Optimum Interplanetary Trajectory Software 2017

Definition Manual (DRAFT)

Adam Hibberd

Contents

Optimum Interplanetary Trajectory Software by Adam Hibberd

1) Interplanetary Trajectory

1.1) Overview

An interplanetary trajectory (IT) connects a specified series of solar system objects (SSO's), be they planets, asteroids, comets or otherwise in a given order at a series of times. (We shall find that adopting a few assumptions, a completely defined IT is dependent only on the times at the SSOs.) For a large part the body or probe P journeying on this trajectory is under the dominant influence of the Sun's gravity. However for a series of relatively short time periods, as P encounters each SSO in turn, the overriding influence switches to become that of the SSO's gravity. Thus, this process of P's exploration through the Solar System takes the form of long periods of relative inactivity, lasting months to years, interspersed with spikes of activity, lasting minutes to hours as P flies past each SSO. Naturally one must include the effect of the probe's own thrust in these calculations. In order to achieve thrust the probe needs fuel which in turn has implications on its mass. This brings us to the subject of the design of the probe. An SSO is of class `Body`.

1.2) Probe Design

The probe will inevitably need to be launched from Earth as a payload to a launch vehicle of some kind. By this manner it will be boosted into space in order to eventually escape Earth's gravitational influence. One can calculate the 'escape velocity' on Earth to be in the region of 11km/s which would be the minimum required speed. The launcher in question will have limitations as far as maximum capable payload mass to orbit is concerned. Consequently, this is the key driver for designing the probe to have as low a mass as possible and therefore as little fuel mass as possible. This then is the motivation for designing an Optimum Interplanetary Trajectory (OIT), i.e. one that most efficiently visits all the SSO's required, efficient in the sense of minimum fuel and therefore as we shall see, ΔV (pronounced delta vee).

1.3) ΔV

So how does one take account of the thrust of P? As stated above minimizing fuel mass is crucial in the process of designing an IT, i.e. one must find an OIT. The effect of P switching its thrusters on for a time interval Δt is to apply a force and therefore acceleration in a prescribed direction. Thus an increase in velocity, ΔV, in the prescribed direction is generated over the time period Δt. In fact for most probe design applications, this time period can be supposed as negligible in comparison with the overall IT flight time and so the thrust can be assumed as impulsive, with an instantaneous change in velocity of magnitude ΔV. From the equation for orbital energy it can be derived that the maximum change in energy is achieved by applying this ΔV in the direction of P's current velocity vector. It should be noted that several such impulses can be delivered at different points along the IT, each with a different ΔV, i.e. $\Delta V_{i.}$

1.4) Dynamics at Home Planet

The probe has to be launched from the home planet (normally Earth!) and inserted into an escape orbit relative to Earth, or hyperbolic orbit. The minimum escape velocity of 11km/s at Earth's surface will yield an overall velocity at infinity, or 'hyperbolic excess velocity', $\mathbf{V}_\infty$, of exactly 0km/s. A 'sphere of influence', or an outer limit whose origin is at the centre of the SSO, can be estimated beyond which the dominant influence switches from that of the SSO to that of the Sun. At this point P can be deemed to have escaped the SSO (various strategies can be adopted to derive the radius of this sphere of influence). The difference in velocity achieved by P at this distance when compared with the hyperbolic excess velocity it would reach at infinity, $\mathbf{V}_\infty$ (or for OITS,**VD)** can be discounted as negligible. (Note the use of a bold font to indicate $\mathbf{V}_\infty$ **{or VD}** is a vector.) This hyperbolic excess is naturally relative to the SSO. In order to determine the velocity relative to the Sun for the OIT calculations, one must add on the velocity of the SSO relative to the Sun, $\mathbf{V}_{SSO}$. Thus the departure velocity is given by $\mathbf{V}_{DEP} = \mathbf{V}_{SSO} + \mathbf{VD}$.

1.5) Assumptions

1) We can therefore claim that our OIT consist of periods of time where the overwhelming influence is either that of the Sun or an SSO, not both at the same time.
2) The OIT is interspersed with several instantaneous applications of ΔV in the direction of its current motion.
3) It is further assumed that ΔV is applied at the moment of closest approach to each SSO in turn (at the points of 'periapsis') as it can be proved, again from the energy equation, that this results in the maximum increase in energy.

1.6) Theory

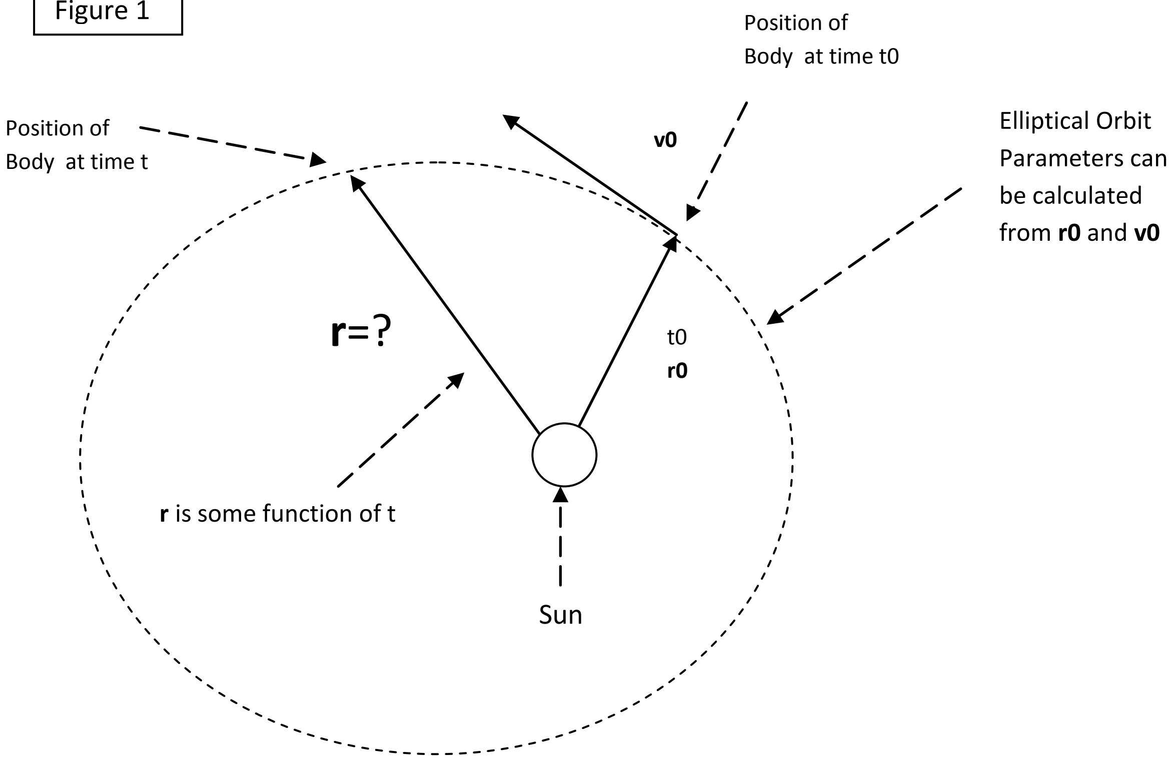
1.6.1) Calculating Ephemeris of a Body B from Time

Given a time t, one can compute the state or ephemeris, i.e. the position **r** and velocity **v** of a Body B. If one assumes the only force applied to B is the gravity of a single object such as the sun, there are three possible methods for calculating the ephemeris:

1) Numerical integration of the trajectory forwards in time from a known Epoch time, t0, and starting position **r0** and velocity **v0,** until the time t is reached.
2) Deriving and employing an equation that has as input time and outputs the ephemeris. Such an equation would require the orbital elements (or orbital parameters) of B which are constants by assumption 1.5.1 (The orbit of a body in such a gravitational field is fixed) .
3) Using the NASA Spice toolkit, **which does not require assumption 1.5.1**.

As indicated in 2 above the orbit of B is unchanged in orientation, plane and shape when subject to an inverse square law gravitational field, for example generated by the sun. The position of B at time t can therefore be narrowed down to somewhere along this orbit. The shape (a 'conic section') can be an ellipse, i.e. a closed orbit, or either a parabolic or a hyperbolic orbit, which are both escape orbits. Figure 1 shows the situation with an ellipse. In travelling from known t0, **r0** and **v0**, the body describes the arc of an ellipse. The orbital parameters needed to completely determine an orbit are (a, e, I , Ω, ω) and are respectively semi-major axis (a), eccentricity (e), inclination (I), longitude of ascending node (Ω) and argument of perigee (ω). These parameters (a, e, I, Ω, ω) are explicit functions of **r0** and **v0** (OITS employs the function `calculate_orbit_from_ephem`); however having obtained them it is not possible to derive explicit functions of **r** and **v** in terms of t and these known parameters. Numerical methods have to be employed to solve the problem using an initial guess, a repeating iteration and convergence to a solution**. Thus for either of options 1 and 2 above, numerical techniques must ultimately be employed.** For option 2 it is customary to have stored the complete set of fixed orbital parameters for the B (a, e, I, Ω, ω) from which any value of **r** and **v** can be calculated for a given t via an iteration technique (refer `calculate_orbit_from_ephem`).

Figure 1



1.6.2) Connecting Just Two SSO's With a Transfer Orbit.

Given an SSO then as we saw in Section 1.6.1, we can fully calculate its position and velocity with just the time t. Now given two SSO's, we may wish to determine a transfer orbit connecting them up. When we talk about connecting up two SSO's we mean a travelling interplanetary probe, P, will be located on departure at SSOD's position **rd** at time td and later be located on arrival at SSOA's position **ra** at time tar; where we know **rd** and **ra** and they are only dependent on td and tar respectively. We assume the only gravitational force between SSOD and SSOA is the inverse square force from the Sun and so the connecting trajectory will be a conic section. The situation for P is pictured in Figures 2 & 3 where the solution trajectory to such a problem is not unique, in fact there are two possible routes the 'short way' and 'long way'.

