

Notes on the Kepler Fitting Method

JOHN R. SMITH

email: john.smith7@ngc.com

*Northrop-Grumman Corporation
301 Voyager Way
Huntsville, Alabama USA*

December 3, 2020

1 Orbit Determination using Kepler Elements

The object of this project is to develop an Orbit Determination (OD) method based on using Kepler Elements as State Vectors.

As regards possible motivations for using such state vectors, consider the standard Cartesian State Vectors and the methods used to represent the trajectory. Basically the standard approach considers the actual trajectory to be a kind of stochastic process where discrete observations are used to update the state vector. If such an approach was imagined to be taken to the limit of continuous updating, then it is very likely that the trajectory would become a stochastic process similar to Brownian motion except that the mean time dependence would be given by the assumed motion model. There is a fundamental mathematical problem which may be of concern for this approach and that is that even continuous stochastic processes such as Brownian motion are continuous but nowhere differentiable. Brownian motion can be considered as the “integral” of white noise and there is a method of mean-squared integration that yields the reverse – that the mean-square “derivative” of Brownian motion is white noise. Therefore one wonders what significance the time-rate-of-change estimates that come out of updating a State Vector really are, or if there are fundamental limitations intrinsic to the method. Similar problems exist in the subject of polymer physics based on path-integral methods – usually a “stiffness” term is inserted ad-hoc into the mathematics to insure some kind of limits on the fluctuations which are encountered in numerical work. Such stiffness terms in a polymer are similar to demanding some level of “smoothness” in addition to continuity. So too with orbit determination methods. Are there intrinsic limitations on the velocity determination which are due to the state vector update methods? The idea behind this study is to examine replacing Cartesian elements with a quasi-state state vector in terms of Kepler elements and adopt a parameter estimation approach which represents not only the trajectory but also all the time-derivatives of that trajectory in a smooth manner and see if the resulting fit has a better performance vis-a-vis the covariance structure in position as well as velocity space.

The data can be a combination of measurement variables including: position \vec{r} , velocity \vec{v} , range ρ , line-of-sight unit vectors (angles θ and ϕ – or Az and El), range-rate $\dot{\rho}$, and angular rates $\dot{\theta}$ and $\dot{\phi}$ or any other quantity that can be expressed in terms of the osculating classical Kepler elements and time. We will start off by using only line-of-sight directions as the measurement data and assume that we are provided with line-of-sight unit vectors as measured by a moving observer – which is not necessarily fixed on the surface of the earth. The unit vectors are assumed to be given in the Earth-Centered Inertial (ECI) frame and are determined by 2 angles which can be taken as Az and El or as θ and ϕ in spherical polar coordinates as measured in the ECI-referential frame attached to the sensor location (the coordinate system whose unit vectors are parallel to the ECI system, but carried along by the sensor). In this case the dimension of the measurement data is 2 corresponding to the angular information. But we will allow for more measurement flavors and multiple sensors.

One way to model the problem of OD is to find a method of predicting the time dependence of the angular variables represented as a column vector $\alpha(t_i) = [\theta(t), \phi(t)]$ for the line-of-sight in such a way to minimize the statistical measure χ^2 defined by

$$\chi^2 = \frac{1}{2} \sum_{i=1}^N [\alpha(t_i) - \tilde{\alpha}_i]^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i], \quad (1)$$

where the sum is over the set of N measurements $\tilde{\alpha}_i = [\tilde{\theta}_i, \tilde{\phi}_i], t = 1, 2, \dots, N$ and we use brackets $[\]$ to denote column vectors denote transposition of a vector or a matrix by $[\]^T$. This method leads to the usual batch Least Squares Estimate (LSE) approach and can be generalized to include other variables such as range, range-rate and the others mentioned in the list above by extending the vectors to dimension n and the covariance structure to an n -by- n real symmetric matrix. The rank n of Cov for each term on the Right Hand Side (RHS) of the above sum and the number of independent measurements in the data which are required in order to make an estimate of the full set of parameters. Typically the number of measured variables is less than the number of parameters being determined and therefore a minimum number of independent measurements are required before the problem reaches full rank and a estimate of the set of parameters possible. Of course, if an initial estimate of the state is available, then of course, updates to the parameter values can ensue via a sequential revision of the Least Squares method.

As an example, an extension of this problem to the list of variables mentioned at the beginning would require a measurement vector of the form

$$\alpha = [\vec{r}, \vec{v}, \rho, \theta, \phi, \dot{\rho}, \dot{\theta}, \dot{\phi}]$$

at time t and a corresponding prediction vector of the same quantities given in terms of the Kepler elements and the time t . This will be apparent to us as we proceed as the estimation process will involve solving a linear system and that linear system is constructed from a sum of terms each of which may have less rank than the set of parameters requires, but enough independent measurements are used so that the complete linear system has full rank. Otherwise the problem in batch mode is unsolvable – we need to have enough data to perform the estimate from scratch. This will not preclude incorporating new data with less than full rank to an initial estimate.

It doesn't hurt to work out a general procedure for handling many flavors of measurements and then apply it to lower dimensional problems by simply turning off some of the terms – a procedure that can do more, can also do less. We will apply this problem to the case of line-of-sight measurements, by ignoring the irrelevant terms in the vector of observations.

Other method of approaching the OD problem might use the Minimum Variance Estimate (MVE), or Maximum Likelihood (ML), or Restricted Maximum Likelihood (REML). We will work out a method based in minimizing χ^2 which has ML as its counterpart, MVE and REML are analogous. In REML the object is to minimize the “error contrasts” which is similar to minimizing the variances occurring in the problem.

The next thing to notice is that the minimization of χ^2 has several manifestations. One can attempt to minimize χ^2 by computing its derivatives with respect to the parameter set $Z = [z_1, z_2, z_3, \dots, z_m]$ by computing the derivative vector and solving for

$$\frac{\partial \chi^2}{\partial z_j} = 0, \quad j = 1, 2, \dots, m \quad (2)$$

This method usually is based on a first order Taylor expansion of the prediction vector $[\theta(t), \phi(t)]$ in terms of the parameter set Z and, in the case of nonlinear functions, performing iterations via the Newton-Raphson technique. Application of the first order technique can be found in [1] to vertex fitting methods in High Energy Physics which are the underlying theory for the SCOUT methods[2]

This isn't the only way of setting up the minimization problem. For example one can look at the problem from the point of view of finding the critical points of the gradient of χ^2 itself without worrying first about whether χ^2 is a minimum. In other words, there is a second order method which seeks to solve for the critical points by expanding the gradient in its own linear Taylor expansion, viz.,

$$\frac{\partial \chi^2}{\partial z_j} \approx \left. \frac{\partial \chi^2}{\partial z_j} \right|_{\hat{Z}} + \sum_{k=1}^N \left. \frac{\partial^2 \chi^2}{\partial z_j \partial z_k} \right|_{\hat{Z}} \delta z_k, \quad j = 1, 2, \dots, m,$$

where $\delta z_k = z_k - \hat{z}_k$ and \hat{Z} is the expansion point for the Taylor series. In this method the idea is to apply

Newton-Raphson iteration to the equation

$$0 = \left. \frac{\partial \chi^2}{\partial z_j} \right|_{\hat{Z}} + \sum_{k=1}^N \left. \frac{\partial^2 \chi^2}{\partial z_j \partial z_k} \right|_{\hat{Z}} \delta z_k, \quad j = 1, 2, \dots, m,$$

which can be turned into a linear system to solve for the increment δz_k to be used as the expansion point

$$\hat{Z} \rightarrow \hat{Z} + \delta z$$

to be used in the next iteration. This updating is continued via Newton-Raphson iteration until (hopefully) the method converges and $\delta Z \rightarrow \mathbf{0}$ (the zero vector). The second order method takes more information into account, for example it uses the information on how the derivatives of χ^2 themselves are changing as a function of the parameter set. Therefore this method also requires computing the Hessian matrix, \mathbf{H} , whose components are given by

$$H_{ij} = \frac{\partial^2 \chi^2}{\partial z_j \partial z_k}, \quad i, j = 1, 2, \dots, m$$

We will proceed to outline both the first order method as well as the second order method with respect to the prediction vector for the line-of-sight prediction vector $[\theta(t), \phi(t)]$. To apply the first order method we make the assumption that the linear approximation to the prediction vector near the minimum of χ^2 is “good enough” meaning that is is “reasonably linear” where the minimum of χ^2 is located. The may not be the case especially if the parameter space has hard boundaries and the minimum is located near those boundaries – various problems can occur, e.g. the parameters may become complex numbers due to exceeding the physical range of possibilities.

2 BFGS – Quasi-Newton Optimization Method

Newton’s Method for finding the minimum of the Objective Function requires computing the matrix of mixed 2nd-order partial derivatives of the Objective function – the Hessian matrix \mathbf{H} whose elements are given by

$$H_{jk} = \frac{\partial^2 \chi^2}{\partial p_j \partial p_k}$$

As long as the Objective Function is reasonably parabolic in the region of the minimum and that the initial guess is not too far away, this method converges quadratically. However, we would like to check these two conditions and construct an alternative to investigate robustness properties.

Let us consider the case of N independent measurements each on of dimension m . Also we consider a list of n parameters, θ , to be estimated. In this case, the gradient of the Objecting Function can be written as a sum over all the measurements using the method of Backwards Differentiation as follows: For each of the N measurements compute the contribution of each derivative to the total and add them together

$$\frac{\partial \chi^2}{\partial p_j} = \sum_{d=1}^N \frac{\partial \chi_d^2}{\partial p_j}$$

Each contributing partial derivative can be obtained using the method of Backwards Differentiation:

1. Initialize $F_j = 0$, for $i < r$ and set $F(r) = 1$ where r corresponds to the line reference for χ_d^2
2. $F_i = F_i + F_j (\partial h_k / \partial h_j)$ for all i such that $h_j \in \Omega_k$, $k = r, r-1, \dots, n+1$
3. Then $\partial h_r / \partial h_i = F_i$, $i = 1, 2 \dots r$.
4. $(\nabla \chi^2)_v = \sum_{d=1}^N \left[\frac{\partial h_r}{\partial p_v} \right]_d$.

where the final sum is over the respective elements in the measurement set. With the sum in hand we can rewind the data and use the summed gradient vector to initialize the lower part of the auxilliary work array \mathbf{Q} using

1. Initialize $Q_v = (\nabla\chi^2)_v, v = 1, \dots, n$, and $Q_j = 0$ for $j > n$ and $S_j = 0, \forall j = 1, 2, \dots, r$
2. $Q_k = Q_K + Q_i(\partial h_k / \partial h_i), h \in \Omega_k$;
 $S_j = S_j + Q_i F_k(\partial^2 h_k / \partial h_i \partial h_j), h_i, h_k \in \Omega_k, k = n+1, n+2, \dots, r$
3. $S_j = S_j + S_k(\partial h_k / \partial h_i), h_j \in \Omega_k, k = r, r-1, \dots, n+1$
4. Then $\partial^2 h_r / \partial x_v \partial x_j = S_i, i = 1, 2, \dots, n$
5. $\mathbf{S}_i^{total} = \sum_{d=1}^N [S_i]_d$

It is important to note that the F array appearing in the above corresponds to the respective Backwards Differentiation of the respective measurement data point under considered.

