is very parallel to the path of the orbits of Earth and Mars at its departure point and arrival point, respectively. This is an indicator of a nice, low cost trajectory.

Finally, we have a report that was generated. Browse to the folder with Windows Explorer where you run the Trajectory Optimization Tool and navigate to the reports folder. You should see a file called "EarthtoMars2003_xyz.txt" where the "xyz" is a string of numbers that represents the time the report was generated. Open this file. Its contents should be similar to the following:

```
TRAJECTORY OPTIMIZATION TOOL v2
Optimization Report - 16-Aug-2011 18:37:37
Flight Plan: Earth to Mars 2003
-----

RESULTS
-----
OPTIMUM EARTH DEPARTURE DATE: 6/5/2003 22:30:40 - C3: 8.7903 km2/sec2
OPTIMUM MARS ARRIVAL DATE: 12/26/2003 14:59:11 - Body-centric Arrival Velocity: 2.7032
km/sec

TOTAL COST: 14.1968325

ORBITAL ELEMENTS - J2000 Reference Frame
-----
```

```
LEG 1
-Semi-major Axis: 188405456.2851462 km
-Eccentricity: 0.1943230
-Inclination: 23.4691145 deg
-Long. of Ascending Node: 0.2764686 deg
-Argument of Periapse: 253.9311254 deg
-True Anomaly (start): 0.6248998 deg
-True Anomaly (end): 153.7514758 deg

GENETIC ALGORITHM INFORMATION
-----
Termination Reason: Optimization terminated: average change in the fitness value less than
options.TolFun.
Number of Generations Computed: 512.0
Number of Fitness Function Evaluations: 51201.0
Random Number Generator Type: mt19937ar
```

And that's it. You've successfully setup a trajectory optimization problem, run the optimization algorithm, and viewed the results. Congrats! Now feel free to experiment with more complex and exotic flight plans.

Additional Flight Planning Tools

There are four additional tools in the Trajectory Optimization Tool. These are a C₃ Energy Plot Generator, an Arrival Velocity Plot Generator, a date conversion tool, and the Departure Analysis Tool.

Using the C₃ Energy Plot Generator

The C₃ Energy Plot Generator can be accessed from the toolbar via **Tools > Generate C3 Energy Plot**. When this menu item is selected, the following window appears (Figure 15).

Trajectory Optimization Tool ver. 2.1.1 User's Guide

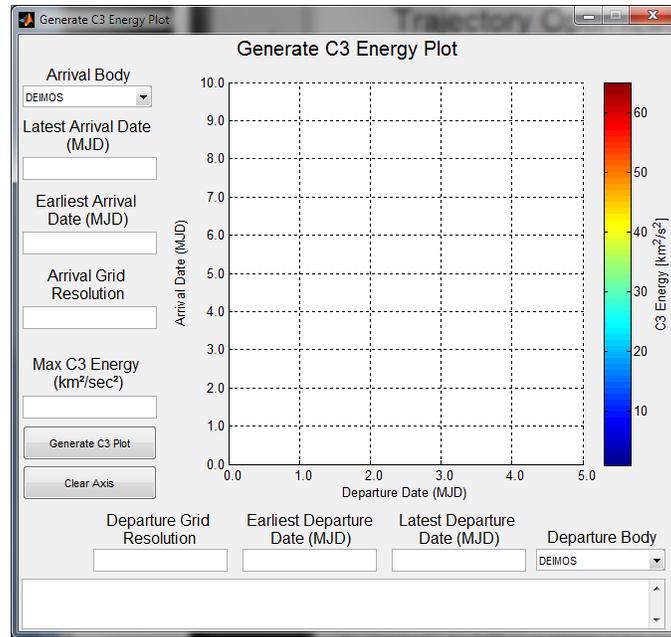


Figure 15: C₃ Energy Plot Generator Window

The options shown in the window required to generate a C₃ energy plot are discussed below.

Arrival Body: The body to which you are going. May also be known as the “destination body.”

Departure Body: The body from which you are leaving. May not be the same as the arrival body. Both the departure and arrival bodies should have as their central body (the body they both orbit) the Sun for this release of the ToT.