A technique to solve such a problem using a 'Universal Variable Formulation' is elaborated in 'Fundamentals Of Astrodynamics' by Roger R. Bate, Donald D. Mueller and Jerry E. White. It involves a numerical iteration technique to converge to the 2 solution transfer orbits. This is implemented in routine '`Calculate_transfer`'. From the solution orbits so derived, the Heliocentric Departure Velocity $\mathbf{V}_{\mathbf{DEP}}$ and Arrival Velocity $\mathbf{V}_{\mathbf{ARR}}$ can easily be found and in fact there will be two of each $\mathbf{V}_{\mathbf{DEP}}$(j) , j=1,2 and $\mathbf{V}_{\mathbf{ARR}}$(j), j=1,2 where j = 1 for the short way and j=2 for the long way.

Figure 2

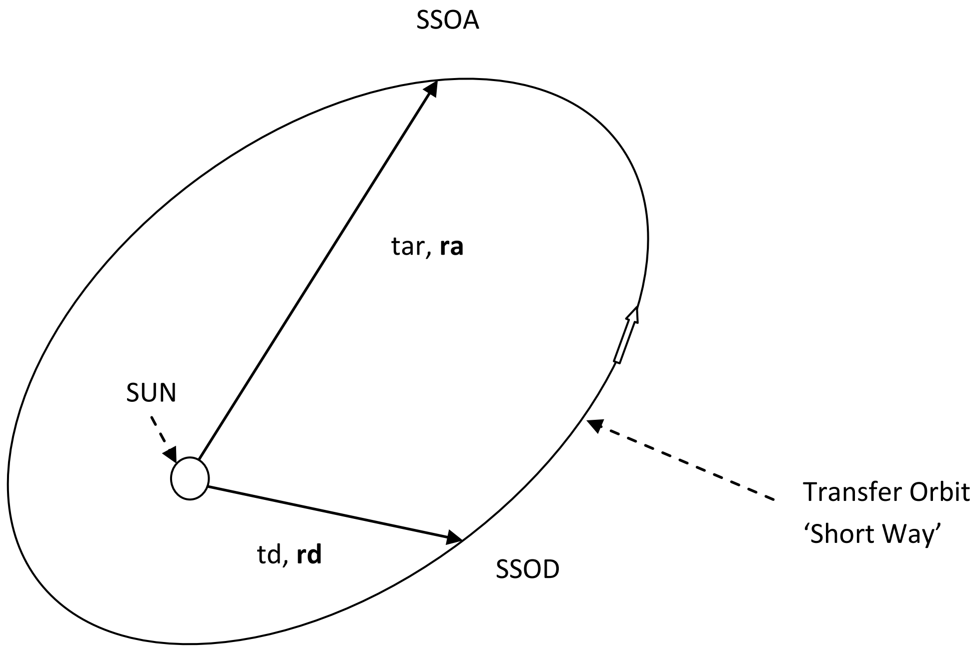
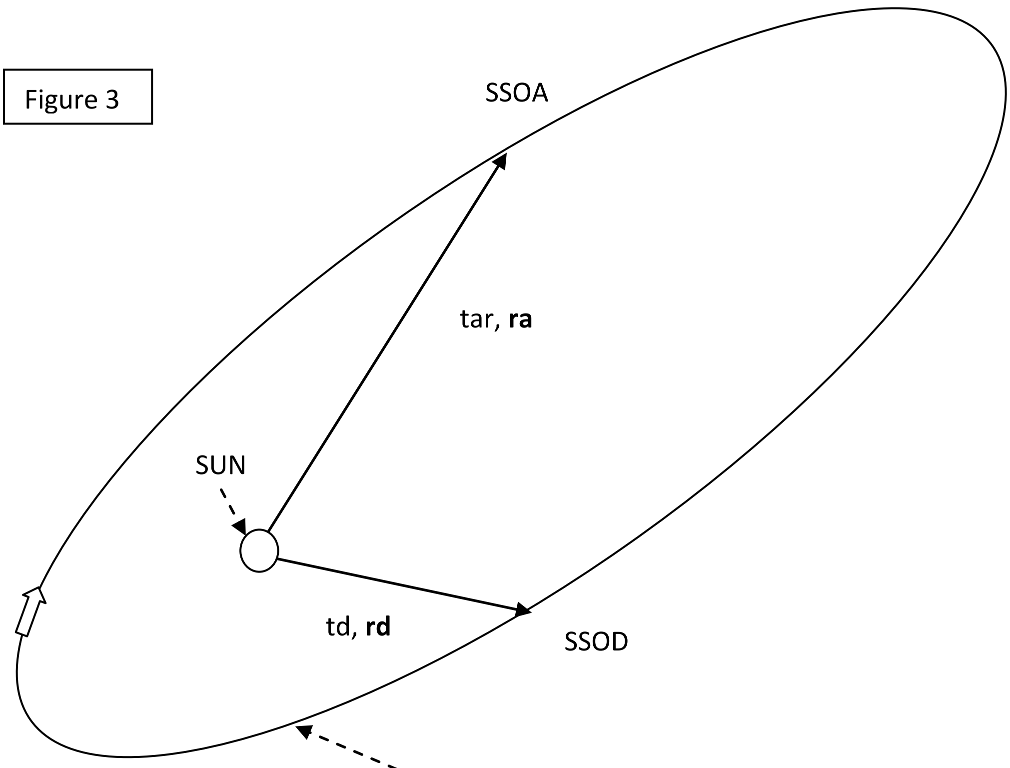


Figure 3



1.6.3) Calculation of ΔV For Connecting Up Just Two SSO's at Given Times

We have seen in the preceding section that connecting two SSO's given two times td and tar results in two possible transfer orbits (refer class `Transfer_orbit`) as follows:

Which Way------->		Short Way	Long Way
Departure SSO	Time	td	td
	Position (Heliocentric)	**rd**	**rd**
	Velocity (Heliocentric)	$\mathbf{V_{SSOD}}$	$\mathbf{V_{SSOD}}$
Transfer Orbit At Deparutre	Time	td	td
	Position (Heliocentric)	**rd**	**rd**
	Velocity (Heliocentric)	$\mathbf{V_{DEP}}(1)$	$\mathbf{V_{DEP}}(2)$
Arrival SSO	Time	tar	tar
	Position (Heliocentric)	**ra**	**ra**
	Velocity (Heliocentric)	$\mathbf{V_{SSOA}}$	$\mathbf{V_{SSOA}}$
Transfer Orbit at Arrival	Time	tar	tar
	Position (Heliocentric)	**ra**	**ra**
	Velocity (Heliocentric)	$\mathbf{V_{ARR}}(1)$	$\mathbf{V_{ARR}}(2)$

Having acquired two possible transfer orbits connecting SSOD to SSOA at times td and tar respectively, it is our purpose to inquire what effort would be needed at SSOD to achieve these two transfers and indeed which of these two possibilities is optimum from this point of view. If we know that our probe originated on SSOD then this immediately tells us that it is in fact already travelling with a considerable heliocentric velocity $\mathbf{V_{SSOD}}$, i.e. it is by its very origin orbiting around the Sun with the SSOD it belongs to. Now the change in velocity **VD** that it must apply in order to acquire the new velocity $\mathbf{V_{DEP}}$ and so adopt a transfer orbit which will ultimately reach SSOA at time tar, is simply given by:

$$\boldsymbol{VD} = \boldsymbol{V_{DEP}} - \boldsymbol{V_{SSOD}}$$

In fact there are two such transfer orbits and so two possible values for **VD**, short way and long way:

$$\boldsymbol{VD}(j) = \boldsymbol{V_{DEP}}(j) - \boldsymbol{V_{SSOD}} \qquad j = 1,2$$

If we allow our velocity impulse to be in any prescribed direction without penalty then it follows that there are two magnitudes of **VD** which we shall call ΔVD(j) , j=1,2 given by:

$$\Delta VD(j) = |\mathbf{VD}(\mathrm{j})| \qquad j = 1,2$$

and in fact the smaller one will be the ΔV, and for that matter the transfer orbit, which we seek:

$$\Delta V = min\{ \;\; \Delta VD(j) \quad : \quad j = 1,2 \quad \}$$

1.6.4) Connecting More Than Two SSO's With an Interplanetary Trajectory (Class `Nbody_Trajectory`)

Let us now deal with the more general case where we have several (possibly more than two) SSO's, in fact let us say we have an integer 'Nbody' of them, each of class *Body* and each associated with a time. We can introduce the notion of a set (or array) of such bodies, given by *Body_Set(j), j=1…Nbody* and the members of this set each have an ephemeris (comprising time t, position **r** and velocity **v**) given by *Body_Set(j).ephemt*. (NB the name *ephemt* is used to indicate complete dependence of ephemeris on time t.) Now *Body_Set(j)* is connected to *Body_Set(j+1)* by a transfer orbit thus there are 'Ntrans' transfer orbits, each of class *Transfer_orbit.* Again we introduce a set or array of such items called *Trans_Set(j'), j'=1…Ntrans.* Not much effort is required to deduce that:

$$Ntrans = Nbody - 1$$

By the nature of *Transfer_orbit* as we have discussed, there are two possible solutions, short way and long way, which for ease of programming we shall define as two separate bodies: *transfer_body(1)* and *transfer_body(2)*.'*transfer_body'* here is referring to our interplanetary probe in two possible incarnations: journeying short way (1) or long way (2).

The situation is illustrated schematically in Figure 4 for the case of *Nbody=5*.