The relevance to the rank-one update version of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm involves using the above construction to express the secant condition at the update for the q -th Newton step:

$$H_{q+1} * (\theta_{q+1} - \theta_q) = \nabla\chi^2(\theta_{q+1}) - \nabla\chi^2(\theta_q)$$

As an alternative to this finite-difference based secant equation, we can use the following relationship between the Hessian matrix \mathbf{H} and the directional derivative and the constructed vector of second derivatives \mathbf{S}^{total} .

$$\mathbf{H}\nabla\chi^2(\theta) = \mathbf{S}^{total}$$

The Update Equations begin with the definition

$$H_{q+1} = H_q + \mathbf{u}\mathbf{u}^T$$

applied to the accumulated gradient and its second derivatives:

$$\begin{aligned}\mathbf{H}_{q+1}\nabla\chi^2 &= (\mathbf{H}_q + \mathbf{u}\mathbf{u}^T)\nabla\chi^2 = \mathbf{S}^{total} \\ \mathbf{u}(\mathbf{u}^T\nabla\chi^2) &= \mathbf{S}^{total} - \mathbf{H}_q\nabla\chi^2\end{aligned}$$

Let $\alpha = \mathbf{u}^T\nabla\chi^2$. Then by pre-multiplying the above equation by $(\nabla\chi^2)^T$ we find that

$$\alpha^2 = (\nabla\chi^2)^T \mathbf{u}\mathbf{u}^T \nabla\chi^2 = (\nabla\chi^2)^T (\mathbf{S}^{total} - \mathbf{H}_q\nabla\chi^2)$$

and finally we find that

$$\mathbf{u} = (\mathbf{S}^{total} - \mathbf{H}_q\nabla\chi^2) / \alpha$$

Also, from the Sherman-Morrison-Woodbury formula for inverses of matrix updates we have

$$(\mathbf{A} + \mathbf{U}\mathbf{B}\mathbf{V}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{B}^{-1} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1}$$

which can be applied in this case with $\mathbf{A} = \mathbf{H}_q$, $\mathbf{U} = \mathbf{V} = \mathbf{u}$ and $\mathbf{B} = \mathbf{I}_{1 \times 1} = 1$ to obtain

$$(\mathbf{H} + \mathbf{u}\mathbf{u}^T)^{-1} = \mathbf{H}_q^{-1} - \mathbf{H}_q^{-1}\mathbf{u}(\mathbf{I} + \mathbf{u}^T\mathbf{H}_q^{-1}\mathbf{u})^{-1}\mathbf{u}^T\mathbf{H}_q^{-1}$$

which, because \mathbf{u} is a vector and therefore $\mathbf{u}\mathbf{u}^T$ is a rank-1 update to \mathbf{B}_q , we find that

$$\mathbf{B}_{q+1} = \mathbf{H}_{q+1}^{-1} = (\mathbf{H}_q + \mathbf{u}\mathbf{u}^T)^{-1} = \mathbf{H}_q^{-1} - \frac{\mathbf{H}_q^{-1}\mathbf{u}\mathbf{u}^T\mathbf{H}_q^{-1}}{(1 + \mathbf{u}^T\mathbf{H}_q^{-1}\mathbf{u})}$$

The final operation to be performed in the Newton-step is to update the parameter list via

$$\theta_{q+1} = \theta_q + \mathbf{B}_{q+1}\nabla\chi^2$$

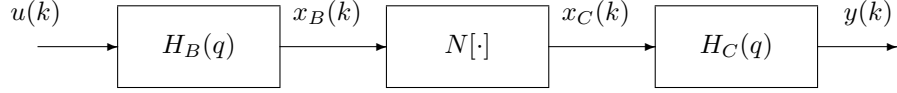


Figure 1: An LNL Block Oriented Model Structure

3 Fundamental Orbit Determination

Prediction theory:

Consider estimating the parameters of a straight line $y = a + bx$. We modify the problem by using the observed data $\{y_i, x_i\}$ $i = 1, \dots, n$ and cast the problem about the mean of the data:

$$y - \bar{y} = b(x - \bar{x}),$$

subject to $\bar{y} = a + b\bar{x}$ which casts the parameter a in the form $a = \bar{y} - b\bar{x}$. Using the data we can obtain estimates of the parameters based on statistical concepts, i.e.,

$$\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x}) = b \sum_{i=1}^n (x_i - \bar{x})^2,$$

or in statistical language $\sigma_{xy} = b\sigma_x^2$ or

$$b = \frac{\sigma_{xy}}{\sigma_x^2}$$

The general non-linear problem of Orbit Determination involves estimating parameters associated with a state motion model of the form:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, t), \quad \mathbf{X}_k = \mathbf{X}(t_k) \quad (3)$$

$$\mathbf{Y}_i = \mathbf{G}(\mathbf{X}_i, t_i) + \epsilon_i; \quad i = 1, 2, \dots, l \quad (4)$$

where \mathbf{X}_k is an unknown n -dimensional State Vector at time t_k . Also \mathbf{Y}_i is a p -dimensional set of measurements for $i = 1, 2, \dots, l$, that are to be used to estimate the unknown value \mathbf{X}_k which we denote by $\hat{\mathbf{X}}_k$. Finally ϵ_i represents the random measurement uncertainty which we usually assume to be given by a multivariate Gaussian distribution with $\mathbf{Cov} = E[\epsilon\epsilon^T]$. In general $p < n$ and $m = p \times l \gg n$. Let $\mathbf{X}^*(t)$ represent the numerical integration of Eq.(3) with initial conditions $\mathbf{X}^*(t_0)$. $\mathbf{X}^*(t)$ will be referred to as the reference State Vector – we will implement an iterative method that numerically moves the reference State Vector so as to minimize an objective function so as to obtain the best estimate of the State Vector at time t_0 .

Now let us look at small changes in the State Vector about the reference State Vector and consider their effect on the prediction of the RHS of the Eq.(4). Expanding to first order in a Taylor series about small differences in the reference State Vector

$$\begin{aligned} \mathbf{Y}_i &\approx \mathbf{G}(\mathbf{X}^*, t_i) + \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}} \right]_i^* [\mathbf{X}(t_i) - \mathbf{X}^*(t_i)] + \epsilon_i, \\ &= \mathbf{G}(\mathbf{X}^*, t_i) + \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}} \right]_i^* \mathbf{x}_i + \epsilon_i \end{aligned}$$

where $\mathbf{x}_i = \mathbf{X}(t_i) - \mathbf{X}^*(t_i)$. If we also use lower case for the difference $\mathbf{y}_i = \mathbf{Y}_i - \mathbf{G}(\mathbf{X}^*, t_i)$ we can rewrite the above expression as

$$\begin{aligned}\mathbf{x}_i &= \mathbf{X}(t_i) - \mathbf{X}^*(t_i), \\ \mathbf{y}_i &\approx \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}} \right]_i^* \mathbf{x}_i + \epsilon_i\end{aligned}$$

If we use numerical integration to determine the time dependence of $\mathbf{X}^*(t)$, then the above can be recast to use the differences in the reference State Vector at t_0 and a nearby State Vector $\mathbf{x}_0 = \mathbf{X}(t_0) - \mathbf{X}^*(t_0)$

$$\begin{aligned}\mathbf{x}_0 &= \mathbf{X}(t_0) - \mathbf{X}^*(t_0), \\ \mathbf{y}_i &\approx \left[\frac{\mathbf{G}(\mathbf{X}^{+\delta}, t_i) - \mathbf{G}(\mathbf{X}^{-\delta}, t_i)}{2\vec{\delta}} \right] \mathbf{x}_0 + \epsilon_i\end{aligned}\tag{5}$$

where $\mathbf{X}^{\pm\delta}$ is the numerical integration of

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, t)$$

with the initial conditions $\mathbf{X}^{\pm\delta}(t_0) = \mathbf{X}^*(t_0) \pm \vec{\delta}$. Let \mathbf{H}_i be defined by using finite differences of two numerical integrations of the system Eq.(3).

$$\mathbf{H}_i = \left[\frac{\mathbf{G}(\mathbf{X}^{+\delta}, t_i) - \mathbf{G}(\mathbf{X}^{-\delta}, t_i)}{2\vec{\delta}} \right]\tag{6}$$

and we can summarize the model in the form

$$\begin{aligned}\mathbf{x}_0 &= \mathbf{X}(t_0) - \mathbf{X}^*(t_0), \\ \mathbf{y}_i &\approx \mathbf{H}_i \mathbf{x}_0 + \epsilon_i.\end{aligned}\tag{7}$$

If required we can also work out the time dependence of $\mathbf{x}(t)$ by numerical integration, *viz.*

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{X}(t) - \mathbf{X}^*(t) \\ &\approx \left[\frac{\partial \mathbf{X}(t)}{\partial \mathbf{X}(t_0)} \right] \mathbf{x}_0 \text{ for small differences}\end{aligned}$$

This leads for a numerical approximation for the State Transition Matrix $\Phi(t, t_0)$:

$$\begin{aligned}\mathbf{x}_0 &= \delta \text{ assume small deviations from the reference value} \\ \Phi(t, t_0) &= \left[\frac{\partial \mathbf{X}(t)}{\partial \mathbf{X}(t_0)} \right] \approx \left[\frac{\mathbf{X}^{+\delta}(t) - \mathbf{X}^{-\delta}(t)}{2\delta} \right] \\ \mathbf{x}(t) &= \Phi(t, t_0) \mathbf{x}_0\end{aligned}$$

We can also use $\Phi(t, t_0)$ to estimate propagation of the fitted Covariance of the State Vector as a function of time

$$\bar{\mathbf{P}}_k = \mathbf{Cov}(t_k) = \Phi(t_k, t_0) * \mathbf{Cov}(t_0) * \Phi^T(t_k, t_0).$$

In this case we are considering the covariance of the fitted parameters and so denote them as $\mathbf{P}_k = \mathbf{Cov}(t_k)$. Similarly we can use this method for a different reference time as follows:

$$\bar{\mathbf{P}}_k = \mathbf{Cov}(t_k) = \Phi(t_k, t_j) * \mathbf{Cov}(t_j) * \Phi^T(t_k, t_j).$$

Using these definitions we can express Eq.(6) as

$$\mathbf{H}_i = \left[\frac{\partial \mathbf{G}(\mathbf{X}, t_i)}{\partial \mathbf{X}(t_i)} \right] \left[\frac{\partial \mathbf{X}(t_i)}{\partial \mathbf{X}(t_0)} \right] \text{ from the chain rule}$$

Let $\tilde{\mathbf{H}}_i = [\partial \mathbf{G}(\mathbf{X}, t_i) / \partial \mathbf{X}(t_i)]$. Then we have $\mathbf{H}_i = \tilde{\mathbf{H}}_i \Phi(t_i, t_0)$.