Arrival Dates: The range of dates at which arrival to the Arrival Body is possible. The earliest arrival date must occur before the latest arrival date. Dates are entered in Modified Julian Day (MJD). There are numerous tools online to convert dates from the Gregorian calendar (month, day, year) to Modified Julian Day. MJD is also the system Orbiter uses for time keeping.

Departure Dates: The range of dates at which departure from the Departure Body is possible. The earliest departure date must occur before the latest departure date. Dates are entered in Modified Julian Day (MJD). There are numerous tools online to convert dates from the Gregorian calendar (month, day, year) to Modified Julian Day. MJD is also the system Orbiter uses for time keeping.

Grid Resolutions: The number of data points along the respective axis (departure/arrival) to compute. The total number of data points that have to be computed to generate the plot is the product of the two resolutions. Resolutions must be a positive integer.

Maximum C₃ Energy: The maximum C₃ energy to plot on the graph. Any C₃ values above this value are clipped and not shown. May be set to “Inf” (infinity) to remove the maximum ceiling.

In addition, the plot may be right-clicked to bring up a context menu. From that menu, the plot may be saved as an image and the C_3 energy at a particular departure/arrival date combination may be computed.

Using the Arrival Velocity Plot Generator

The Arrival Velocity Plot Generator can be accessed from the toolbar via **Tools > Generate Arrival Velocity Plot**. When this menu item is selected, the following window appears (Figure 16).

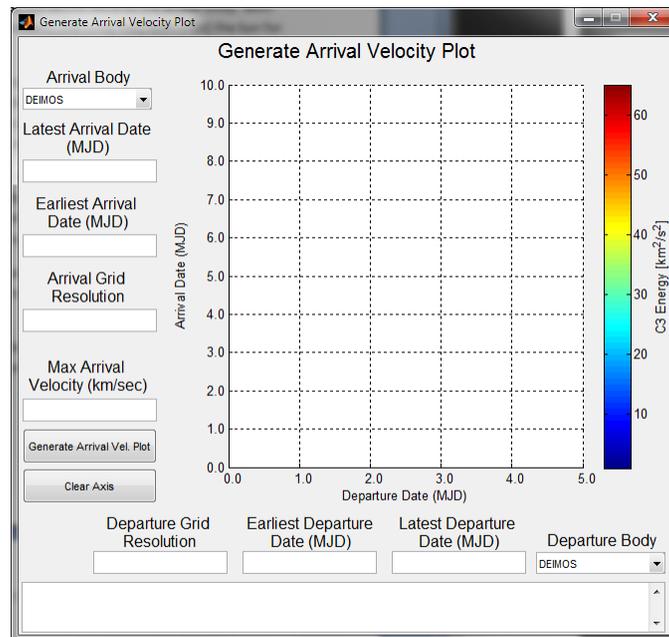


Figure 16: Arrival Velocity Plot Generator Window

The options shown in the window required to generate an arrival velocity energy plot are discussed below.

Arrival Body: The body to which you are going. May also be known as the “destination body.”

Departure Body: The body from which you are leaving. May not be the same as the arrival body. Both the departure and arrival bodies should have as their central body (the body they both orbit) the Sun for this release of the ToT.

Arrival Dates: The range of dates at which arrival to the Arrival Body is possible. The earliest arrival date must occur before the latest arrival date. Dates are entered in Modified Julian Day (MJD). There are numerous tools online to convert dates from the Gregorian calendar (month, day, year) to Modified Julian Day. MJD is also the system Orbiter uses for time keeping.

Departure Dates: The range of dates at which departure from the Departure Body is possible. The earliest departure date must occur before the latest departure date. Dates are entered in Modified Julian Day (MJD). There are numerous tools online to convert dates from the Gregorian calendar (month, day, year) to Modified Julian Day. MJD is also the system Orbiter uses for time keeping.

Grid Resolutions: The number of data points along the respective axis (departure/arrival) to compute. The total number of data points that have to be computed to generate the plot is the product of the two resolutions. Resolutions must be a positive integer.

Maximum Arrival Velocity: The maximum arrival velocity to plot on the graph. Any arrival velocity values above this value are clipped and not shown. May also be set to “Inf” (infinity) to remove the maximum ceiling.