Note that *Body_Set(j)* refers to the jth member of the set of SSO's which make up the *Nbody_Trajectory*. *Trans_Set(j')* refers to the j'th member of the set of transfers, i.e. that member which connects SSO j=j' to SSO j=j'+1.

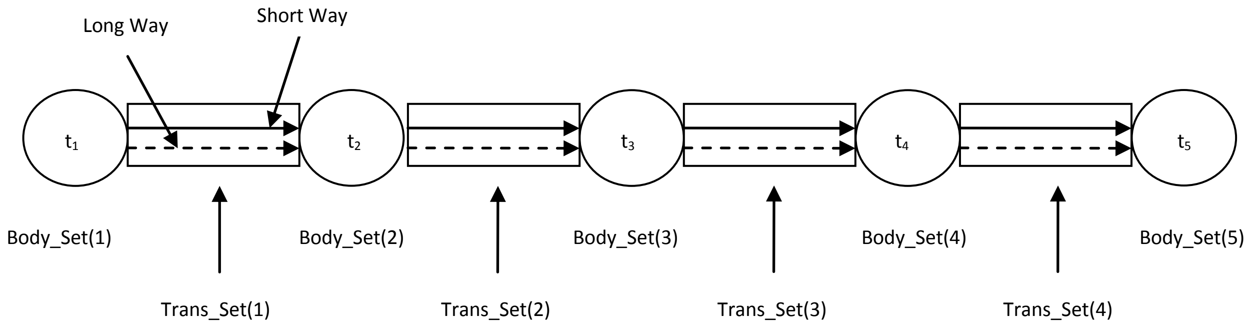


Figure 4 : `Nbody Trajectory`

With *Nbody_Trajectory* now defined we observe that each member of *Trans_Set* has two possible solutions, long way and short way thus all in all, given *Nbody* SSO's, each associated with a time t_j , and *Ntrans (= Nbody-1)* connections, there are *NP* possible permutations of interplanetary trajectory (IT) where:

$$NP = 2^{Ntrans}$$

Trans_Set(j).transfer_body(1).ephem0 is the ephemeris of the probe at *Body_Set(j)* in incarnation k=1, and *Trans_Set(j).transfer_body(2).ephem0* the ephemeris of the probe at *Body_Set(j)* in incarnation k=2. Similarly, *Trans_Set(j).transfer_body(1).ephemt* is the ephemeris of the probe at *Body_Set(j+1)*in incarnation k=1, and *Trans_Set(j).transfer_body(2).ephemt* the ephemeris of the probe at *Body_Set(j+1)*in incarnation k=2.

1.6.5) Calculation of ΔV For Connecting Up Two or More SSO's at Given Times

Let us now focus on a random SSO in our *Nbody_Trajectory* formulated in the previous section. Thus we are referring to the jth member *Body_Set(j)*.This is connected to *Body_Set(j-1)* and *Body_Set(j+1)*via *Transfer_orbit(j-1)* and *Transfer_orbit(j)* respectively. The situation is summarised in the table below:

Nbody_Trajectory	IT								
SSO (Body)	*Body_Set(j-1)*				*Body_Set(j)*				*Body_Set(j+1)*
Transfer_orbit		*Trans_Set(j-1)*				*Trans_Set(j)*			
Transfer Body (Body)		*transfer_body(k=1)*		*transfer_body(k=2)*		*transfer_body(k=1)*		*transfer_body(k=2)*	
Heliocentric Velocity	*Body_Set(j-1).ephemt.v*	*transfer_body(1).ephemt.v*		*transfer_body(2).ephemt.v*	*Body_Set(j).ephemt.v*	*transfer_body(1).ephem0.v*		*transfer_body(2).ephem0.v*	*Body_Set(j+1).ephemt.v*
Velocity rel. to SSO		**VA**(j,k=1)		**VA**(j,k=2)		**VD**(j,k=1)		**VD**(j,k=2)	

Thus there are two possible arrival velocities at *Body_Set(j)*, one for each *transfer_body(k=1,2)* belonging to *Trans_Set(j-1)* as follows:

$$\textit{IT.VA(j,k)} \quad = \quad \textit{IT.Trans\_Set(j-1).transfer\_body(k) .ephemt.v} \quad - \quad \textit{IT.Body\_Set(j).ephemt.v} \quad k=1,2$$

And there are two possible departure velocities at *Body_Set(j)*, one for each *transfer_body(k=1,2)* belonging to *Trans_Set(j)* as follows:

$$\textit{IT.VD(j,k)} \quad = \quad \textit{IT.Trans\_Set(j).transfer\_body(k) .ephem0.v} \quad - \quad \textit{IT.Body\_Set(j).ephemt.v} \quad k=1,2$$

If we now further suppose that the SSO's have no gravitational influence on the probe, i.e. we ignore encounter dynamics of the probe with each SSO, the ΔV at each SSO is the magnitude (or 'norm') of the vector difference of **VA** with **VD**.

The sum of all these ΔV's over all the SSO's make the total '*DeltaV*' for the IT. However there is more than one such *DeltaV* for the IT, each with a different combination of short ways and long ways.

Indeed for each SSO, there are four ΔV's. Refer to Figure 5

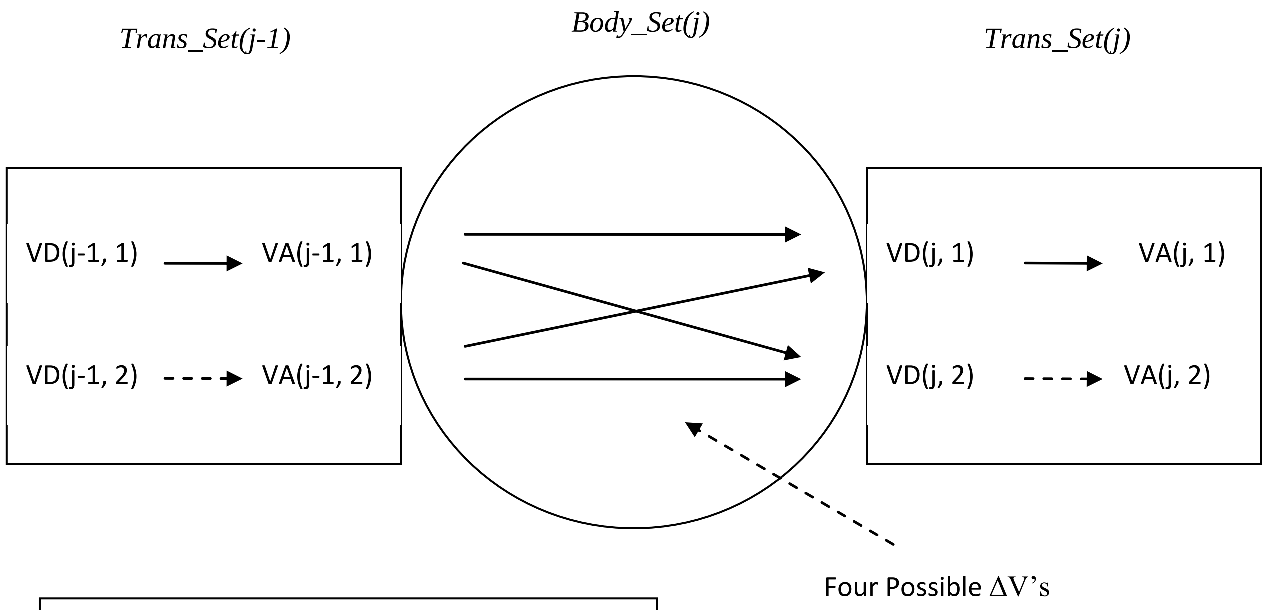


Figure 5 : Zoom in on `Nbody Trajectory`

This is in accord with previously stated i.e. that there are $NP=2^{Ntrans}$ possible ways to travel in an IT with *Nbody* SSO's where *Ntrans = NBody-1*. Each permutation *i=1...NP* can be appropriately accounted for by introducing the 2-dimensional constant array *perm(NP,Ntrans)*.

Let us consider the situation where *Nbody=5*, so *Ntrans=4*, so *NP=16*. The contents of array *perm* are shown below:

perm(i,j)	j=1	j=2	j=3	j=4
i=1	1	1	1	1
i=2	2	1	1	1
i=3	1	2	1	1
i=4	1	1	2	1
i=5	1	1	1	2
i=6	2	2	1	1
i=7	1	2	2	1
i=8	1	1	2	2
i=9	2	2	2	1
i=10	1	2	2	2
i=11	2	1	2	1
i=12	1	2	1	2
i=13	2	1	2	2
i=14	2	2	1	2
i=15	2	1	1	2
i=16	2	2	2	2

The ΔV for each SSO, j and permutation i, is assigned the variable *dV(i,j)* where :

$$dV(i,j)=norm(\boldsymbol{VD(j,perm(i,j))-VA(j,perm(i,j-1))})$$

and the sum of these over j makes the ith possible permutation of *DeltaV(i)*:

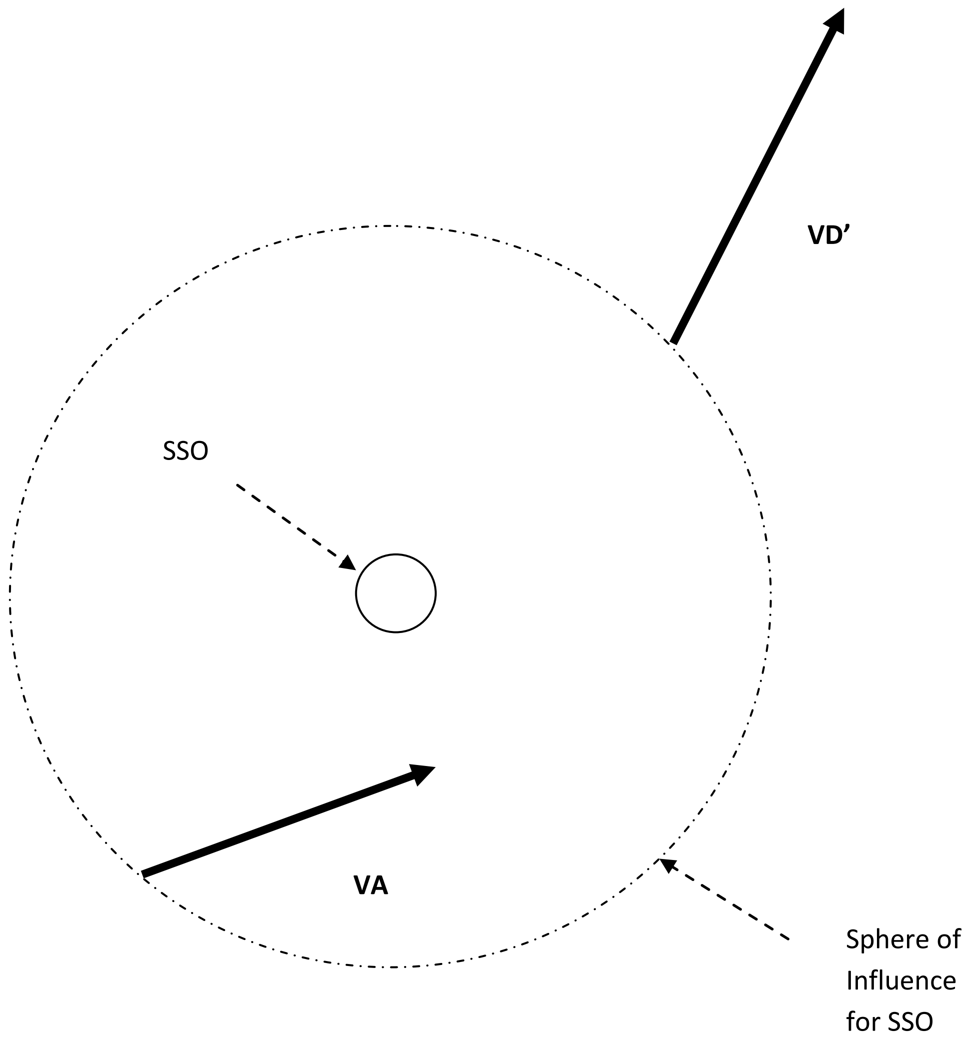
$$DeltaV(i) = \sum dV(i,j) \qquad j=1...Ntrans$$

And the minimum of these *DeltaV*'s gives the *BestDeltaV*, thus:

$$BestDeltaV = min\{ \;\; DeltaV(i) \;\; : \;\; i = 1 \dots NP \;\; \}$$

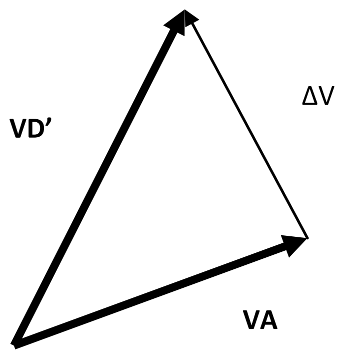
1.6.6) Incorporating Encounter Dynamics (in 2D) – (Class *Nbody_Trajectory_With_Encounters*)

Figure 6



Let us clarify the situation we have reached with Figure 6. The encounter dynamics with each SSO have up to now been ignored and the ΔV at each SSO has been simply the norm of the vector difference between **VD'** (the reason for the introduction of the apostrophe will be revealed later) and **VA** as shown in Figure 7:

Figure 7



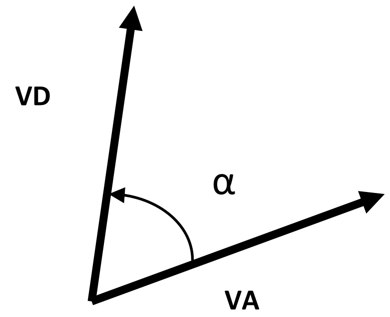
Now we are presented with the problem of incorporating the encounter dynamics with the SSO.

We have derived the approach velocity as **VA**. As far as the SSO is concerned, the probe is approaching it from infinity, or as we have seen from a distance equal to the radius of the 'Sphere of Influence'. This defines the trajectory of the probe with respect to the SSO as a hyperbola with excess speed equal to the magnitude of **VA**. Now without any other force, the effect of the gravitational influence of SSO would be to change the direction of **VA** to give **VD** (without the apostrophe) whose magnitude would be equal to that of **VA**:

$$|\boldsymbol{VD}| = |\boldsymbol{VA}|$$

This is shown in Figure 8:

Figure 8



Such an encounter is illustrated in Figure 9.

The following equations apply to such an encounter:

$$\alpha + \pi = 2\theta$$

$$cos\theta = -\frac{1}{e}$$

$$e = \frac{r_{PER}}{\mu} VA^2 + 1$$

$$V_{PER}{}^2 = 2\left(\frac{\mu}{r_{PER}}\right) + VA^2$$

However with the vectors of **VA** and **VD'** for our probe at the encounter SSO, it is not generally the case that |**VD'**| = |**VA**|. We thus need to accommodate the more likely circumstance where |**VD'**| ≠ |**VA**| - refer Figure 10. This necessitates a change in kinetic energy along the encounter trajectory in order to increase (or decrease) the escape speed of the probe with respect to the SSO. It can be shown that such a change in energy is most efficiently delivered at the point of periapsis with respect to the SSO and is best manifested by a kick in velocity, ΔV, in the direction of the velocity vector (which will be perpendicular to the radius vector at periapsis). This ΔV so delivered now replaces the ΔV derived in the previous section. Refer to Figure 11 for our modified trajectory.