In our application we want to estimate the parameters of the State Vector at a particular time t_k . In principle there is a state vector \mathbf{x}_i corresponding to each measurement. In the above context that would imply that there are $n \times l$ State Vector components for l measurements:

$$\begin{aligned} \mathbf{y}_1 &= \tilde{\mathbf{H}}_1 \mathbf{x}_1 + \epsilon_1 \\ \mathbf{y}_2 &= \tilde{\mathbf{H}}_2 \mathbf{x}_2 + \epsilon_2 \\ &\vdots \\ \mathbf{y}_l &= \tilde{\mathbf{H}}_l \mathbf{x}_l + \epsilon_l \end{aligned} \tag{8}$$

The State Vectors in Eq.(8) can be related to a particular State Vector at time t_0 by use of the State Transition Matrix $\Phi(t_k, t_0)$, $k = 1, 2, \dots, l$:

$$\begin{aligned} \mathbf{y}_1 &= \tilde{\mathbf{H}}_1 \Phi(t_1, t_0) \mathbf{x}_0 + \epsilon_1 \\ \mathbf{y}_2 &= \tilde{\mathbf{H}}_2 \Phi(t_2, t_0) \mathbf{x}_0 + \epsilon_2 \\ &\vdots \\ \mathbf{y}_l &= \tilde{\mathbf{H}}_l \Phi(t_l, t_0) \mathbf{x}_0 + \epsilon_l \end{aligned} \tag{9}$$

In terms of the matrices \mathbf{H}_i , obtained by applying finite differences to the numerical integration of Eq. (6), we have

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{H}_1 \mathbf{x}_0 + \epsilon_1 \\ \mathbf{y}_2 &= \mathbf{H}_2 \mathbf{x}_0 + \epsilon_2 \\ &\vdots \\ \mathbf{y}_l &= \mathbf{H}_l \mathbf{x}_0 + \epsilon_l \end{aligned} \tag{10}$$

We seek a sequential method for estimating the initial State Vector \mathbf{x}_0 based on adding a new measurement $k = l + 1$ to the previous estimate at $k = l$. In other words, we can add a new measurement to the estimation

$$\mathbf{x}_0 \approx \hat{\mathbf{x}}_{l+1} = \mathbf{X}(t_0)_{l+1} - \mathbf{X}(t_0)_l = \Delta \mathbf{X}(t_0)_{l+1}$$

4 Sequential Least Squares

Consider the Generalized Least Square averaging formula applied to the Linear Model $Y = \mathbf{H}\beta + \epsilon$. The covariance matrix for the measurements y is given by $\mathbf{Cov} = E(\epsilon\epsilon^T)$. The Least Squares estimate for β is given by

$$\hat{\beta}_N = \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} \mathbf{H}_i \right)^{-1} \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} Y_i \right) \tag{11}$$

Let $\mathbf{A}_N = \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} \mathbf{H}_i \right)^{-1}$ be the coefficient matrix on the RHS of Eq. (11). \mathbf{A}_N is the estimated Covariance of $\hat{\beta}_N$. We can write $\hat{\beta}_N = \mathbf{A}_N \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} Y_i \right)$. Now consider what happens when we add a new measurement. Using all the data again in a batch estimate we have $\hat{\beta}_{N+1} = \mathbf{A}_{N+1} \left(\sum_{i=1}^{N+1} \mathbf{H}_i^T \mathbf{Cov}_i^{-1} Y_i \right)$, with $\mathbf{A}_{N+1} = \left(\sum_{i=1}^{N+1} \mathbf{H}_i^T \mathbf{Cov}_i^{-1} \mathbf{H}_i \right)^{-1}$. Let us break off the last measurement $i = N + 1$ and consider the separate items:

$$\hat{\beta}_{N+1} = \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} \mathbf{H}_i + \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} \mathbf{H}_{N+1} \right)^{-1} \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} Y_i + \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} Y_{N+1} \right) \tag{12}$$

Comparing the above expressions we can also express the obvious relationship between \mathbf{A}_N and \mathbf{A}_{N+1} as

$$\mathbf{A}_{N+1}^{-1} = \mathbf{A}_N^{-1} + \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} \mathbf{H}_{N+1}.$$

In terms of the above matrices we can write the previous estimate with N data points:

$$\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} Y_i = \mathbf{A}_N^{-1} \hat{\beta}_N = (\mathbf{A}_{N+1}^{-1} - \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} \mathbf{H}_{N+1}) \hat{\beta}_N$$

The Sherman-Morrison-Woodbury formula for inverses takes the form;

$$(\mathbf{A}_N^{-1} + \mathbf{U} \mathbf{B}^{-1} \mathbf{V}^T)^{-1} = \mathbf{A}_N - \mathbf{A}_N \mathbf{U} (\mathbf{B} + \mathbf{V}^T \mathbf{A}_N \mathbf{U})^{-1} \mathbf{V}^T \mathbf{A}_N \quad (13)$$

Let \mathbf{A}_N be defined as above, $\mathbf{B} = \mathbf{Cov}_{N+1}$, $\mathbf{U} = \mathbf{H}_{N+1}^T$ and $\mathbf{V}^T = \mathbf{H}_{N+1}$. Inserting these expressions and using Eq. (13) into the first term on the RHS of Eq.(12) we obtain the update to the fitted Covariance Matrix:

$$\begin{aligned} \mathbf{A}_{N+1} &= \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} \mathbf{H}_i + \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} \mathbf{H}_{N+1} \right)^{-1} \\ &= \mathbf{A}_N - \mathbf{A}_N \mathbf{H}_{N+1}^T (\mathbf{Cov}_{N+1} + \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T)^{-1} \mathbf{H}_{N+1} \mathbf{A}_N \end{aligned} \quad (14)$$

We also obtain the update to the fitted parameter state:

$$\begin{aligned} \hat{\beta}_{N+1} &= \mathbf{A}_{N+1} \left(\sum_{i=1}^N \mathbf{H}_i^T \mathbf{Cov}_i^{-1} Y_i + \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} Y_{N+1} \right) \\ &= \mathbf{A}_{N+1} \left[(\mathbf{A}_{N+1}^{-1} - \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} \mathbf{H}_{N+1}) \hat{\beta}_N + \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} Y_{N+1} \right] \\ &= \hat{\beta}_N + \mathbf{A}_{N+1} \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} (Y_{N+1} - \mathbf{H}_{N+1} \hat{\beta}_N) \end{aligned}$$

Using Eq.(14) we also can express

$$\begin{aligned} \mathbf{A}_{N+1} \mathbf{H}_{N+1}^T &= [\mathbf{A}_N - \mathbf{A}_N \mathbf{H}_{N+1}^T (\mathbf{Cov}_{N+1} + \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T)^{-1} \mathbf{H}_{N+1} \mathbf{A}_N] \mathbf{H}_{N+1}^T \\ &= \mathbf{A}_N \mathbf{H}_{N+1}^T [\mathbf{I}_N - (\mathbf{Cov}_{N+1} + \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T)^{-1} \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T] \\ &= \mathbf{A}_N \mathbf{H}_{N+1}^T (\mathbf{Cov}_{N+1} + \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T)^{-1} [(\mathbf{Cov}_{N+1} + \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T) - \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T] \\ &= \mathbf{A}_N \mathbf{H}_{N+1}^T (\mathbf{Cov}_{N+1} + \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T)^{-1} \mathbf{Cov}_{N+1} \end{aligned}$$

If we define a Gain Matrix as follows, $\mathbf{K}_{N+1} = \mathbf{A}_{N+1} \mathbf{H}_{N+1}^T \mathbf{Cov}_{N+1}^{-1} = \mathbf{A}_N \mathbf{H}_{N+1}^T (\mathbf{Cov}_{N+1} + \mathbf{H}_{N+1} \mathbf{A}_N \mathbf{H}_{N+1}^T)^{-1}$, we can also write the update formulae as:

$$\mathbf{A}_{N+1} = [\mathbf{I}_N - \mathbf{K}_{N+1} \mathbf{H}_{N+1}] \mathbf{A}_N \quad \text{and} \quad \hat{\beta}_{N+1} = \hat{\beta}_N + \mathbf{K}_{N+1} (Y_{N+1} - \mathbf{H}_{N+1} \hat{\beta}_N).$$

5 Application of the first order method

Let us denote the prediction vector as

$$\alpha(t) = \begin{bmatrix} \theta(t) \\ \phi(t) \end{bmatrix} \quad \text{and the measurement data vectors as } \tilde{\alpha} = \begin{bmatrix} \tilde{\theta} \\ \tilde{\phi} \end{bmatrix}$$

In this case the dimension n of the measurement vector is $n = 2$. We expand $\alpha(t)$ about a suitably chosen expansion point (which will be iterated) in the m parameters Z_A , and define $\alpha(t)|_{Z_A} \equiv \alpha_A(t)$ as the value of the predicted observation at time t with respect to the current parameter set (the expansion point A). We also define the differences $\delta\alpha(t) = \alpha(t) - \alpha_A$ and $\delta\tilde{\alpha}(t) = \tilde{\alpha} - \alpha_A$.

The first order method involves a Taylor expansion of the prediction functions $\alpha_i(t)$ in terms of the components of the basic parameter set \mathbf{Z} :

$$\alpha(t) \approx \alpha + A(t) + \frac{\partial \alpha(t)}{\partial Z_k} \Big|_{Z_A} \delta z_k$$

In order to economize the notation let us define the following n -by- m Jacobian matrix \mathbf{D} :

$$D_{ik} = \frac{\partial \alpha_i(t)}{\partial Z_k} \Big|_{Z_A}, \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m$$

Performing the differentiation of each term in the sum on the RHS of Eq.(1), each of the $i = 1, 2, \dots, N$ measurements yields a set of equations of the form

$$\mathbf{D}(t_i)^T \mathbf{Cov}^{-1} \delta \tilde{\alpha}(t_i) = \mathbf{D}(t_i)^T \mathbf{Cov}_i^{-1} \mathbf{D}(t_i) \delta \mathbf{z}$$

summing over the N data points we obtain the LSE equation which the increment $\delta \mathbf{z}$ must satisfy

$$\delta \mathbf{z} = \left[\sum_{i=1}^N \mathbf{D}(t_i)^T \mathbf{Cov}_i^{-1} \mathbf{D}(t_i) \right]^{-1} \left[\sum_{i=1}^N \mathbf{D}(t_i)^T \mathbf{Cov}_i^{-1} \delta \tilde{\alpha}(t_i) \right]$$

At this point in the development please note that the rank of each of the terms of the form $\mathbf{D}(t_i)^T \mathbf{Cov}_i^{-1} \mathbf{D}(t_i)$ is at most of $\text{rank}(\mathbf{Cov})$. The Jacobian terms usually don't lower the rank as long as the transformation they correspond to is 1-to-1. Therefore if $\text{rank}(\mathbf{Cov}) < \text{length of } \delta \mathbf{z}$, then it is necessary to obtain enough data such that the sum over the data is full rank and the linear system has a unique solution.

The above LSE estimate can also be sequentialized to emulate a Kalman filter method. As is usually the case, the convergence properties of iterative fitting methods such as the first or second order fitting methods depend on the closeness of the initial estimate of the parameters and the goodness of the assumption of linearity used in the Taylor expansion. The Angles-Only orbit determination method involves many transformations and opportunities to test out these assumptions.