In addition, the plot may be right-clicked to bring up a context menu. From that menu, the plot may be saved as an image and the arrival velocity at a particular departure/arrival date combination may be computed.

Using the Departure Analysis Tool

The Departure Analysis Tool is used for determining how to depart onto an interplanetary trajectory. It can be accessed from the toolbar via **Tools > Departure Analysis Tool**.

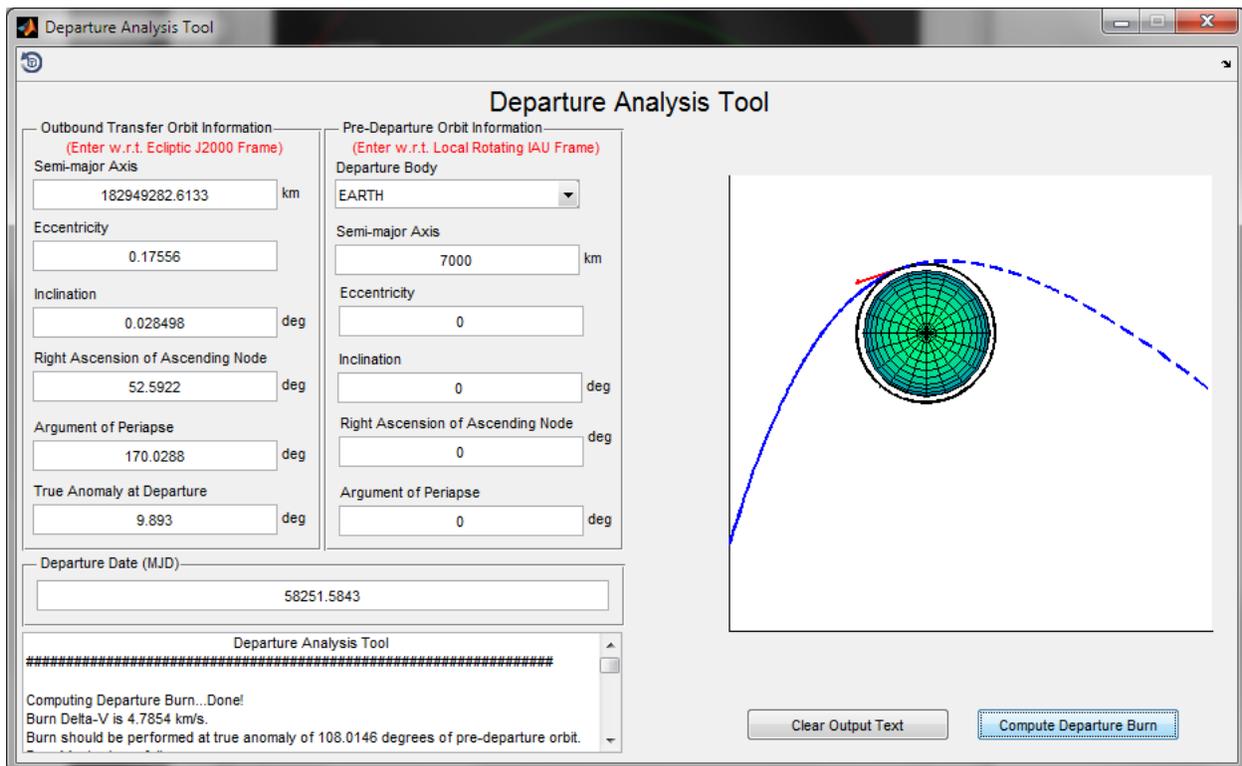


Figure 17: Departure Analysis Tool User Interface

Outbound Transfer Orbit Information: The outbound transfer orbit information describes the orbit of the first leg of the interplanetary trajectory, typically as computed by the Trajectory Optimization Tool. This is the trajectory around the main central body of the flight plan (for example, the Sun). If a trajectory has just been found using the Trajectory Optimization Tool, then information from the first leg of the trajectory will be automatically entered here. If you are entering this information manually, ensure that all values are with respect to the **J2000 frame (ecliptic plane reference)** and that you enter

consistent values for the orbital elements (eccentricity greater than 1, negative semi-major axis, for example).

