Kerbal Space Program Launch Window Calculator by GrizzlyBall

Quick and Dirty

I found the best way to use the results of the calculator in game is to wait until the desired launch date, put my ship into the specified parking orbit, and add a maneuver node with the Vi (injection delta V) I expect from the planner. Since the planner doesn't give you launch geometry, it's easiest just to adjust the maneuver node based on what you expect from the plot until you get an intercept with the given delta v. For advanced intercepts like Moho, you may have to do a lot of tweaking to make it look right and then add a second maneuver node to nail down the intercept. Use the Top-down orbital plot to help you get the sight picture right.

Step 1 – Set Your Transfer Inputs

Transfer Inputs		
Origin	Kerbin	
Parking Orbit [km]	90	*height above surface*
Destination	Duna	
Parking Orbit [km]	90	*height above surface*
Δv Budget [m/s]	4000	*max Δv displayed*

Choose your origin and destination. The pair must make sense and be a simple transfer. Examples of a simple transfer are Kerbin to Jool, Mun to Minmus, and Bop to Vall. Examples that **won't** work here are Kerbin to Ike and Mun to Duna.

Leave the parking orbits at 90 km unless you have the desire to change them. This ought to get you in the ballpark.

The delta v budget is essentially the max delta v shown on the Launch Window Plot – lowering this value will clean up the plot. Make sure this is above your expected minimum delta v required or the plot won't show you anything useful. You'll also get an error that says "The ETE duration must be greater than 0" as the plotting function fails.

Step 2 – Set Your Launch Window Inputs

Launch Window Inputs (Dates In Kerbal Time)			
start	*year*	1	**Window Progress:**
	day	1	finished!
duration	*years*	4	Calculate
	days	0	

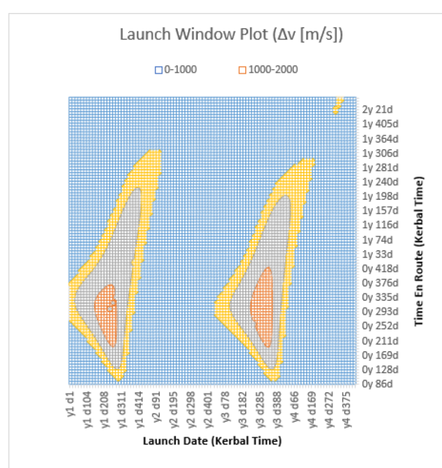
If you're searching for the best launch window at a certain point in your Kerbal campaign, set the start date to the current game date. Otherwise choose any start date and adjust as desired later on.

Set the duration based on the length of time you want to consider for planning your transfer. Longer launch window durations will potentially allow for lower fuel burns, but gives the plot lower fidelity. 2-4 years is a good place to start when planning a launch from Kerbin, and you can adjust from there.

You're all set! Hit "Calculate" and watch excel calculate 10,000 possibilities for your transfer window. Regardless of the duration, the program will break the launch window into 100 equally spaced launch times starting and ending at the dates you specify. For each launch time, it will calculate 100 different transfers, equally spaced in "Time En Route," or the time it takes to get to your destination.

Step 3 – Understanding the Output

The Launch Window Plot – You'll get a plot looking something like this. It's not interactive, just a big picture overview of transfer possibilities. In this case the best launch times (orange) are around year 1, day 250 and year 3, day 300. There's an interactive plot on the next excel tab, we'll get to that later.



Optimum Transfer – The code will automatically select the launch date and ETE of the best transfer within the bounds of your Launch Window Inputs. It is displayed here and will remain until you hit "Calculate" again. Initially the excel program will copy this data into the Plot Transfer Inputs box and plot the transfer for you.

Launch Window Results (next tab for interactive plot)			
Best Δv [m/s]		1666	
Launch	year	3	*(1 year = 426 days)*
Date	day	302.3	
ETE	year	0	
	day	293.3	

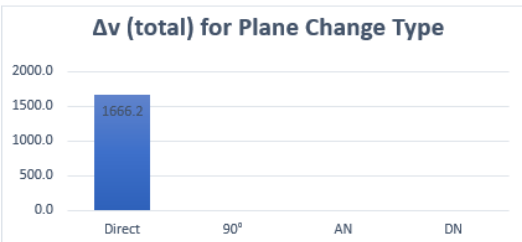
Step 3 – Understand the Output (continued)

Single Transfer Plot – When you first calculate the launch window, the optimum transfer will automatically be plotted for you. Afterwards you are free to tweak these values to see what any other transfer would look like. Be careful though – the excel code doesn't like hyperbolic orbits very much and will sometimes crash if you put ridiculous values in here. If you run into problems just CTRL-ALT-DEL and try again.