6 Application of the second order method

The second order method involves taking the first and second derivatives of Eq.(1) which we write for a general vector of observations and predictions as:

$$\chi^2 = \frac{1}{2} \sum_{i=1}^N [\alpha(t_i) - \tilde{\alpha}_i]^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i]$$

Under the assumption that the covariance matrices are symmetric we can compute the first derivative as

$$\frac{\partial \chi^2}{\partial z_j} = \sum_{i=1}^N \left[\frac{\partial \alpha(t_i)}{\partial z_j} \right]^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i] \quad j = 1, 2, \dots, m,$$

Proceeding to the second derivatives of Eq.(1) we obtain:

$$\frac{\partial^2 \chi^2}{\partial z_j \partial z_k} = \sum_{i=1}^N \left[\frac{\partial^2 \alpha(t_i)}{\partial z_j \partial z_k} \right]^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i] + \left[\frac{\partial \alpha(t_i)}{\partial z_j} \right]^T \text{Cov}_i^{-1} \left[\frac{\partial \alpha(t_i)}{\partial z_k} \right] \quad j, k = 1, 2, \dots, m.$$

We define the Jacobian matrix, \mathbf{J} corresponding to the partial derivatives appearing by

$$J_{ij} = \left[\frac{\partial \alpha(t_i)}{\partial z_j} \right]$$

and the Hessian matrix, \mathbf{H} , of mixed partial derivatives as

$$H_{jk}(t_i) = \frac{\partial^2 \alpha(t_i)}{\partial z_j \partial z_k}$$

With \mathbf{J} and \mathbf{H} defined in this manner the first and second derivatives can be written as

$$\frac{\partial \chi^2}{\partial z_j} = \sum_{i=1}^N J_{ji}^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i] \quad j = 1, 2, \dots, m,$$

and

$$\frac{\partial^2 \chi^2}{\partial z_j \partial z_k} = \sum_{i=1}^N \left[H_{jk}(t_i)^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i] + J_{ji}^T \text{Cov}_i^{-1} J_{ik} \right] \quad j, k = 1, 2, \dots, m.$$

The idea of the second order method is to expand the gradient of χ^2 into a Taylor series and set the resulting components each to 0 and apply Newton-Raphson iterations to the result as follows:

$$\frac{\partial \chi^2}{\partial z_j} \approx \frac{\partial \chi^2}{\partial z_j} \Big|_{\hat{\mathbf{z}}} + \sum_{k=1}^N \frac{\partial^2 \chi^2}{\partial z_j \partial z_k} \Big|_{\hat{\mathbf{z}}} \delta z_k = 0, \quad j = 1, 2, \dots, m,$$

which leads a linear system for the increment vector δz_k

$$\delta z_k = - \sum_{j=1}^m \left[\sum_{i=1}^N [H_{jk}(t_i)^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i] + J_{ji}^T \text{Cov}_i^{-1} J_{ik}] \right]^{-1} \left[\sum_{i=1}^N J_{ji}^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i] \right] \Big|_{\hat{\mathbf{z}}}$$

The above system can be interpreted using matrix multiplication. Let \mathbf{V} be the vector in the parameter space with components

$$V_j = \sum_{i=1}^N J_{ji}^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i]$$

and \mathbf{M} the matrix in the parameter space with components

$$M_{jk} = \sum_{i=1}^N \left[H_{jk}(t_i)^T \text{Cov}_i^{-1} [\alpha(t_i) - \tilde{\alpha}_i] + J_{ji}^T \text{Cov}_i^{-1} J_{ik} \right],$$

Then the solution to the linear system can be written in matrix form as

$$\delta \mathbf{z} = -\mathbf{M}^{-1} \mathbf{V}$$

7 Computation of the Predictions

Assume the data consists of line-of-sight measurements from an observer whose position as a function of time is $\vec{R}_{obs}(t)$. At each of the measurement times t_i , the observer is located at $\vec{R}_i = \vec{R}_{obs}(t_i)$ and makes a measurement of a line-of-sight vector $\mathbf{l}_i = [x_i, y_i, z_i]$ which we assume has been transformed into the Earth-Centered Inertial (ECI) reference system by a suitable transformation from whatever reference frame the measurement was taken in Earth-Centered Earth Fixed (ECEF) or some other typical observation frame. Assuming we have an initial estimate of the object-of-interest's state vector at a reference time t_0 which might be given in position and velocity space or the classical orbital elements or others. We can implement the respective equations of motion to propagate these state vectors as needed to the current time of interest. Obviously whatever state vectors are used as the underlying independent variables, to compute the predicted state which can be compared to the observed data, we must identify the functional form this prediction is to take. As an example, the predicted line-of-sight vector is

$$\vec{l}_i = \vec{R}(t_i) - \vec{R}_i$$

One can translate this vector into more convenient quantities related to the data format e.g., if the range is not measured, then we can work with (Az, El) or θ, ϕ . In terms of spherical polar coordinates this vector's components in ECI are given by

$$\vec{l} = [l_x, l_y, l_z] = l_{\text{mag}} [\sin(\theta) \cos(\phi), \sin(\theta) \sin(\phi), \cos(\theta)],$$

where $l_{\text{mag}} = \sqrt{l_x^2 + l_y^2 + l_z^2}$. The unit vector that describes the line-of-sight direction, is given by

$$\vec{u} = [\sin(\theta) \cos(\phi), \sin(\theta) \sin(\phi), \cos(\theta)]$$

and the relationship between these quantities and the components of \vec{l} are given by

$$\begin{aligned} l_{\text{mag}} &= \sqrt{l_x^2 + l_y^2 + l_z^2} \\ \theta &= \cos^{-1}(l_z/l_{\text{mag}}) \\ \phi &= \text{atan2}(l_y, l_x) \end{aligned} \tag{15}$$

Note that $\text{atan2}(l_y, l_x)$ takes care of the quadrant determination automatically. If the range rate and rates of change in the angles were available, we could make room for those measurements as well in the above list. Some care may still be needed so that directions near $\phi = 0$ and $\phi = 2\pi$ are taken to be close to each other and not far apart. Closeness in phi can be determined by using dot products and cross products followed by a further atan2 to pin down the actual relative phi-angle difference in such cases.

Ultimately we need to work out the Jacobians that will take us all the way back to the propagated state vectors in such a manner that the LSE method can be iterated and the state vector estimate updated. We shall proceed starting with the items corresponding to the above list and work our way back to the relevant state vector by either application of the chain rule, or by direct computation based on the method of backwards differentiation which determines each Jacobian based directly on the functional dependence according to the respective composition of the functions involved and their direct differentiation.

8 Time Development of the Kepler Elements and Perturbation Theory

The State Space approach in the time domain applied to sequential filtering methods is a standard approach. Control Theory and Stochastic Estimation still make use of the State Space model – although there are other alternative approaches currently under development in Control Theory.

Some applications involve ordinary differential equations which might represent some sort of “propagation” in time or perhaps some other independent variable. These differential equations are very often non-linear in the dependent variables and normally one resorts to a linearization method so that standard State-Space methods can be employed – although this needs to be implemented iteratively in the hopes that the initial estimates are “good enough” and that the iteration process converges to the correct solution (if it converges at all).

Let us apply these concepts to the OD problem. The standard State Space representation is in terms of position, velocity and time. The relevant state then incorporates position and velocity which then are envisioned as functions of the independent variable (time).

The typical State Vector, $Y(t)$, representation for the OD problem is therefore of the form

$$Y(t) = \begin{bmatrix} \vec{r}(t) \\ \vec{v}(t) \end{bmatrix}$$

and then the equations of motion can be written as a first order system of differential equations for the time development in the following form:

$$\dot{Y}(t) = \begin{bmatrix} \dot{\vec{r}}(t) \\ \dot{\vec{v}}(t) \end{bmatrix} = \begin{bmatrix} \vec{v}(t) \\ \vec{\ddot{r}}(t) \end{bmatrix}$$

At this point, one usually substitutes for $\vec{\ddot{r}}$ based on Newton’s 2nd Law (assuming constant mass)

$$\vec{F} = m\vec{\ddot{r}}$$

where m is the mass and \vec{F} are the forces acting on the mass. Newton’s 2nd Law can also be extended to non-inertial coordinate systems in the usual manner provided care is taken to adjust the mass terms and forces accordingly.

These adjustments are standard in OD theory where the reference origin is the center of the earth and the position vector is given by relative position between the center of the earth and the sensor (wherever the sensor is located – attached to the earth, or orbiting, etc.).

Another approach to representing the State Vector associated with an orbiting trajectory is to use the classical Kepler elements in the form

$$K(t) = [e(t), a(t), i(t), \omega(t), \Omega(t), M(t)]^T$$

where e is the eccentricity, a is the semi-major axis, i is the inclination, ω is the argument at periapsis, Ω is the longitude of the line of nodes, and M is the mean anomaly and all of these elements are given at time t . In the case of the pure Kepler orbit, then these classical elements are all constant except for the mean anomaly which is a linear function of time.

To apply the state space approach to the OD estimation problem in the presence of perturbations one normally linearizes the equations of motion by expanding them about a reference orbit which is chosen to approximate the true orbit of interest closely but at the same time to have a simpler time development because the perturbations are left out of the time evolution of the reference orbit. This method goes by the name of “chief and deputy” where the chief is the reference orbit and the deputy is the orbit of interest. One works with the relative difference between the chief and deputy. Of course, to get back to the deputy’s orbit with respect to the earth, then the components of the chief are added to the deputy by vector addition. Because a pure Kepler orbit (pure $1/r^2$ gravity with no perturbations) has a simple time development, it can be used as the chief and the time development involves numerically solving Kepler’s equation. Even with a numerical solution of Kepler’s equation in hand, all the subsequently required derivatives present no problem due to application of the Implicit Function Theorem to Kepler’s equations: $M = E - e \sin(E)$ (with E, M, e known numerically).

Regarding the gravitational model, we will work with up-to-J6 zonal harmonics and consider the pros-and-cons of various methods of expanding the equations of motion about a Kepler orbit used as the reference as a relative origin. The up-to-J6 zonal harmonics leads to a gravitational potential of the form

$$\begin{aligned} \Phi = \frac{mu}{r} & \left[1 + \frac{J_2}{2r^2} (1 - 3 \sin^2 \delta) \right. \\ & + \frac{J_3}{2r^3} (3 - 5 \sin^2 \delta) \sin \delta \\ & + \frac{J_4}{8r^4} (3 - 30 \sin^2 \delta + 35 \sin^4 \delta) \\ & + \frac{J_5}{8r^5} (15 - 70 \sin^2 \delta + 63 \sin^4 \delta) \sin \delta \\ & + \frac{J_6}{16r^6} (5 - 105 \sin^2 \delta + 315 \sin^4 \delta - 231 \sin^6 \delta) \\ & \left. + \epsilon \right] \end{aligned} \tag{16}$$

where μ is the gravitational constant, J_i is the coefficient of the i th zonal harmonic, ϵ represents terms of higher order and other missing harmonics (tesseral and sectoral). The distance from the center of the earth is measured in canonical units is r and $\sin \delta = z/r$ where z is the respective coordinate along the North-South axis in the same units as r . If canonical units are not used, then replace r by r/a_e , where a_e is the equatorial radius of the earth.

In the case of the spherical and homogenous earth, the gravitational potential is just

$$V = \frac{\mu}{r}$$

which gives rise to the pure Kepler orbits where the classical elements e, a, i, ω, Ω are constants and M is a linear function of time such that $M = 0$ at periapsis. The mean anomaly for this case is given by $M = n(t - t_p)$ and n is the mean motion, t is the time, and t_p is the clock time at periapsis. The disturbing potential R due to the gravitational perturbations beyond the purely spherical earth is defined as

$$R = \Phi - V$$

Some approaches to approximating the equations of motion make use of binomial expansions and trigonometric series. I believe the most expedient approach is to use Taylor expansions of the full expressions so that the linear term is determined without recombining the binomial and trigonometric series into first order and second order terms. Also implicit differentiation eliminates application of similar expansions of Kepler's equation – so why not try to work with Taylor expansions throughout since only the linear terms are kept in the expansion of the deputy's orbit relative to the chief? Another thing to note is that position and velocity states are rapidly varying and perhaps it is better to describe the trajectory in a representation that is less rapidly varying. On the other hand, perturbations tend to remain small in the position and velocity space representation which might have advantages over other representations where possible singularities might be present in the expansion of the equations of motion.