All fields must be entered to compute a departure solution.

Pre-Departure Orbit Information: The pre-departure orbit information describes the orbit (and the body around which that orbit exists) the interplanetary spacecraft “starts” on. This may be a parking orbit around Earth or some such similar orbit. All fields must be entered to compute a departure solution. Values entered should be with respect to the **IAU's local reference frame** for that body and should be self-consistent (e.g. negative semi-major axis implies an eccentricity greater than 1).

The IAU local frames are rotating frames fixed to the surface of the body. *Typically*, the third component of the IAU local frame will be parallel to the body's spin axis, and the first and second components form a plane parallel to the body's equator. The departure date and time will fix the location of the IAU frame internally.

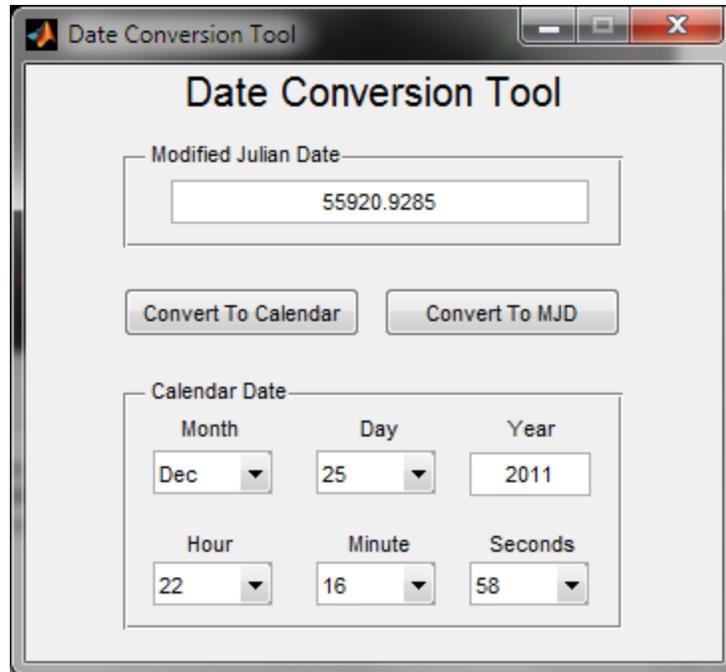
Departure Date (MJD): This is the date and time of departure from the Departure Body, entered in MJD (modified Julian date). This field must be entered to compute a departure solution.

Compute Departure Burn Button: Pushing this button checks the inputs for errors and determines two quantities: A) the burn vector and magnitude of the velocity change, and B) the true anomaly the burn should be executed at (were the burn to be perfectly impulsive). The computed departure solution is shown in the text output window, and the burn vector is given in both local IAU coordinates and VNC (velocity, normal, co-normal) coordinates. The latter are typically used for actually executing maneuvers.

Clear Output Text Button: Pushing this button resets the text in the output window.

Using the Date Conversion Tool

The date conversion tool is used to convert modified Julian dates to the Gregorian calendar and vice-versa. It can be accessed from the toolbar via **Tools > Date Conversion Tool**.



Modified Julian Date Field: Enter the modified Julian date to be converted to a calendar date here. Also shows a calendar date converted to MJD.

Calendar Date Fields: Enter the calendar date and time to convert to modified Julian date here. Also shows a modified Julian date converted to a calendar date.

Convert To Calendar Button: Takes the number entered in the MJD field and converts it to a calendar date. The result is shown in the Calendar Date fields.