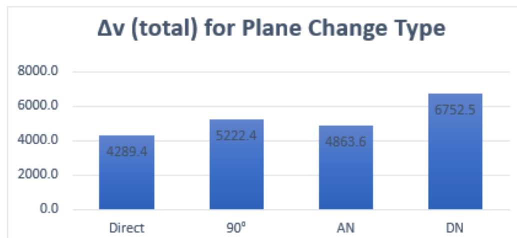
Plot Transfer Inputs (Dates In Kerbal Time)			
date	*year*	3	*(1 year = 426 days)*
	day	302.3	
ETE	*years*	0	Plot Orbit
	days	293.3	

Delta v vs. Plane Change Type – In this simple Duna example (shown on the left), the target planet lies so close to the orbital plane of Kerbin that the excel program ignores plane change possibilities and only shows you a "direct" transfer. Take a look at a Kerbin-to-Moho transfer (shown on the right).

In this example, we consider changing from the Kerbin orbital plane to the Moho orbital plane 4 different ways: direct (with injection maneuver in Kerbin parking orbit), 90° from intercept (typically halfway through the transfer), at the Ascending Node (AN), and at the Descending Node (DN). In this case we can see that the direct transfer in most efficient, but you may find it easier to plan a plane change at one of the nodes.

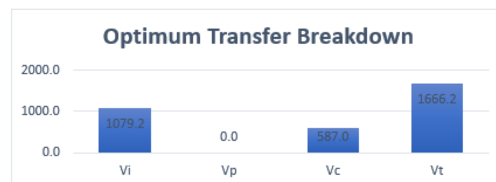


Duna example from previous diagrams

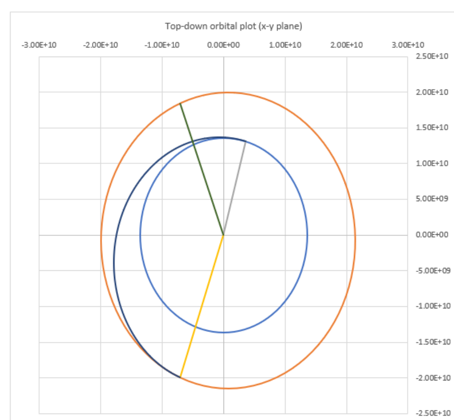


A New Moho example showing different plane change options

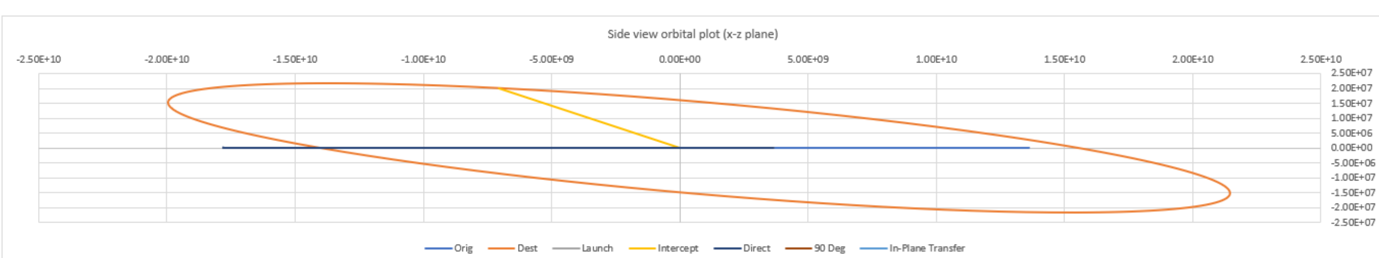
Optimum Transfer Breakdown – The transfer with the lowest delta v is further broken down into its components. Vi – injection delta v (assumes circular orbit at origin parking orbit of specified height); Vp – plane change delta v; Vc – capture delta v (similar assumptions as Vi); and Vt – total delta v.



Top-Down and Side-View Orbital Plots – very useful in helping to visually grasp what's going on. The key to understanding the orbital plots is to remember that it's for a single transfer whose inputs are in the "Plot Transfer Inputs" boxes (the ones in yellow). The line colors can change on excel's whim, but here the grey line is pointing to our origin at launch, green is destination at launch, and yellow is destination at intercept. The nodes would be normally be displayed but are suppressed in this example because Duna is in a very similar inclination to Kerbin.