9 Linearization of the Perturbed Equations of Motion in Position and Velocity Space

Since perturbations are supposed to be “small effects” one approach to linearizing the equations of motion in terms of the Kepler elements is to work out the corresponding effects in position and velocity space and then transform the results over to Kepler space by implementing the appropriate Jacobians.

Since we need to expand the equations of motion about the Kepler chief's orbit we will first examine the situation based on relative position and velocity state vectors and see how they can be transformed over to Kepler classical elements. The disturbing potential is given by R from above. Now the forces which arise from the contribution of R follow in the usual manner by taking gradients:

$$\vec{F}_D = \nabla R$$

Using the standard position and velocity state vector representation $Y(t)$ from above, we have for the equations of motion, the following linear system corresponding to the time rate of change of deputy's state vector as seen from the chief:

$$\dot{Y}_{deputy}(t) = \begin{bmatrix} \vec{r}(t) \\ \vec{v}(t) \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \nabla_x D \\ \nabla_y D \\ \nabla_z D \end{bmatrix}$$

The acceleration due to the D is given by $\vec{a} = \nabla(D)$. We can assume that the initial conditions for this problem are given by the following state vector where the deputy and chief are together at t_0 :

$$Y_{deputy}(t_0) = [0, 0, 0, \nabla_x D, \nabla_y D, \nabla_z D]^T$$

Assuming that small changes in coordinate space are related by Jacobians to small changes in Kepler space we might expect that the accelerations could be expanded about their values at t_0 in a Taylor series corresponding to small changes in coordinate space along the lines:

$$a_i \approx a_i|_{\hat{Z}} + \sum_{j=1}^3 \frac{\partial a_i}{\partial Z_j} |_{\hat{Z}} (Z_j - \hat{Z}_j) = \frac{\partial D}{\partial z_i} |_{\hat{Z}} + \sum_{j=1}^3 \frac{\partial^2 D}{\partial Z_i \partial Z_j} |_{\hat{Z}} (Z_j - \hat{Z}_j),$$

where the sum over j corresponds to a sum over the spatial coordinates $Z_1 = x, Z_2 = y, Z_3 = z$.

If the accelerations are almost constants over a short time period then the velocity of the deputy with respect to the chief would go approximately like

$$\vec{v}(t) \approx \vec{a} \times (t - t_0)$$

Let \mathbf{H} be the 3-by-3 matrix of second derivatives of the disturbing potential with respect to the spatial coordinates, i.e.,

$$H_{ij} = \frac{\partial^2 D}{\partial Z_i \partial Z_j}, \quad i, j = 1, 2, 3$$

Inserting these approximations into the time rate of change of the relative coordinates of the deputy and chief we have approximately

$$\dot{\mathbf{Y}}_{deputy}(t) \approx \begin{bmatrix} \vec{v} - \vec{v}_K \\ \nabla D \end{bmatrix} \approx \begin{bmatrix} \mathbf{0}_3 \\ \nabla D \end{bmatrix}_{\hat{\mathbf{z}}} + \begin{bmatrix} 0 & \mathbf{I}_3 \\ \mathbf{H} & 0 \end{bmatrix}_{\hat{\mathbf{z}}} \begin{bmatrix} \vec{r} - \vec{r}_K \\ \vec{v} - \vec{v}_K \end{bmatrix}$$

The first term on the RHS represents the Inhomogeneous vector term, $\mathbf{I} = [\vec{v} - \vec{v}_K, \nabla D]$, in the expanded equations of motion and the second term on the RHS is the Homogeneous matrix term \mathbf{Q} . The eigenvalues of the Homogeneous term determine the stability properties of the solution of the system of differential equations. If the eigenvalues have real parts which are > 1 then the solution will grow unboundedly at t increases.

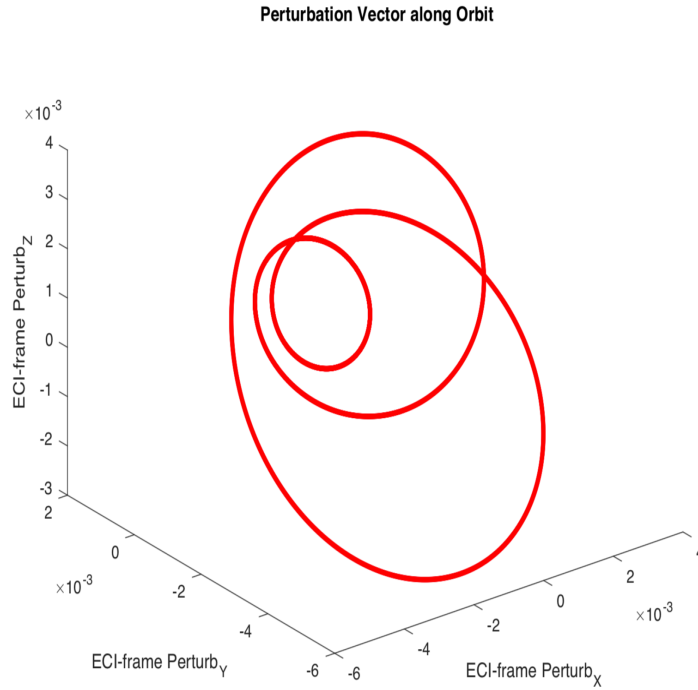


Figure 2: J6 Perturbations vector transported around one orbit

Given that we consider the differences between the components of the chief and the deputy to be small, the relative displacements in position and velocity space and in Kepler space should transform the way vectors transform under change of coordinates:

$$\bar{A}^j = \sum_{k=1}^m \frac{\partial \bar{x}^j}{\partial x^k} A^k$$

Therefore we should be able to use a Jacobian, \mathbf{J} to transform $\Delta \mathbf{Y} = [\Delta \vec{r}, \Delta \vec{v}]$ displacement vectors into to Kepler space displacement vectors as follows

$$\Delta \mathbf{K} = \mathbf{J} \Delta \mathbf{Y} \text{ and the inverse relationship } \Delta \mathbf{Y} = \mathbf{J}^{-1} \Delta \mathbf{K}$$

Therefore the transformed equations of motion in Kepler space obtained by first going through position and velocity space looks like

$$\dot{\mathbf{K}} = \mathbf{J} \dot{\mathbf{Y}} \approx \mathbf{J} \mathbf{I} + \mathbf{J} \mathbf{Q} \mathbf{J}^{-1} \mathbf{K}$$

The transformed Homogeneous term in K space is related by a similarity transformation to \mathbf{Q} . As long as the Jacobian \mathbf{J} is invertible, the eigenvalues of \mathbf{Q} and $\mathbf{J} \mathbf{Q} \mathbf{J}^{-1}$ are identical.

We will test out the above approach to perturbation theory and see how it compares to the canonical approach based on Lagrange's Planetary Equations.

10 Linearization of the Perturbed Equations of Motion in Kepler Space

Canonical perturbation theory in the context of Hamiltonian mechanics proceeds via canonical transformations where Hamilton's equations of motion are invariant. This approach to perturbation theory which has been worked out for the rate of change of the Kepler elements in the presence of gravitational perturbations is given by the Lagrange Planetary Equations which will consider. One thing to bear in mind regarding this approach to perturbation theory is that there are numerical problems as $e \rightarrow 0$ or $\sin(i) \rightarrow 0$.

Let us take our reference orbit (the chief) to be a the trajectory of an object around a spherical earth whose Kepler elements are designated by $[e_K, \alpha_K, I_K, \omega_K, \Omega_K, M_K]$ and we set these equal to the Kepler elements of the deputy at time $t = t_0$. In this case the time rate of change (equations of motion) for the chief around a spherical earth are given by

$$\begin{aligned}\frac{de_K}{dt} &= 0 \\ \frac{da_K}{dt} &= 0 \\ \frac{di_K}{dt} &= 0 \\ \frac{d\omega_K}{dt} &= 0 \\ \frac{d\Omega_K}{dt} &= 0 \\ \frac{dM_K}{dt} &= n\end{aligned}$$

N.B. the n in the reference orbit is the same n appearing in the deputy's orbit at $t = t_0$.

Now moving on to the method of linearization as applied to a trajectory described by Kepler elements in the presence of gravitational perturbations we form the differenced state vector

$$\begin{aligned}\tilde{e} &= e - e_K \\ \tilde{a} &= a - a_K \\ \tilde{i} &= i - i_K \\ \tilde{\omega} &= \omega - \omega_K \\ \tilde{\Omega} &= \Omega - \Omega_K \\ \tilde{M} &= M - M_K\end{aligned}$$

The Lagrange Planetary Equations in terms of the classical elements take the form:

$$\begin{aligned}
\frac{de}{dt} &= \frac{1-e^2}{na^2e} \frac{\partial R}{\partial M} - \frac{\sqrt{1-e^2}}{na^2e} \frac{\partial R}{\partial \omega} \\
\frac{da}{dt} &= \frac{2}{na} \frac{\partial R}{\partial M} \\
\frac{di}{dt} &= \frac{\cos i}{na^2\sqrt{1-e^2} \sin i} \frac{\partial R}{\partial \omega} - \frac{1}{na^2\sqrt{1-e^2} \sin i} \frac{\partial R}{\partial \Omega} \\
\frac{d\omega}{dt} &= \frac{\sqrt{1-e^2}}{na^2e} \frac{\partial R}{\partial e} - \frac{\cos i}{na^2\sqrt{1-e^2} \sin i} \frac{\partial R}{\partial i} \\
\frac{d\Omega}{dt} &= \frac{1}{na^2\sqrt{1-e^2} \sin i} \frac{\partial R}{\partial i} \\
\frac{dM}{dt} &= n - \frac{1-e^2}{na^2e} \frac{\partial R}{\partial e} - \frac{2}{na} \frac{\partial R}{\partial a}
\end{aligned} \tag{17}$$

Observe the terms involving $\sqrt{1-e^2}$ which become imaginary if $e > 1$ as well as the denominators which vanish as $e \rightarrow 0$ and $\sin i \rightarrow 0$. We clearly need to be mindful of these possibilities in any numerical work which searches the parameter space to minimize our objective function. If the minimum lies outside of the physical boundaries for elliptical orbits, then we have to accommodate the possibility that the true orbit is hyperbolic or parabolic. For the moment we will develop the algorithm under the assumption that the conic sections involved are approximately elliptical trajectories.