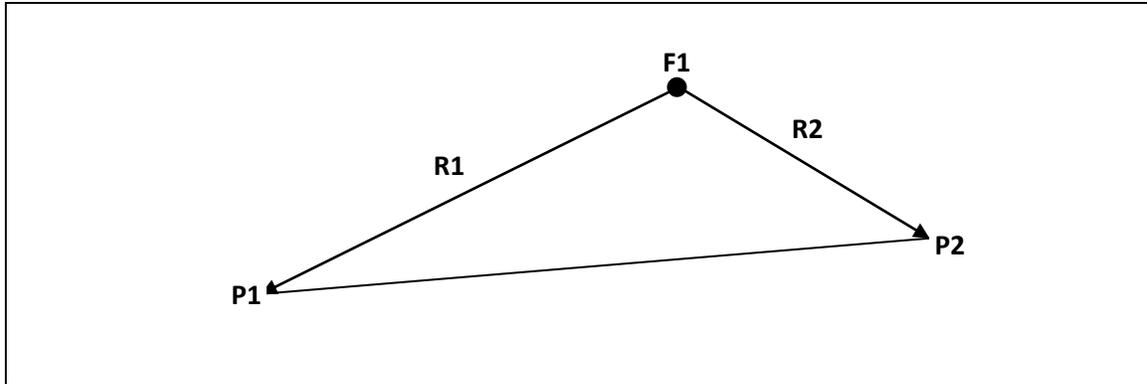
Convert To MJD Button: Takes the number entered in the Calendar Date fields and converts it to a modified Julian date. The result is shown in the Modified Julian Date field.

Theory of Operation

This section describes the theory behind a number of the features of the Trajectory Optimization Tool.

Lambert's Problem

Lambert's problem may be described as the problem of finding the arc that connects two points in three-dimensional space over a set time of flight. Lambert's problem is posed under the two-body approximation and thus always yields as a solution a Keplerian orbit. Put another way, consider the following plot of two points in space.



Given two points P1 and P2 with position vectors from F1 known as R1 and R2, respectively, and two different times T1 and T2, the goal in solving Lambert's problem is to find the solution to the differential equation

$$\ddot{\vec{r}} = -\mu \frac{\hat{r}}{r^2}$$

such that

$$\begin{aligned}\vec{r}(t_1) &= \vec{r}_1 \\ \vec{r}(t_2) &= \vec{r}_2\end{aligned}$$

Lambert's Equation formalizes the solution of this differential equation, but the actual discussion of that equation is outside the scope of this text.

There are a number of solution methods to Lambert's problem. All are numerical (except for special cases) and the simplest requires a basic root finding technique. There are actually two different solutions to each Lambert problem that results in an elliptic conic section as a solution, representing both sides of the conic section that connect P1 and P2. These arcs are described below in the Transfer Type discussion.

For more information on the Lambert problem and its solution, see any introductory textbook on astrodynamics. The key points in grasping the basics of the Lambert problem are the following:

- Given two points in space, their associated position vectors from some central point, and the time of flight between those two points, there exists a unique conic section connecting the points.
- In practical terms, the two points in space are typically the celestial bodies that one departs from and arrives at in a spacecraft. Of course, the first point P1 is the point in space one leaves the departure body and the point P2 is the point one arrives at the arrival body.
- The central point F1 is, in practical terms, the celestial body that the bodies at points P1 and P2 orbit about. In many cases, this is the Sun, but it could also be Jupiter or another planet with

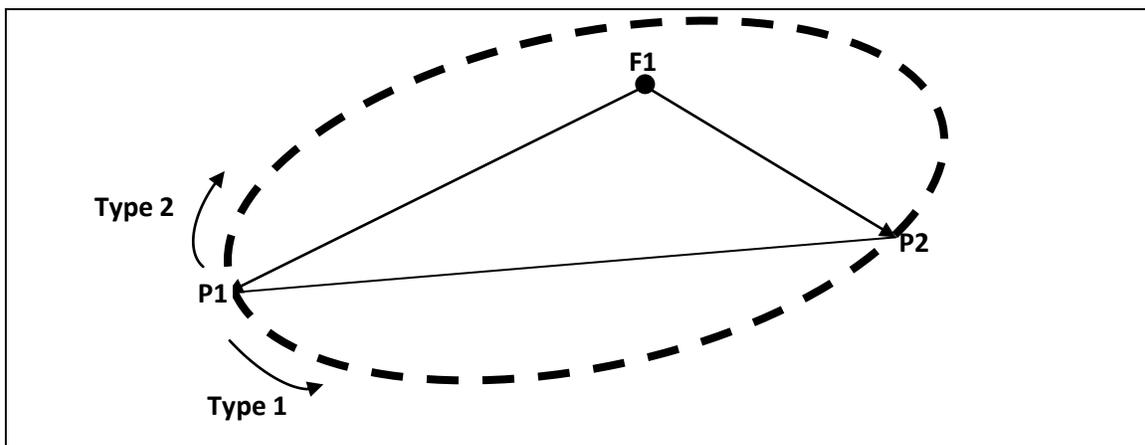
many moons. The body at the central point F1 must be the dominant gravitational influence along the entire trajectory being considered.