Figure 9

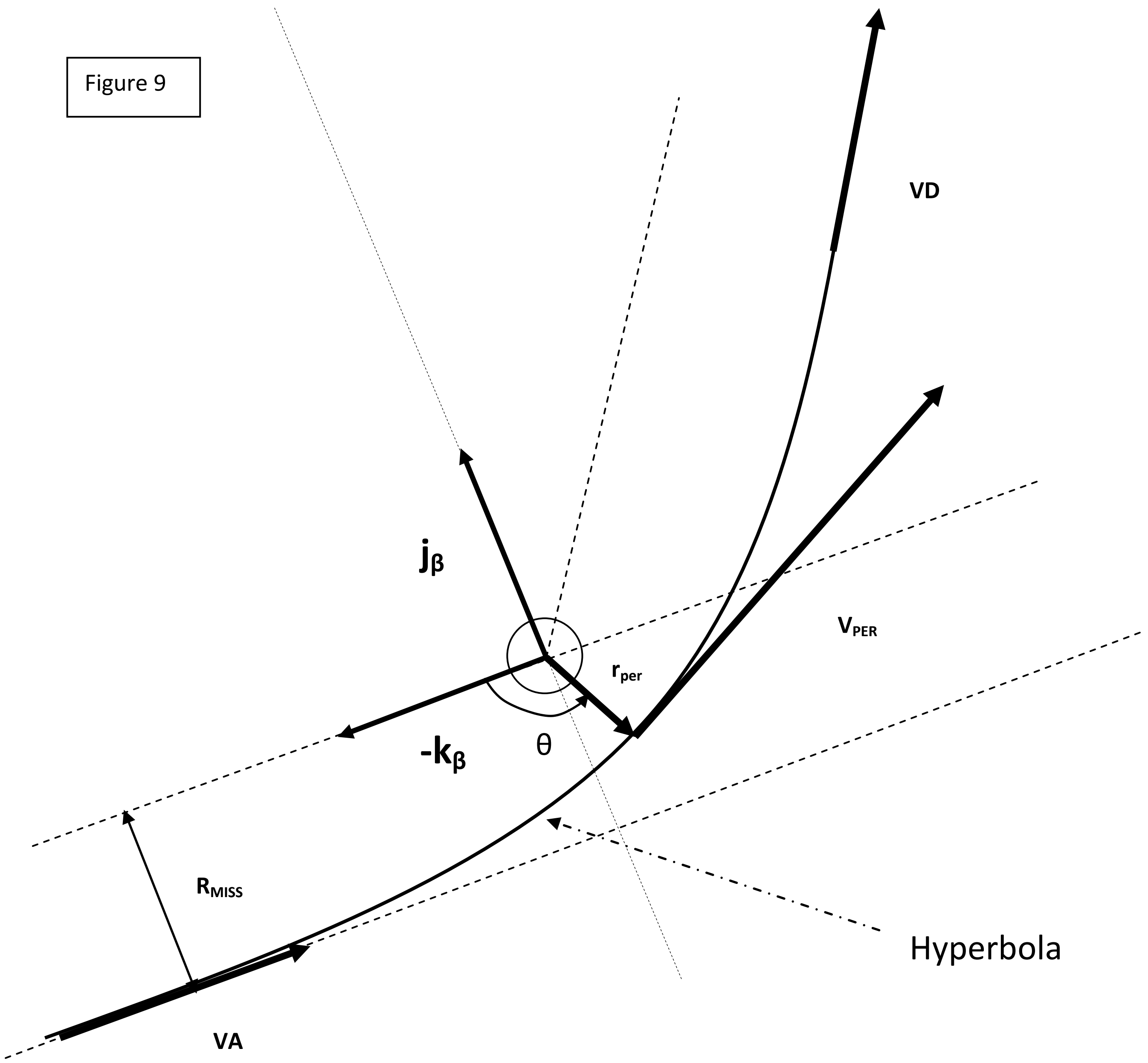


Figure 10

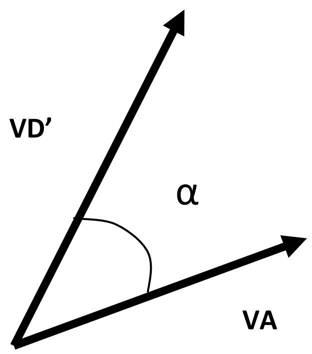
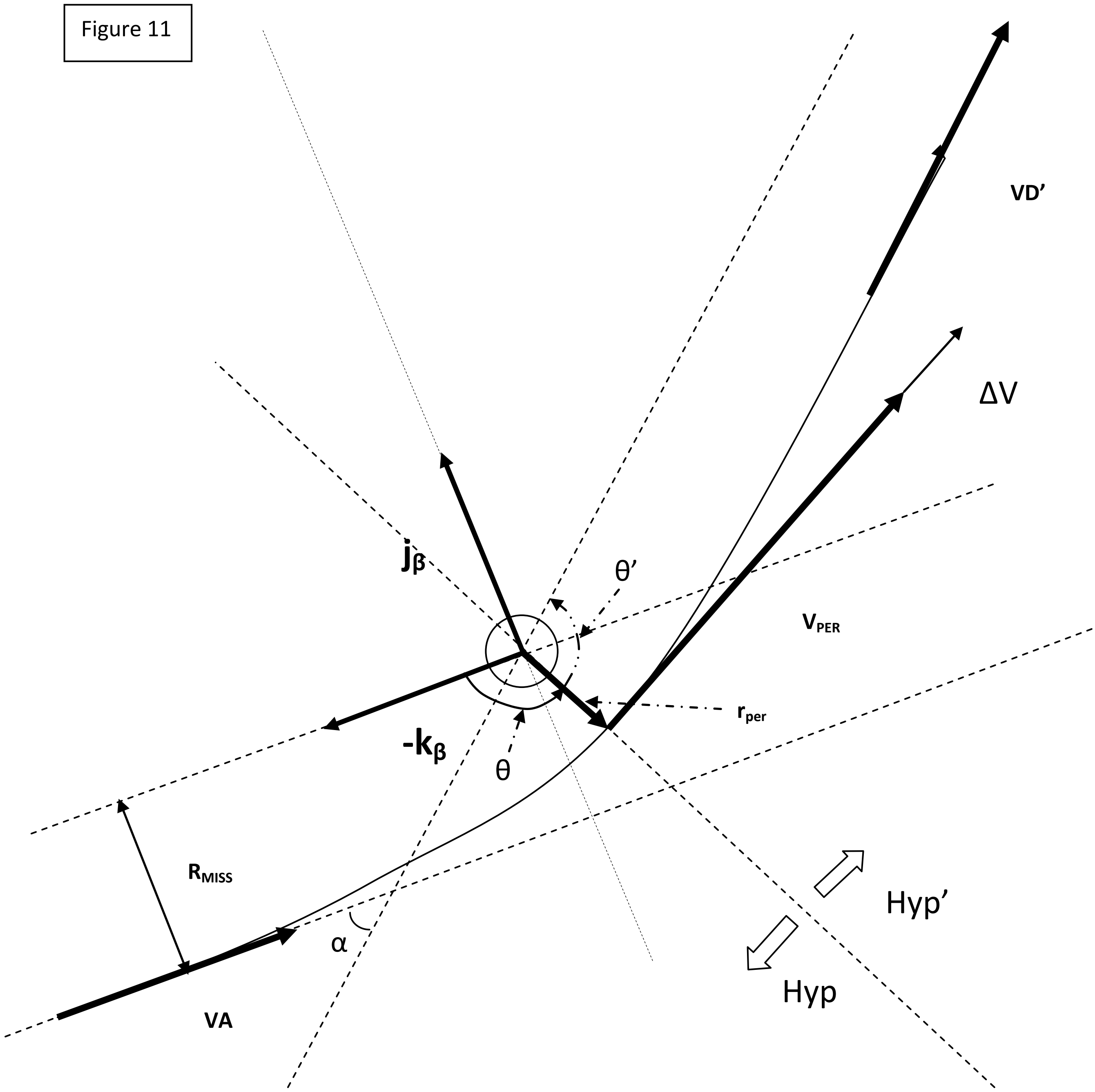


Figure 11



We know the arrival velocity vector **VA** and the departure velocity vector **VD'**. We assume an extra velocity impulse of ΔV is applied in the direction of the velocity at periapsis and in the orbital plane. This requires the presence of two hyperbolas, Hyp and Hyp' which meet at the periapsis point. The first, Hyp, is the hyperbola that the probe follows up to the periapsis point, the second, Hyp', is the one followed from periapsis to escape, which has the extra velocity at periapsis.

Thus:

$$r'_{PER} = r_{PER}$$

$$V'_{PER} = V_{PER} + \Delta V$$

We also have the following equations which apply respectively to Hyp and Hyp':

$$cos\theta = -\frac{\mu}{(\mu + r_{PER}VA^2)}$$

$$cos\theta' = -\frac{\mu}{(\mu + r_{PER}VD'^2)}$$

$$\alpha + \pi = \theta + \theta'$$

The above comprise three equations with three unknowns, i.e. θ, θ' and r_{PER}. (μ, the gravitational mass of the SSO, and **VA**,**VD',** and also the angle between them, α are known.) This is therefore a soluble problem with three simultaneous equations. The easiest approach is to solve for θ, thus:

$$cos\theta' = -(sin\alpha sin\theta + cos\alpha cos\theta)$$

$$VD'^2cos\theta'(1 + cos\theta) = VA^2cos\theta(1 + cos\theta')$$

With some manipulation we get the following equations which are of the form $f(\theta)=0$:

$$(VA^2 + VD'^2cos\alpha)cos\theta + \left(VD'^2 - VA^2\right)sin \propto cos\theta sin\theta + \left(VD'^2 - VA^2\right)cos \propto cos^2\theta + VD'^2sin \propto sin\theta = 0$$

$$\left((VD'^2 - VA^2)cos\theta + VD'^2\right)cos(\alpha - \theta) + VA^2cos\theta = 0$$

Alternatively solving for θ' we get the following equation, $g(\theta')=0$:

$$\left((VA^2 - VD'^2)\cos(\alpha - \theta') + VD'^2\right)cos(\theta') + VA^2\cos(\alpha - \theta') = 0$$

These can be solved numerically for either θ or θ' respectively by an appropriate technique such as a Newton-Raphson. The Derivative of $f(\theta)$ w.r.t θ. is:

$$(VD'^2 - VA^2)\sin(\alpha - 2\theta) + VD'^2\sin(\alpha - \theta) - VA^2sin\theta$$

And the derivative of $g(\theta')$ w.r.t. θ' is:

$$(VA^2 - VD'^2)\sin(\alpha - 2\theta') + VA^2\sin(\alpha - \theta') - VD'^2sin\theta'$$

The eccentricities, e, and e’ of the orbits Hyp and Hyp’ can be calculated from θ and θ’ respectively:

$$e = \frac{-1}{cos\theta}$$

$$e' = \frac{-1}{cos\theta'}$$

And r_{PER} can be calculated as follows:

$$r_{PER} = r'_{PER} = \frac{\mu}{VA^2}(e-1)$$

And a and a’ can be calculated from :

$$a = \frac{r_{PER}}{(1-e)}$$

$$a' = \frac{r_{PER}}{(1-e')}$$

The Value of ΔV is calculated as follows:

$$\Delta V = \sqrt{\left(\frac{2\mu}{r_{PER}} + VD'^2\right)} - \sqrt{\left(\frac{2\mu}{r_{PER}} + VA^2\right)}$$

1.6.7) Incorporating Encounter Dynamics (in 3D)

In the preceding section we determined all the necessary in-plane calculations (i.e. in 2D) for the trajectory of the probe inside the sphere of influence of an SSO. It transpires that given just the vectors **VA** and **VD’**, we can calculate, a, e, and a’, e’. These are all the parameters required to specify the shape of the orbits, Hyp and Hyp’.

However in addition to this information it can be observed that **VA** and **VD’** are also quite sufficient to specify the orientation and plane of the orbits Hyp and Hyp’. By our assumption Hyp and Hyp’ are co-planar and are in fact defined by the plane containing **VA** and **VD’**. The arrival geometry and 3D dynamics relative to the SSO (usually a Planet) are shown in Figures 12 to 18. Figure 18 shows the Angular Momentum Vector $\mathbf{H_A}$ for Hyp which as is clear, can be used to determine the angle of inclination of the orbit I (= I’) . The unit vector along $\mathbf{H_A}$ is $\mathbf{u_H}$ and is:

$$\boldsymbol{u_H} = \frac{\boldsymbol{cross(VA,\ VD')}}{\mathrm{VA\ VD'sin\alpha}}$$

$\mathbf{u_H}$ can also be used to get the Longtiude of Ascending Node, Ω (= Ω'). Finally Argument of Perigee ω (= ω') can be derived from the Laplace-Range-Lenz vector which for arrival, $\mathbf{E_A}$ is:

$$\boldsymbol{E_A = cross(VA,\ H_A) + \frac{\mu}{VA}\ VA}$$

Figure 14

$$\cos\delta = \frac{VA_y}{\sqrt{VA_y^2 + VA_x^2}}$$

$$\sin\delta = \frac{VA_x}{\sqrt{VA_y^2 + VA_x^2}}$$

$$\begin{bmatrix} i' \\ j' \\ k' \end{bmatrix} = \begin{bmatrix} cos\delta & -sin\delta & 0 \\ sin\delta & cos\delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix}$$

Figure 15

$$\cos\gamma = \frac{VA_z}{VA}$$

$$\sin\gamma = \frac{\sqrt{VA_y^2 + VA_x^2}}{VA}$$

$$\begin{bmatrix} i'' \\ j'' \\ k'' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{bmatrix} \begin{bmatrix} i' \\ j' \\ k' \end{bmatrix}$$

Figure 16

$$\begin{bmatrix} i_\beta \\ j_\beta \\ k_\beta \end{bmatrix} = \begin{bmatrix} cos\beta & sin\beta & 0 \\ -sin\beta & cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i'' \\ j'' \\ k'' \end{bmatrix}$$

Figure 18

$$\boldsymbol{uH_A} = \frac{cross\ (\boldsymbol{VA}, \boldsymbol{VD'})}{VA.VD'.sin\alpha} = \boldsymbol{i_\beta}$$

$\boldsymbol{uH_A}$ is the unit Angular Momentum Vector.