We can express the RHS of Eq.(17) using matrices. Let us define the following matrices:

$$\begin{aligned}
\mathbf{K} &= [e, a, i, \omega, \Omega, M]^T \\
\mathbf{n} &= [0, 0, 0, 0, 0, n]^T \\
\mathbf{M} &= \begin{bmatrix} 0 & 0 & 0 & -\frac{\sqrt{1-e^2}}{na^2e} & 0 & \frac{1-e^2}{na^2e} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{na} \\ 0 & 0 & 0 & \frac{\cos i}{na^2\sqrt{1-e^2} \sin i} & -\frac{1}{na^2\sqrt{1-e^2} \sin i} & 0 \\ \frac{\sqrt{1-e^2}}{na^2e} & 0 & -\frac{\cos i}{na^2\sqrt{1-e^2} \sin i} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{na^2\sqrt{1-e^2} \sin i} & 0 & 0 & 0 \\ -\frac{1-e^2}{na^2e} & -\frac{2}{na} & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{D} &= \left[\frac{\partial R}{\partial e}, \frac{\partial R}{\partial a}, \frac{\partial R}{\partial i}, \frac{\partial R}{\partial \omega}, \frac{\partial R}{\partial \Omega}, \frac{\partial R}{\partial M} \right]^T
\end{aligned}$$

Notice that the matrix \mathbf{M} is anti-symmetric and therefore has imaginary eigenvalues – one possible consequence of this anti-symmetry it associated contributions to periodic effects in the time dependence of the solution. With these matrices defined as above, Lagrange's Planetary Equations can be written in terms of the following first order differential system of equations:

$$\dot{\mathbf{K}} = \mathbf{n} + \mathbf{MD}$$

We now proceed to linearize this system of equations along the lines that we used with the perturbation equations of motion in position and velocity space:

$$\dot{\mathbf{K}} \approx (\mathbf{n} + \mathbf{MD})_{\hat{\mathbf{z}}} + \sum_{j=1}^m \left[\frac{\partial(\mathbf{n} + \mathbf{MD})}{\partial Z^j} \right]_{\hat{\mathbf{z}}} (Z^j - \hat{Z}^j) \tag{18}$$

The expansion point in Eq.(18) represents the chief's parameterization denoted by \mathbf{K}_C with components \hat{Z}^j . Subtracting the chief's equation of motion

$$\dot{\mathbf{K}}_C = \mathbf{n}$$

from both sides.

$$\mathbf{K}^j - \mathbf{K}_C^j = Z^j - \hat{Z}^j \equiv z^j$$

We have the following differential equation for z^j :

$$\dot{\mathbf{z}} \approx (\mathbf{MD})_{\hat{\mathbf{z}}} + \sum_{j=1}^m \left[\frac{\partial(\mathbf{n} + \mathbf{MD})}{\partial Z^j} \right]_{\hat{\mathbf{z}}} z^j$$

In matrix notation we can express this differential equation as

$$\dot{\mathbf{z}} \approx (\mathbf{MD})_{\hat{\mathbf{z}}} + \left[\frac{\partial(\mathbf{n} + \mathbf{MD})}{\partial \mathbf{Z}} \right]_{\hat{\mathbf{z}}} \mathbf{z} \quad (19)$$

We can classify Eq.(19) as a first order system of differential equations with both an Inhomogeneous term as well as a Homogeneous term. Note that the inhomogeneity arises because the first term on the RHS of Eq.(19) contains the gravitational perturbations evaluated along the reference trajectory of the chief. This term is a 6-by-1 column vector. The reference trajectory itself moves along a pure Kepler orbit and the location of the chief serves as a moving “origin” in this problem which is used to define the relative coordinates of the deputy. The Inhomogeneous term captures the contribution of the disturbing potential as seen at the location of the reference orbit.

Expanding out the terms appearing in Eq.(19) we have:

$$\dot{\mathbf{z}} \approx (\mathbf{MD})_{\hat{\mathbf{z}}} + \left[\frac{\partial \mathbf{n}}{\partial \mathbf{Z}} + \frac{\partial \mathbf{M}}{\partial \mathbf{Z}} \mathbf{D} + \mathbf{M} \frac{\partial \mathbf{D}}{\partial \mathbf{Z}} \right]_{\hat{\mathbf{z}}} \mathbf{z} \quad (20)$$

Recall that for a general matrix product of the form $\mathbf{B} = \mathbf{CD}$, that the columns of \mathbf{B} are linear combinations of the columns of \mathbf{C} with coefficients from the columns of \mathbf{D} . In terms of the components of \mathbf{z} :

$$\left(\frac{\partial \mathbf{M}}{\partial \mathbf{Z}} \mathbf{D} \right) \mathbf{z} = \sum_j \left(\frac{\partial \mathbf{M}}{\partial Z^j} \mathbf{D} \right) z^j$$

The 1st matrix on the RHS $\partial \mathbf{n} / \partial \mathbf{Z}$ is a 6-by-6 matrix all of whose elements are zero except the last row which is given by

$$\left[\frac{\partial n}{\partial Z^1} \mid \frac{\partial n}{\partial Z^2} \mid \frac{\partial n}{\partial Z^3} \mid \frac{\partial n}{\partial Z^4} \mid \frac{\partial n}{\partial Z^5} \mid \frac{\partial n}{\partial Z^6} \right]$$

Using the fact that the columns of a product of two matrices are linear combinations of the columns of the first matrix with coefficients from the columns of the second matrix, we deduce that the j th column of the middle matrix which multiplies \mathbf{z} are given by:

$$\mathbf{C}_j = \left[\frac{\partial \mathbf{M}}{\partial Z^j} \mathbf{D} \right]$$

Therefore we can assemble the product matrix column-by-column in the form

$$\mathbf{C} = \left[\frac{\partial \mathbf{M}}{\partial Z^1} \mathbf{D} \mid \frac{\partial \mathbf{M}}{\partial Z^2} \mathbf{D} \mid \frac{\partial \mathbf{M}}{\partial Z^3} \mathbf{D} \mid \frac{\partial \mathbf{M}}{\partial Z^4} \mathbf{D} \mid \frac{\partial \mathbf{M}}{\partial Z^5} \mathbf{D} \mid \frac{\partial \mathbf{M}}{\partial Z^6} \mathbf{D} \right]$$

The third matrix on the RHS involves the 6-by-6 matrix $\mathbf{M} \frac{\partial \mathbf{D}}{\partial \mathbf{Z}}$ evaluated at the chief's elements. Now \mathbf{D} is already the derivative of the disturbing potential R with respect to $\hat{\mathbf{Z}}$, we have that the derivatives of \mathbf{D} with respect to \mathbf{Z} is the Hessian matrix for R , which we denote by \mathbf{H} . Therefore we have

$$\mathbf{M} \frac{\partial \mathbf{D}}{\partial \mathbf{Z}} = \mathbf{MH}$$

Putting everything together we can write the equations of motion for the linearized Kepler elements with respect to the chief as

$$\dot{\mathbf{z}} \approx (\mathbf{MD})_{\hat{\mathbf{z}}} + \left[\frac{\partial \mathbf{n}}{\partial \mathbf{Z}} + \mathbf{C} + \mathbf{MH} \right]_{\hat{\mathbf{z}}} \mathbf{z}$$

11 Setting Up the Objective Function and Its Derivatives for the Orbit Determination Problem

Assuming we have chose a fit time t_0 and have an initial guess for the State Vector at t_0 , let us use Eq.(1) as the function to be minimized for the purposes of performing a fit for the OD problem. The actual procedure could be by first or second order batch which can be modified to perform a sequential fit similar to a Kalman filter. As mentioned earlier the second order method requires first and second derivatives. If we specialize to the problem of fitting line-of-sight measurements to an orbit, then we have from Eq.(15):

$$\begin{aligned} l_{\text{mag}} &= \sqrt{l_x^2 + l_y^2 + l_z^2} \\ \theta &= \cos^{-1}(l_z/l_{\text{mag}}) \\ \phi &= \text{atan2}(l_y, l_x) \end{aligned}$$

where the components of the line-of-sight vector are given at the i th data record by

$$\vec{l}_i = \vec{R}(t_i) - \vec{R}_i$$

and $\vec{R}(t_i)$ is the predicted position vector of the object of interest at time t_i and \vec{R}_i is the i th sensor location data point measured at time t_i . The function to be minimized in order to perform the fit is

$$\chi^2 = \frac{1}{2} \sum_{i=1}^N [\theta(t_i) - \tilde{\theta}_i, \phi(t_i) - \tilde{\phi}_i]^T \text{Cov}_i^{-1} [\theta(t_i) - \tilde{\theta}_i, \phi(t_i) - \tilde{\phi}_i]$$

The predicted orbit position $\vec{R}(t)$ can be determined from the osculating Kepler elements at time t . Symbolically we write the time-dependent Kepler elements as

$$K(t) = [e(t), a(t), i(t), \omega(t), \Omega(t), M(t)]^t$$

A pure Kepler state vector evolves in time as

$$\text{Kepler}(t) = [e_0, a_0, i_0, \omega_0, \Omega_0, M_0 + n(t - t_0)]^T$$

where the elements with subscript 0 correspond to constants defined for the chief's orbit defined at t_0 , n is the mean motion and t is the current time. To include the effects of the perturbations of gravity we must add the particular and homogeneous solutions $\mathbf{Z}_p(t)$ and $\mathbf{Z}_h(t)$. As the particular solution only depends on the chief's original classical elements we can combine $\text{Kepler}(t)$ with $\mathbf{Z}_p(t)$ to obtain an estimate of the position of the deputy in Kepler space at time t :

$$K(t) = [e_0, a_0, i_0, \omega_0, \Omega_0, M_0 + n(t - t_0)]^T + \mathbf{Z}_p(t) + \mathbf{Z}_h(t)$$

These expressions will involve the propagator function which comes out of the solution to the approximate first order differential system associated with the linear expansion of the equations of motion in the presence of gravitational perturbations.

To proceed with the first or second order fitting methods we need to Taylor expand the vector of predictions $\alpha(t) = [\theta(t), \phi(t)]^T$ about the estimate of the differenced state vector $\hat{\mathbf{Z}}$ at t_0 :

$$\alpha^I(t_0) \approx \alpha^I(t_0)|_{\hat{\mathbf{Z}}} + \sum_{j=1}^3 \frac{\partial \alpha^I(t_0)}{\partial Z_j} |_{\hat{\mathbf{Z}}} (Z_j - \hat{Z}_j)$$

where α^I is the I th component in the measurement vector. The expansion is set up under the assumption that at the first iteration of any fitting method the chief and deputy coincide at t_0 . This initial condition can change as the fit iterations proceed.

By using this linear expansion of the prediction vector the objective function χ^2 is converted into a quadratic form in Z which has a well-defined minimum, i.e., the quadratic form is a multidimensional “bowl”. We can reach the bottom of the bowl in one Newton step. But we still need to iterate because the bowl, as it were, may change from iteration to iteration until, hopefully, the whole procedure converges. This iteration procedure is the strategy of the first order method.

At this point the difficult part of the problem is not so much the details of the minimization equations for a quadratic form as it is organizing all the Jacobians necessary to compute the partial derivative appearing on the RHS of the linearized vector in the χ^2 function. We represent these Jacobians as

$$\frac{\alpha^I(t_i)}{\partial Z^j} = \sum_{r,s,t} \frac{\partial a^I(t_i)}{\partial l^r} \frac{\partial l^r}{\partial R^s} \frac{\partial R^s}{\partial K^t} \frac{\partial K^t}{\partial Z^j} \quad (21)$$

Note that the final partial derivative appearing on the far RHS is the Fundamental Matrix $\Phi(t - t_0)$ (the Propagator function for the first order system of differential equations).

One of the nice things about the Backwards Differentiation method is that all of the Jacobians are computed directly by differentiation of the final function. Rather than compute each of the partial derivatives that appear in Eq.(21) we can apply the method of Backwards Differentiation to the algorithm that computes $\alpha(t)$ and obtain the derivatives on the LHS directly. The method of Backwards Differentiation is described below.

12 Taking derivatives by the method of Backwards Differentiation

A computer algorithm is embodied in a list of assignment statements. It can be thought of as an ordered list of recursions where each subsequent line in the list is a function of the preceding lines whose values have been stored and are retrievable when needed later. Taking derivatives of the subsequent quantities can be done line at a time. There are two methods that one can consider – forwards differentiation where one starts at the beginning of the list and works down towards the end or starts at the bottom (assuming all the intermediate quantities are retrievable from storage) and works back towards the beginning of the list. Let us set up the framework in which to define both methods of differentiation and how they compare.