- For points that are connected with an elliptic conic section, there will be two different transfer orbits/arcs that connect the points.
- The solution to Lambert's problem is almost always carried out numerically except under special circumstances where an analytical solution may exist. These special cases are discussed in the scientific literature.

Lambert's problem is relevant for the Trajectory Optimization Tool because each segment of the flight plan is computed by solving a single Lambert's problem. Over the course of many waypoints and many iterations, you can imagine that a significant number of Lambert's problems are solved in order to find an optimum trajectory.

Transfer Types

As stated above, for an elliptic conic section that connects two points in space, there exists two solutions to Lambert's problem. Recall that these solutions to Lambert's problem give, in this case, elliptic transfer orbits about the central point/body. It turns out that one of these transfer orbits will transverse the small angle between P1 and P2, and the other will transverse the large angle between P1 and P2. The former transfer orbit is called a Type 1 transfer. The latter transfer orbit is called a Type 2 transfer. This is shown below.



The bold dashed line above represents the elliptic conic section that connects P1 and P2. We see that the lower transfer arc between P1 and P2 only covers about 90 degrees, and so it is Type 1. The upper transfer arc covers about 270 degrees, and so it is Type 2.

Genetic Algorithms

The trajectory optimization routine used in the Trajectory Optimization Tool is a genetic algorithm. As excerpted from the MATLAB documentation...

What Is the Genetic Algorithm?

The genetic algorithm is a method for solving both constrained and unconstrained

optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. You can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear.

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

- *Selection rules* select the individuals, called *parents*, that contribute to the population at the next generation.
- *Crossover rules* combine two parents to form children for the next generation.
- *Mutation rules* apply random changes to individual parents to form children.

The genetic algorithm differs from a classical, derivative-based, optimization algorithm in two main ways, as summarized in the following table.

Classical Algorithm	Genetic Algorithm
Generates a single point at each iteration. The sequence of points approaches an optimal solution.	Generates a population of points at each iteration. The best point in the population approaches an optimal solution.
Selects the next point in the sequence by a deterministic computation.	Selects the next population by computation which uses random number generators.

A warning concerning use of the genetic algorithm:



Warning! The genetic algorithm is effectively a heuristic brute-force technique and optimality of the returned solution is not guaranteed. The optimizer should be run at least three times on the same problem setup to determine if the optimized solution is in fact optimal.

Acknowledgements

The author would like to thank Dr. Martin Schweiger for providing [Orbiter Spaceflight Simulator](#), which has provided countless hours of enjoyment and learning over the years, as well as inspiration for the author's future career in astrodynamics.

The author would like to thank Orbiter-Forum.com member *Pipcard* for his donation of the new Trajectory Optimization Tool logo as seen in the **Help > About Trajectory Optimization Tool** menu. Thank you!

The author also would also like to thank the Jet Propulsion Laboratory for publically providing ephemerides data for open use.

Additional Reading

For those interested in the subject of space travel and some of theory used to develop this tool, the following sources are recommended:

1. Bate, Roger R., Donald D. Mueller, and Jerry E. White, *Fundamentals of Astrodynamics*, Dover, New York, 1971.
2. Jet Propulsion Laboratory. "Basics of Space Flight." Accessed 16 January 2011.
<<http://www2.jpl.nasa.gov/basics/index.php>>
3. Thomson, W. T., *Introduction to Space Dynamics*, Wiley, New York, 1961.
4. Gooding, R. H., "A Procedure for the Solution of Lambert's Orbital Boundary-Value Problem." *Celestial Mechanics and Dynamical Astronomy*. **48**: 145-165, 1990.