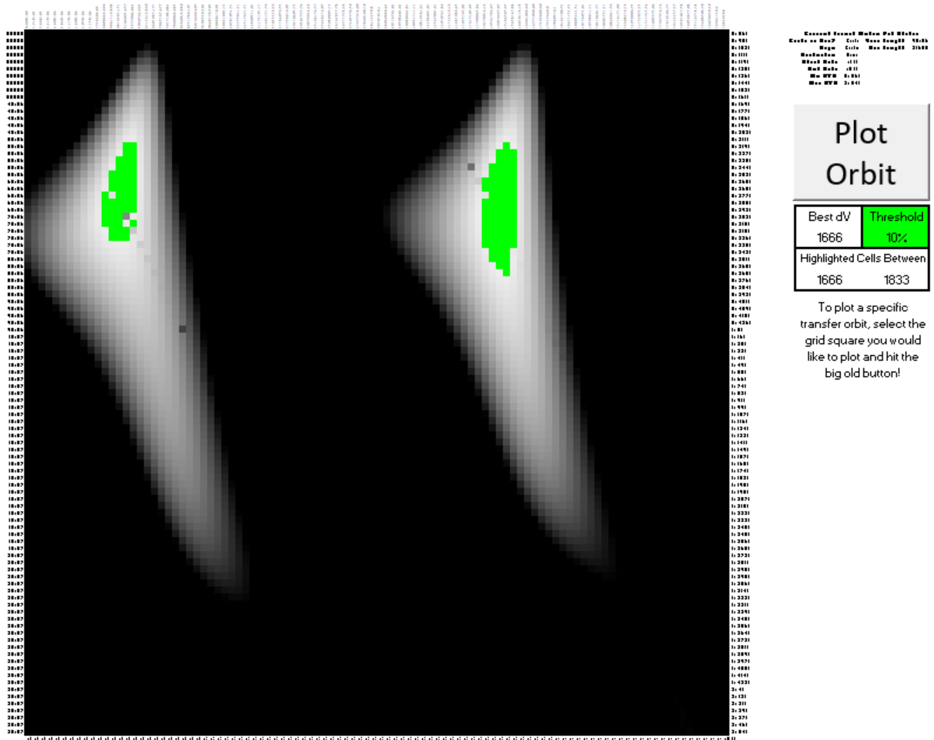
Note: The plot scale adjusts automatically and can sometimes be deceiving. Duna looks to be relatively inclined in the side-view but this is just induced by the scale on the z-axis (several orders of magnitude smaller than the x-axis).



The Interactive Launch Window Plot

Once you get the hang of the calculator, you can head on over to the next tab in excel. Here you will find an interactive, black-and-white version of the plot we previously discussed. Besides being upside down, it has another neat feature. You can click on any cell (lighter cells are lower delta v) and hit “Plot Orbit” to see what that transfer looks like on the plots from the previous tab. It’ll take you there automatically but you can return and re-plot orbits as many times as you like.

Note: Changing the green “Threshold” number can help you identify the best transfers.



Notes

Additional Launch Window Options – you can further constrain your plot by bounding your ETE. This is particular useful for transfers with overwhelming options, like Moho to Eeloo. Not that you would ever try that in game, right?

Real-world plotting – Located on the first tabs in excel, it’s “real” in the sense that the planets are real, but the launch times are just representative of an arbitrary point in time. I didn’t spend any time on the non-patched-conics side of this to flesh out real-life planetary locations over long periods of time. The start date (year 1 day 1) is coincident with the mean anomaly of the planets as listed on Wikipedia.

Bugs – You can see the bugs in the random dark cells in the interactive plot. These are intercepts that have at least partially failed. Usually its due to a problem in the trig functions in the iterative solvers (the Gauss method function is the primary iterative solver). Most of the time the code will skip that particular transfer and move on, but I’ve had crashes occasionally. Forgive me for the crappy code, I programmed this thing on a boat without internet access for my excel help functions!

Expanding the Code – My hope is that somebody will find this useful as a starting point for their own project. There’s lots that could be done, from fixing the hyperbolic orbit functionality to planning fly-bys and what-not. Please feel free to edit, expand, and/or use code snippets for your own. Credit me if you find it useful!