$\boldsymbol{i_\beta} = (cos\beta cos\delta - sin\beta cos\gamma sin\delta)\boldsymbol{i} + (cos\beta sin\delta + sin\beta cos\gamma cos\delta\ )\boldsymbol{j} - sin\beta sin\gamma\boldsymbol{k}$

By definition

$$\boldsymbol{i_\beta}.\boldsymbol{k} = \cos I$$

So

$$cosI = -sin\beta\ sin\gamma = \frac{cross(\boldsymbol{VA},\ \boldsymbol{VD'}).\boldsymbol{k}}{VA\,VD'sin\alpha}$$

$$sin\beta = 2\pi - \frac{VA_yVD'_x - VA_xVD'_y}{VD'sin\alpha\ \sqrt{VA_y^2 + VA_x^2}}$$

$$\boldsymbol{VD'} = -\boldsymbol{j}_{\boldsymbol{\beta}}(cos\theta sin\theta' + sin\theta cos\theta')VD' + \boldsymbol{k_\beta}(sin\theta sin\theta' - cos\theta cos\theta')VD'$$

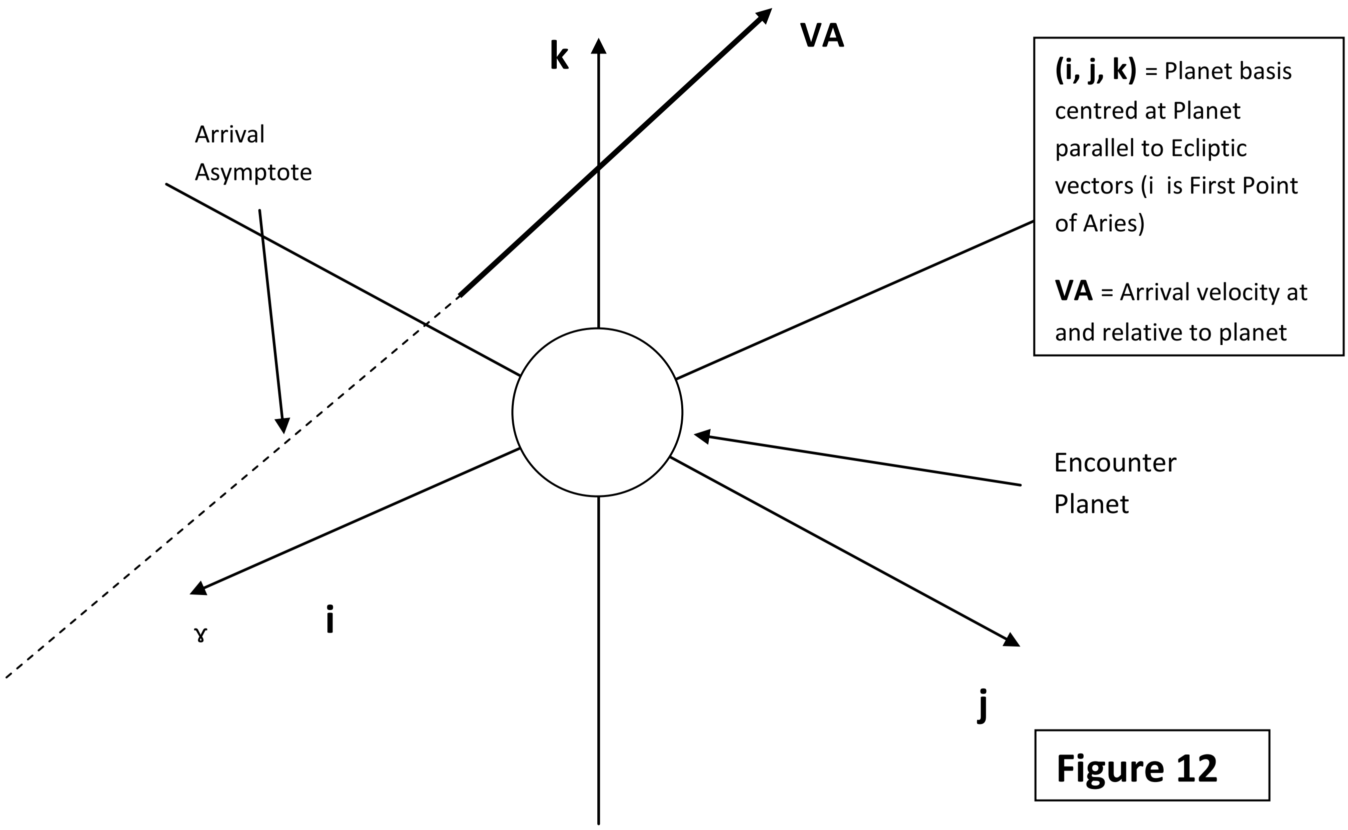
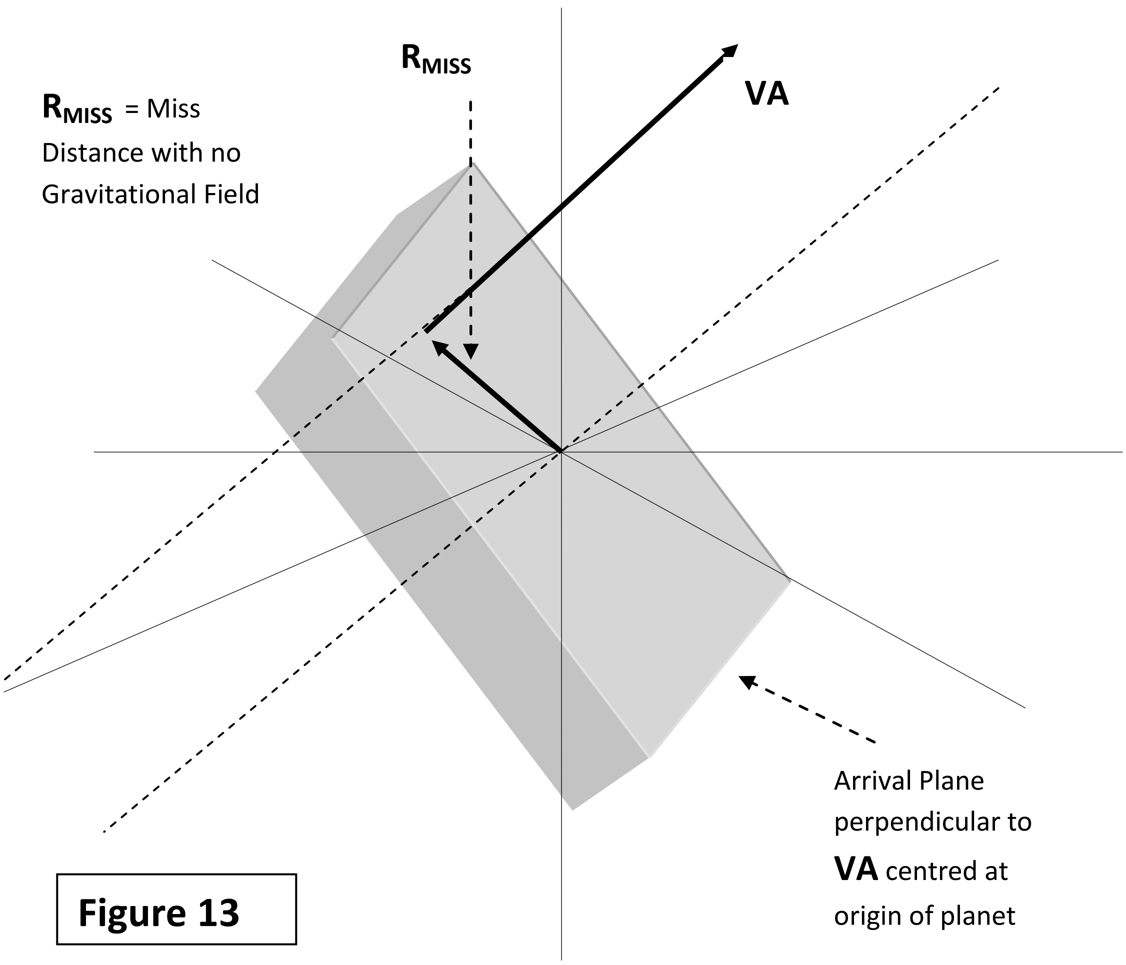