Consider an ordered list of r recursions

$$\begin{aligned} 1. & \ h_1 = h_1(\Omega_1) \\ 2. & \ h_2 = h_2(\Omega_2) \\ & \vdots \\ r. & \ h_r = h_r(\Omega_r) \end{aligned}$$

where Ω_k represents the set of all previously computed quantities and the parameters, \mathbf{x} which the algorithm takes as input. Now for the k th line we can express this set of parameters as a union of all the previously computed items: $\Omega_k \subseteq \mathbf{x} \cup \{h_i : i < k\}$ together with the list of parameters \mathbf{x} . If there are n such parameters we can implicitly include them in the list of recursions symbolically as $h_1 = x_1, h_2 = x_2, \dots, h_n = x_n$, so we may assume $\Omega_k \subseteq \{h_i : i < k\}$. The objective is to compute the derivatives of the final assignment statement (or any of its predecessors) in terms of the parameters list, i.e., we wish to have a general method to compute $\partial h_j / \partial \mathbf{x}$ for any $j \leq r$.

The method of Forwards Differentiation proceeds by a straightforward application of the chain rule as follows:

$$\begin{aligned}
1. \quad h_1 &= h_1(\Omega_1), \quad \frac{\partial h_1}{\partial \mathbf{x}} = \sum_{h_i \in \Omega_1} \frac{\partial h_1}{\partial h_i} \frac{\partial h_i}{\partial \mathbf{x}} \\
2. \quad h_2 &= h_2(\Omega_2), \quad \frac{\partial h_2}{\partial \mathbf{x}} = \sum_{h_i \in \Omega_2} \frac{\partial h_2}{\partial h_i} \frac{\partial h_i}{\partial \mathbf{x}} \\
3. \quad h_3 &= h_3(\Omega_3), \quad \frac{\partial h_3}{\partial \mathbf{x}} = \sum_{h_i \in \Omega_3} \frac{\partial h_3}{\partial h_i} \frac{\partial h_i}{\partial \mathbf{x}} \\
&\vdots \\
r. \quad h_r &= h_r(\Omega_r), \quad \frac{\partial h_r}{\partial \mathbf{x}} = \sum_{h_i \in \Omega_r} \frac{\partial h_r}{\partial h_i} \frac{\partial h_i}{\partial \mathbf{x}}
\end{aligned}$$

By extension second derivatives are found similarly. It should be pointed out here that the process of gathering up the full set of derivatives with respect to all the parameters in the list \mathbf{x} requires n forward passes through the above differentiation algorithm. This is the nature of Forwards Differentiation.

The method of Backwards Differentiation or Adjoint Differentiation proceeds by implementing a work array whose length is the same as the list of recursions including the parameters and starting at the bottom of the list and working upwards towards the top. The rules for Backwards Differentiation are captured in the following:

1. Initialize $F_j = 0$, for $i < r$ and set $F(r) = 1$
2. $F_i = F_i + F_j(\partial h_k / \partial h_j)$ for all i such that $h_j \in \Omega_k$, $k = r, r-1, \dots, n+1$
3. Then $\partial h_r / \partial h_i = F_i$, $i = 1, 2, \dots, r$.

We need to determine i and k going backward and compute $\partial h_k / \partial h_i$ dynamically from h_1, h_2, \dots, h_r . There are also various tools available to implement such an algorithm automatically such as ADIFOR for FORTRAN and AutoDiff for MATLAB, but for the projects that I have applied this method to, the number of lines of code is not prohibitive to code the above algorithm “by hand.” The main point of Backwards Differentiation is that the work vector F stores the intermediate results in an efficient manner and would otherwise require a large amount of computation to expand the whole differentiation procedure using formulas for each derivative. Also, one backwards pass through the algorithm produces the entire gradient of h_r (or any other intermediate computation).

Second derivatives can be obtained by applying backwards differentiation twice, or by applying backwards differentiation and forwards differentiation in either order. Three work vectors each of length r (the length of the list of recursions in the algorithm) are required. Let \mathbf{F} be provided by the backwards differentiation algorithm above to compute all the derivatives of h_r . Let \mathbf{Q} and \mathbf{S} each be arrays also of length r . Then second derivatives can be computed using the following algorithm:

1. Initialize $Q_v = 1, Q_j = 0$ and $v \in \{1, 2, \dots, n\}$ and $S_j = 0, \forall j = 1, 2, \dots, r$
2. $Q_k = Q_k + Q_i(\partial h_k / \partial h_i), h \in \Omega_k$;
 $S_j = S_j + Q_i F_k(\partial^2 h_k / \partial h_i \partial h_j), h_i, h_k \in \Omega_k, k = n+1, n+2, \dots, r$
3. $S_j = S_j + S_k(\partial h_k / \partial h_i), h_j \in \Omega_k, k = r, r-1, \dots, n+1$
4. Then $\partial^2 h_r / \partial x_v \partial x_j = S_i, i = 1, 2, \dots, n$

Note that \mathbf{Q} is calculated by the forward differentiation equations and Step 3 is the first derivatives by backwards differentiation algorithm. This is an example of applying forwards differentiation to the backwards differentiation

Example $f(x,y)=(x+y)(x-y)$

Ordered list	x=2 y=1	Backward Differentiation				
		start	k=5	k=4	k=3	end
$h_1=x$	$h_1=2$	$F_1=0$		+3	+1	= 4
$h_2=y$	$h_2=1$	$F_2=0$		-3	+1	=-2
$h_3=h_1+h_2$	$h_3=3$	$F_3=0$	+1			= 1
$h_4=h_1-h_2$	$h_4=1$	$F_4=0$	+3			= 3
$h_5=h_3h_4$	$h_5=3$	$F_5=1$				= 1

where $r=5, n=2$

$$\partial f(x,y)/\partial x=4$$

$$\partial f(x,y)/\partial y=-2$$

Figure 3: Example of Backwards Differentiation

algorithm in order to calculate the 2nd derivatives of h_r with respect to the parameter set \mathbf{x} . As mentioned, one can also apply backwards differentiation twice or forwards differentiation followed by backwards differentiation. The above keeps the bookkeeping aligned in the sense that all three work vectors \mathbf{F} , \mathbf{Q} and \mathbf{S} have the same length.

13 Time Considerations

Consider the case we are evaluating a vector-valued function $\partial \mathbf{f}(h_1, h_2, \dots, h_r)/\partial \mathbf{x}$ where $\dim(\mathbf{f}) = m$ and $\dim(\mathbf{x}) = n$. Let Ψ equal the computational work need to evaluate h_1, h_2, \dots, h_r .

Then forward differentiation requires an amount of computational work $\propto \Psi$ for each first and second derivative (assuming $m = 1$): n first derivatives use $o(n\Psi)$ and $n^2\Psi$ amount of work respectively. Nevertheless forward differentiation is preferred if m is large and n is small.

Backwards differentiation is preferred when n is large and m is small. Evaluating all n first derivatives uses only work $\propto \Psi$ (assuming $m = 1$). All $n^2/2$ second derivatives uses work $\propto n\Psi$.

14 Initialization of Orbit Determination

The fitting methods described above require a good initial guess of the orbit and, because the prediction functions involved are non-linear in terms of the underlying parameters, proceed by iteration beginning with the initial guess.

There are several classical orbit determination methods based on Angles-Only data: Laplace method of successive differentiation, Gauss's method, Escobal's Double-R Iteration method, etc. In the case of a single orbiting sensor there are severe limitations for short periods of observation due to the large number of completely different ballistic trajectories that match the Line-of-Sight data very closely. It is necessary to observe enough trajectory curvature in the orbit in order to single out the orbits which are good candidate initial states. Data from two orbiting sensors greatly improves initial state estimation.

Let us set up a time-dependent polynomial fit for the purposes of making a good initial state orbit estimation. The following represents a linear model for the m -vector polynomial

$$Y(t) = X(t)\beta + \epsilon.$$

The m -by- $n+1$ incidence matrix $X(t)$ is given by

$$X(t) = [\mathbf{I} \mid \mathbf{t}\mathbf{I} \mid \mathbf{t}^2\mathbf{I} \mid \mathbf{t}^3\mathbf{I} \mid \dots \mid \mathbf{t}^n\mathbf{I}]$$

and I is the m -by- m Identity matrix and ϵ is the uncertainty in the linear model.

This method can be extended in a manner tailored for line of sight data and satisfy Newton's 2nd law. Because line of sight measurements are two dimensional the difference between the observed and predicted line of sight unit vectors needs to be projected into the perpendicular plane in order to compare with the estimated measurement error. Gravitation, at least up to $1/r^2$ contributions can be used as a constraint via Lagrange multipliers. In principal it is probably immaterial whether one minimizes the projected spatial vector difference or actually the projected unit-vector difference, since the minimums are closely related except for scaling and the purpose is just to estimate the initial state. As an example the spatial difference between the polynomial at time t and the sensor measurement location S and line of sight measured unit vector \vec{u} is given by the difference between the polynomial and the closest position on the ray extending from the sensor towards the polynomial along the line of sight, i.e., $S - u^T(S - Y)$ or

$$\vec{r} = S - u^T(S - Y(t)) - Y(t) = (I - u^T)(S - Y(t))$$

and a χ^2 can be constructed for this problem as follows

$$\chi^2 = \sum_{i=1}^N [S_i - Y(t_i)]^T [1 - u_i^T]^T [I - u_i^T] [S_i - Y(t_i)] + \lambda^T [\ddot{Y}(t_i) + \mu \frac{Y(t_i)}{|Y|^3}],$$

Since the line of sight data is 2-d the $[I - u^T]$ matrix can be replaced by a 3-x-2 projection matrix that take 3-vectors and projects them into the plane perpendicular to the line of sight measurement. Both

$$\mathbf{M}_{UV}(i) = \begin{bmatrix} \cos \theta_i \cos \phi_i & \cos \theta_i \sin \phi_i & -\sin \theta_i \\ -\sin \phi_i & \cos \phi_i & 0 \end{bmatrix}$$

Now \mathbf{M}_{UV} and $[I - u^T]$ are both rank-2 projection matrices which project spatial differences into the plane perpendicular to the line-of-sight direction and satisfy the following relationship:

$$\mathbf{M}_{UV}\mathbf{M}_{UV}^T = [I - u^T][I - u^T]^T$$

Also the covariance matrix for the line-of-sight measurements are conformable with \mathbf{M}_{UV} and we can insert them also into the objective function in place of $[I - u^T]$ as follows:

$$\tilde{\chi}^2 = \sum_{i=1}^N [S_i - Y(t_i)]^T \mathbf{M}_{UV}^T \mathbf{Cov}_{UV}^{-1} \mathbf{M}_{UV} [S_i - Y(t_i)] + \lambda^T [\ddot{Y}(t_i) + \mu \frac{Y(t_i)}{|Y|^3}],$$

which differs from the usual expression in that the displacements are not necessary with respect to unit vector line-of-sight displacements on the unit sphere. In practice that is somewhat immaterial to the minimization scheme and can always be put back in to calculate the fitted covariance of the β parameters.