Figure 12



R_{MISS} = Miss Distance with no Gravitational Field

Arrival Plane perpendicular to **VA** centred at origin of planet

Figure 13

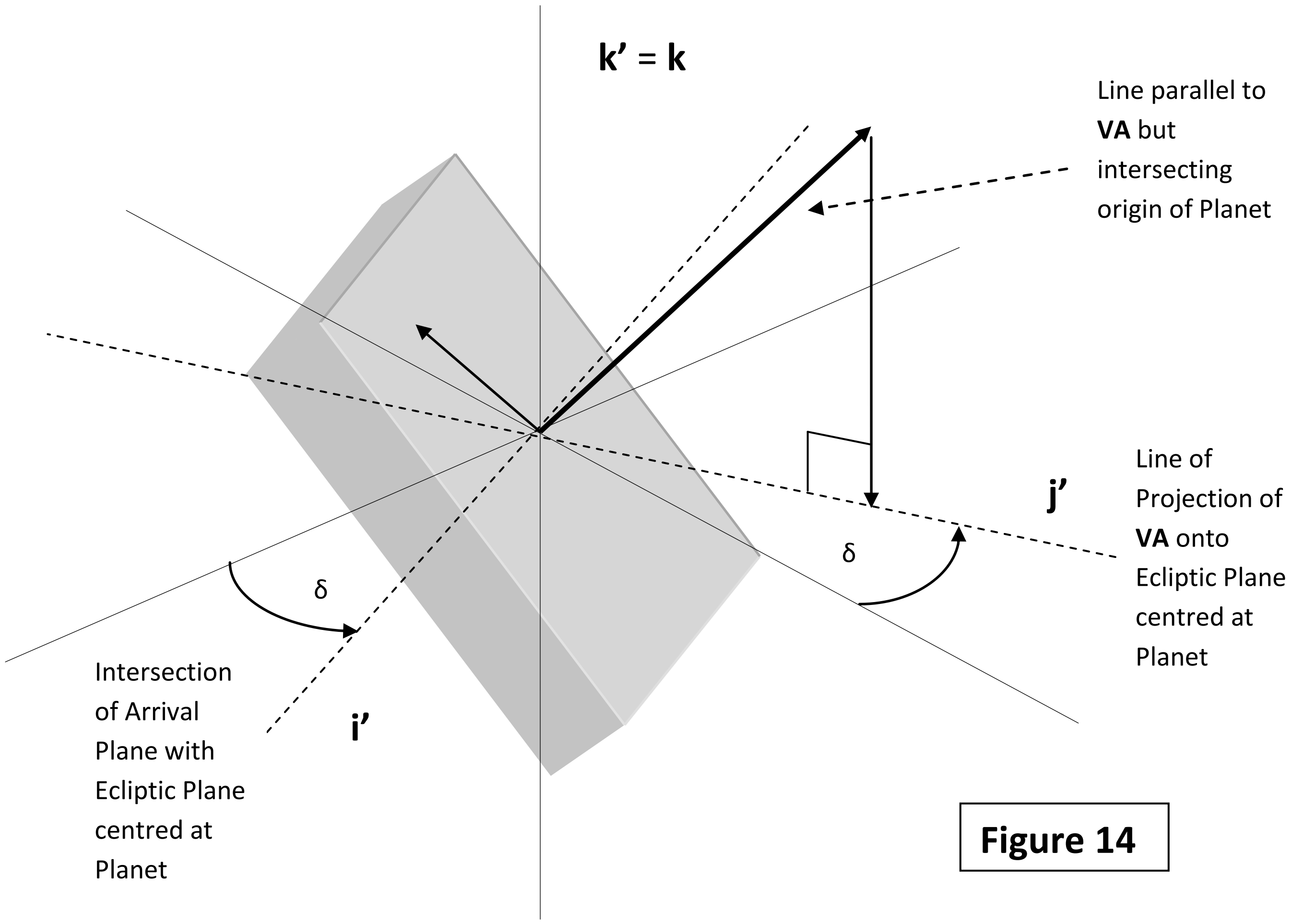


Figure 14

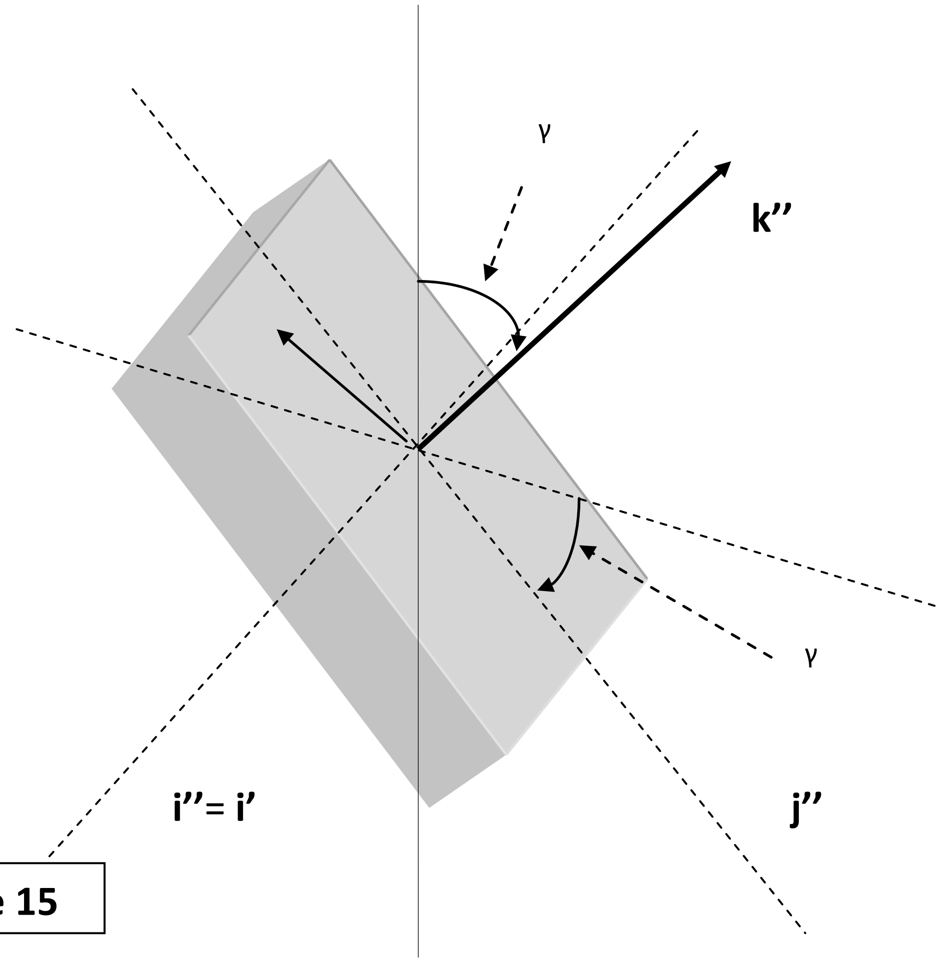


Figure 15

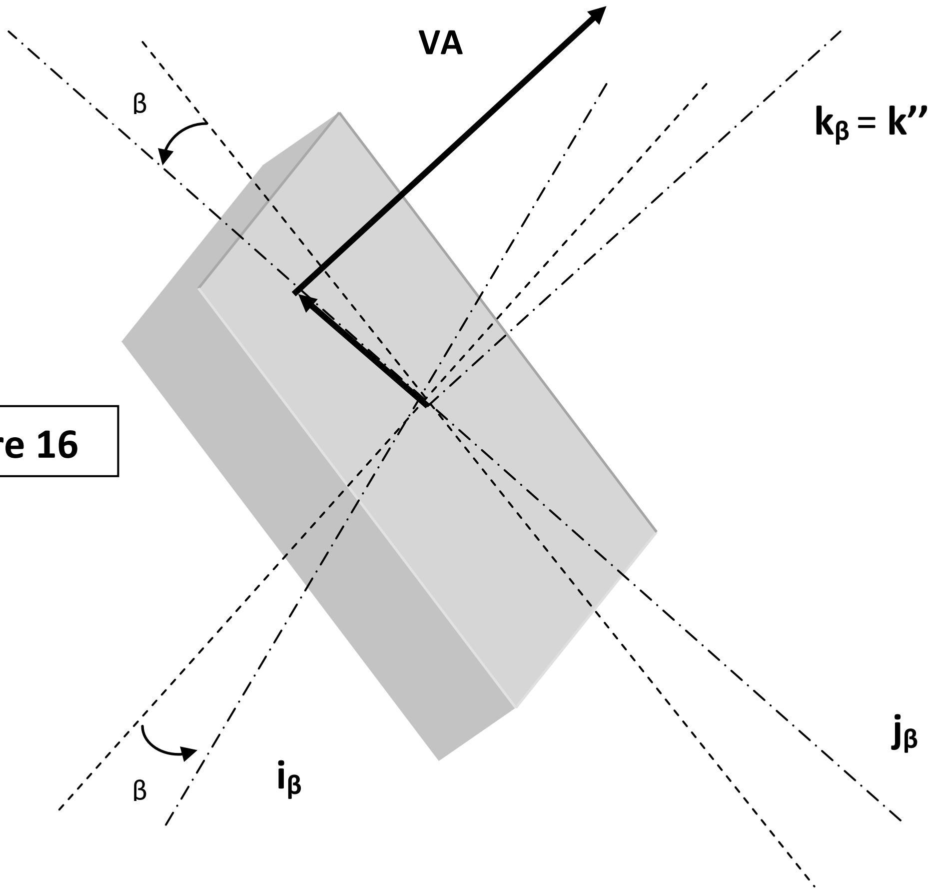


Figure 16

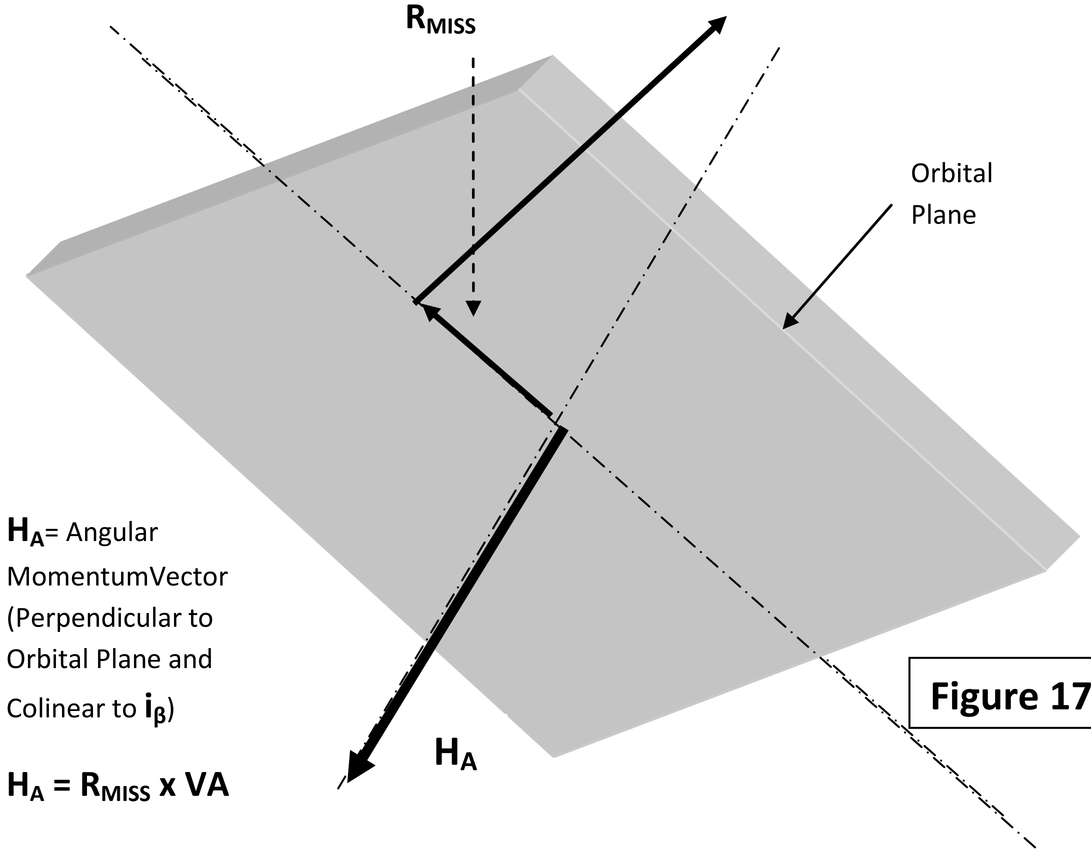


Figure 17

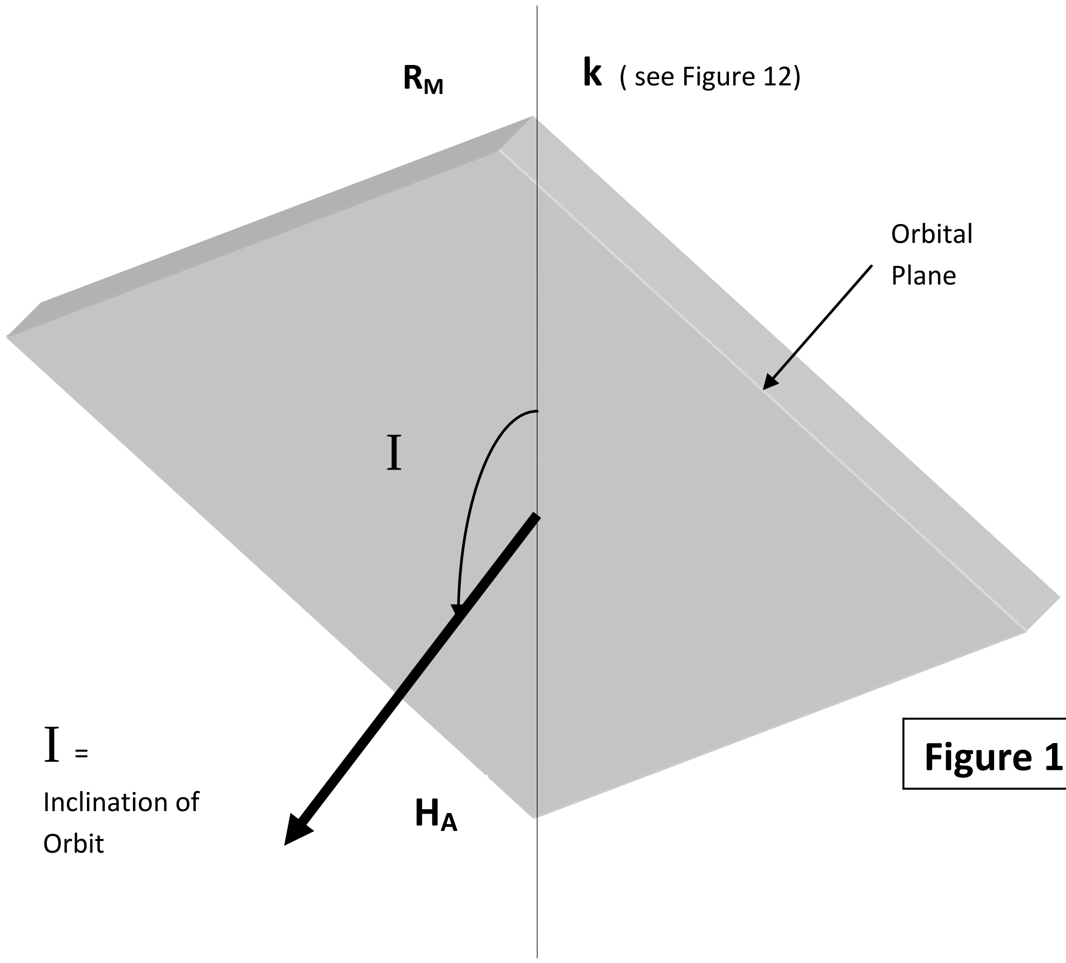
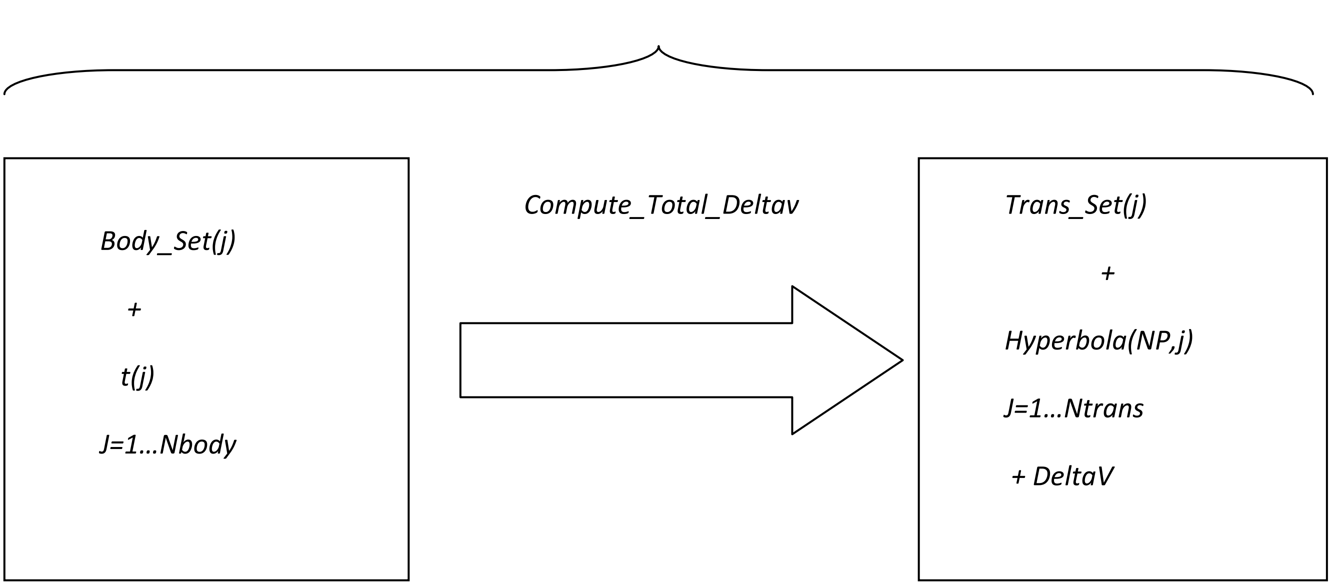


Figure 18

2) Application

We have dealt so far with the problem of determining an IT given a set of SSO's each at a certain time. We have found that by adopting a few assumptions, this is sufficient information to completely specify the IT, in terms of the interplanetary orbits connecting each successive pair of SSO's and the encounter orbits relative to each SSO in turn and so the overall DeltaV. Refer to Figure 19.

Figure 19



Object	Definition	Class	Base Class	Attribute of
IT	Interplanetary Trajectory	Nbody_Trajectory_With_Encounters	Nbody_Trajectory	-
Body_Set	Array of SSO's	Body	-	IT
Nbody	Number of SSO's	Integer	-	IT
Trans_Set	Array of Transfer Orbits	Transfer_orbits	-	IT
Ntrans	Nbody-1	Integer	-	IT
Hyperbola	2-Dim array of Connecting Hyperbolas	Connecting_Hyperbola	-	IT
BestDeltaV	Best DeltaV	double	-	IT

Stated simply, we have a function f as follows:

$$\Phi \quad = \quad f(t_1, t_2, t_3 \ldots t_{Nbody})$$

Where Φ is equivalent to *BestDeltaV* and f is essentially equivalent to *Compute_Total_Deltav*, a member function of *Nbody_Trajectory_With_Encounters*.

For each of the SSO encounters, j=1..Ntrans-1, there will be a minimum periapsis radius (possibly the radius of the Planet's equator). This provides us with the following inequality constraints:

$$g_j(t_1, t_2, t_3 \ldots t_{Nbody}) \leq 0 \quad j=1\ldots Ntrans-1$$

This is a Non-Linear Global Optimization Problem with Inequality Constraints.