We include here a description of Laplace's method of orbit determination which can be found in many books[12] and papers [13]. Laplace's technique requires a polynomial approximation to a sequence of 3 - d vectors which can be obtained by using linear regression to fit the vector functional form using a linear model[15].

$$\mathbf{y}(t) = \mathbf{M}(t) \beta$$

where the *design matrix* $\mathbf{M}(t) = [\mathbf{X}, \mathbf{X}t, \mathbf{X}t^2, \dots, \mathbf{X}t^{n-1}]^T$ is the 3-by-3n unit matrix (in the case of 3-d vector polynomials $\mathbf{X} = \mathbf{I}_3$ and β is a 3n-by-1 vector of *regression parameters*. The Least Squares estimate of the regression parameters given $i = 1, 2, \dots, N$ data points is

$$\hat{\beta} = \left[\sum_{i=1}^N \mathbf{M}^T(t_i) \mathbf{M}(t_i) \right]^{-1} \left[\sum_{i=1}^N \mathbf{M}^T(t_i) \mathbf{y}(t_i) \right]$$

The above method can be used to fit vector polynomials to sensor positions and line-of-sight vectors so that approximations can be made to the vector as well as its first and second derivatives with respect to time which are needed to apply Laplace's method.

Let us assume that we have at least three line-of-sight unit vector measured from the moving sensor (following the description in [12])

$$\mathbf{L}_i = \begin{bmatrix} L_I \\ L_J \\ L_K \end{bmatrix} = \begin{bmatrix} \sin(\theta_i) \cos(\phi_i) \\ \sin(\theta_i) \sin(\phi_i) \\ \cos(\theta_i) \end{bmatrix} = \mathbf{M}_{\mathbf{UV}}(i) \quad i = 1, 2, \dots, N \quad \text{assume } N \geq 3$$

Laplace's technique involves writing the slant range vector $\vec{\rho} = \rho \mathbf{L}$ in terms of the line-of-sight vector from the sensor to the target and then adding the sensor location with respect to the origin at the center of the earth \mathbf{R} to obtain the vector location of the target with respect to the origin in the form $\mathbf{r} = \rho \mathbf{L} + \mathbf{R}$ and then using the polynomial fits to \vec{L} and \vec{R} and their derivatives to express the first and second derivatives of \vec{r} in the form

$$\begin{aligned} \mathbf{r} &= \rho \mathbf{L} + \mathbf{R} \\ \dot{\mathbf{r}} &= \dot{\rho} \mathbf{L} + \rho \dot{\mathbf{L}} + \dot{\mathbf{R}} \\ \ddot{\mathbf{r}} &= 2\dot{\rho} \dot{\mathbf{L}} + \ddot{\rho} \mathbf{L} + \rho \ddot{\mathbf{L}} + \ddot{\mathbf{R}} \end{aligned}$$

At this point Laplace's method uses an approximation for $\ddot{\mathbf{r}}$ and Newton's 2nd Law to insert the $1/r^2$ approximate gravitational force law into the above system so that it can be written in the form of a linear system which can be used to set up an 8th order polynomial to solve for the distance of the target to the origin. After that equation is solved numerically, then the result can be back-substituted into the vector relationships to estimate the targets position and velocity at a particular time. That becomes the initial estimate for the least squares fitting method to use to start the iterations. This state vector can be transformed into Kepler classical elements or any other set of elements desired.

15 Line of Sight Displacements

Using spherical polar coordinates we define a position on the unit sphere in terms of θ and ϕ as

$$\mathbf{r} = [\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta]$$

We can define two tangent vectors to the unit sphere at $[\theta, \phi]$ by taking the derivative of the position vector with respect to the coordinates θ and ϕ . Let \mathbf{u} be the tangent vector in the θ direction and \mathbf{v} be the tangent vector in the ϕ direction. By taking derivatives we have that these vectors are proportional to

$$\begin{aligned} \mathbf{u} &\propto [\cos \theta \cos \phi, \cos \theta \sin \phi, -\sin \theta] \\ \mathbf{v} &\propto [-\sin \theta \sin \phi, \sin \theta \cos \phi, 0] \end{aligned}$$

We will define \mathbf{u} and \mathbf{v} to be the respective unit vectors in the above directions:

$$\begin{aligned} \mathbf{u} &= [\cos \theta \cos \phi, \cos \theta \sin \phi, -\sin \theta] \\ \mathbf{v} &= [-\sin \theta \sin \phi, \sin \theta \cos \phi, 0] \end{aligned}$$

The three vectors $\{\mathbf{r}, \mathbf{u}, \mathbf{v}\}$ form an orthonormal set of vectors with \mathbf{r} being along the line of sight and \mathbf{u}, \mathbf{v} spanning directions in the plane perpendicular to the line of sight. Coordinates in the perpendicular plane represent angular differences between two line of sight vectors and there are several ways of parameterizing these angular differences. For example we can project the vector differences between two line of sight vectors along \mathbf{u} and \mathbf{v} by taking dot products of those vector differences with these unit vectors. Alternatively we can take the dot product directly between the line of sight vectors and use polar coordinates where the radius represents the magnitude of the angular deflection. We can also use the angular differences $\Delta\theta$ and $\Delta\phi$, and we can work out expressions for the projections

of the line of sight vectors in terms of their components using trigonometric identities.

Consider two line of sight vectors \mathbf{l}_1 and \mathbf{l}_2 and consider their vector difference in coordinate space:

$$\Delta \mathbf{l} = \mathbf{l}_1 - \mathbf{l}_2$$

If we take the dot product with the components perpendicular to, say \mathbf{l}_1 can be expressed as

$$\begin{aligned}\Delta u &= \Delta \mathbf{l} \cdot \mathbf{u} = \Delta l_x \cos \theta \cos \phi + \Delta l_y \cos \theta \sin \phi - \Delta l_z \sin \theta \\ \Delta v &= \Delta \mathbf{l} \cdot \mathbf{v} = -\Delta l_x \sin \phi + \Delta l_y \cos \phi\end{aligned}$$

This set of relations can be cast in matrix form as

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi & \cos \theta \sin \phi & -\sin \theta \\ -\sin \phi & \cos \phi & 0 \end{bmatrix} \begin{bmatrix} \Delta l_x \\ \Delta l_y \\ \Delta l_z \end{bmatrix} \quad (22)$$

which can be viewed as a transformation from a 3-dimensional space to a 2-dimensional space. The Jacobian of such a transformation has rank 2. Another way to express $[\Delta u, \Delta v]$ is in terms of the spherical polar coordinates of \mathbf{l}_1 and \mathbf{l}_2 :

$$\begin{aligned}\mathbf{l}_1 &= [\sin \theta_1 \cos \phi_1, \sin \theta_1 \sin \phi_1, \cos \theta_1] \\ \mathbf{l}_2 &= [\sin \theta_2 \cos \phi_2, \sin \theta_2 \sin \phi_2, \cos \theta_2]\end{aligned}$$

Let \mathbf{l}_1 define the line of sight direction and the coordinates u, v will lie in the plane defined perpendicular to \mathbf{l}_1 . Form the difference vector as

$$\Delta \mathbf{l} = \begin{bmatrix} \sin \theta_1 \cos \phi_1 - \sin \theta_2 \cos \phi_2 \\ \sin \theta_1 \sin \phi_1 - \sin \theta_2 \sin \phi_2 \\ \cos \theta_1 - \cos \theta_2 \end{bmatrix}$$

Taking the dot product with \mathbf{u} and \mathbf{v} defined in the perpendicular plane of \mathbf{l}_1 we have

$$\begin{aligned}\Delta u &= \Delta \mathbf{l} \cdot \mathbf{u} = (\sin \theta_1 \cos \phi_1 - \sin \theta_2 \cos \phi_2) \cos \theta_1 \cos \phi_1 \\ &\quad + (\sin \theta_1 \sin \phi_1 - \sin \theta_2 \sin \phi_2) \cos \theta_1 \sin \phi_1 \\ &\quad - (\cos \theta_1 - \cos \theta_2) \sin \theta_1\end{aligned}$$

$$\begin{aligned}\Delta v &= \Delta \mathbf{l} \cdot \mathbf{v} = -(\sin \theta_1 \cos \phi_1 - \sin \theta_2 \cos \phi_2) \sin \phi_1 \\ &\quad + (\sin \theta_1 \sin \phi_1 - \sin \theta_2 \sin \phi_2) \cos \phi_1\end{aligned}$$

Using trigonometry the above relationship between Δu and Δv in terms of the angles defining the line of sight directions reduce to

$$\begin{aligned}\Delta u &= \sin \theta_1 \cos \theta_2 - \sin \theta_2 \cos \theta_1 \cos(\phi_1 - \phi_2) \\ \Delta v &= \sin \theta_2 \sin(\phi_1 - \phi_2)\end{aligned} \quad (23)$$

Another representation that might be useful is to normalize the $[\Delta u, \Delta v]$ displacements so that each of the variables is bounded between -1 and 1 . As an example we could consider:

$$\begin{aligned}\Delta \tilde{u} &= \sin \theta_1 \cos \theta_2 - \sin \theta_2 \cos \theta_1 \cos(\phi_1 - \phi_2) \\ \Delta \tilde{v} &= \sin(\phi_1 - \phi_2)\end{aligned} \quad (24)$$

In this case the displacements lie in the rectangle $-1 \leq \Delta \tilde{u} \leq 1$ and $-1 \leq \Delta \tilde{v} \leq 1$. We have several representations Eq.(22), Eq.(23), and Eq.(24) with which to express the displacements in the perpendicular plane with respect to \mathbf{l}_1 . The first relates the (x,y,z) components of the line of sight difference vector and a transformation Jacobian to arrive at $[\Delta u, \Delta v]$. The second relates the spherical polar angles of \mathbf{l}_1 and \mathbf{l}_2 to the differences $[\Delta u, \Delta v]$ as seen in the perpendicular plane and the third is a modified version of the second such that the possible displacements cover a fixed rectangular region.

The choice of which displacement representation to use depends on the details of the measurements and their covariance structure and how it fits in with the objective function, χ^2 , to be minimized in the orbit determination method.

References

- [1] Paul Avery, “Applied Fitting Theory I: Generalized Least Squares Theory, CBX-91-72”
- [2] O.I. Dahl, T.B. Day, F.T. Solmitz, N. L. Gould, “Group A Programming Note No. P-126”
- [3] Stephen P. Smith, “The Cholesky Decomposition and Its Derivatives”
- [4] Escobal, “Methods of Orbit Determination”
- [5] Brouwer and Clemence, “Celestial Mechanics”
- [6] Tapley, Schutz, Born, “Statistical Orbit Determination”
- [7] Griewank, *et al.*, “Computational Differentiation”
- [8] John Georganas, William Kovalcik, “Angle Sensor Tracking Concepts”
- [9] John Georganas, “Converted Specific Angular Momentum Measurements Estimator”
- [10] Elizabeth M. Keil, “Kalman Filter Implementation to Determining Orbit and Attitude of a Satellite in a Molniya Orbit”
- [11] David R. Bizup, Donald E. Brown, “The Over-Extended Kalman Filter – Don’t Use It!”
- [12] Roger R. Bate, Donald D. Mueller, Jerry E. White, “Fundamentals of Astrodynamics”, Dover 1971, pp. 117-122
- [13] P.O. Bell, “Orbit Determination by Angular Measurements”
- [14] A.J.Claus, R.B. Blackman, E.G. Halline, W.C. Riggway, III, “Orbit Determination and Prediction and Computer Programs”
- [15] Alvin C. Rencher and G. Bruce Schaalje, “Linear Models in Statistics”