Variational Equations for Orbit Determination by Differential Correction

Dennis Milbert, Ph.D.
Chief Geodesist, National Geodetic Survey, NOAA, Retired
Gaithersburg, Md.

Draft: October 9, 2023

ABSTRACT

In the past decade it has been claimed that the standard application of variational equations in orbit determination (OD), as well as in other applications, has neither a sound mathematical nor physical basis. This argument is numerically and mathematically tested. The computational tests use standard processing of perfect, synthetic data derived from exact analytic models. An iterated least squares OD procedure involving force models, variational equations, and numerical differential equation integration is initialized at an approximate, perturbed solution. The converged least squares solution is compared to the defining parameters of the ideal solution to test claims against validity. Four scenarios are examined: simple 2-D ballistic motion, 3-D central force (Keplerian) motion, a simple and a forced harmonic oscillator. Each scenario is varied by including solution for unknown force model parameters. It is found in all cases that the standard application of variational equations in OD, including the zero initial condition of the sensitivity matrix, does, in fact, recover scenario and variant parameter sets to the limits of machine precision. A further test, where an element of the sensitivity matrix initial conditions is set non-zero, produced a decidedly detrimental effect on convergence to the correct defining parameters. Mathematically, exact analytic solutions are derived for the state transition (STM) and sensitivity matrices for both oscillator scenarios, confirming the variational initial values used in standard OD practice. The correct, general mathematical derivation of the variational equations and initial values, made by other authors, is displayed, and proves that standard OD practice has a sound mathematical basis.

1. Introduction

Xu (2018) claims that the standard application of variational equations in orbit determination (OD), as well as in other applications, has neither a sound mathematical nor physical basis. This broad assertion specifies mathematics and statistics, chemistry and physics, and satellite gravimetry. Such statements invite, if not require, computational testing and mathematical review to determine if such claims are, in fact, valid.

It must be emphasized that no new theory is developed in this study. Quite the contrary. The *existing* theory of differential correction (DC) is inspected, implemented, and tested in the most rigorous fashion possible, to the limits of computational precision.

In the mathematical review portions of this study (Sections 11, 13, 14, 15), expressions for the exact analytic solutions for the state transition (STM) and sensitivity matrices for simple and forced harmonic oscillators are derived. These expressions *might* be new; but given the vast body of literature addressing harmonic oscillators in mathematics, physics, and engineering, it is unlikely the expressions are new. Further, a general derivation of initial conditions for columns of the sensitivity matrix has been located, and is related in Section 11.

Four simple scenarios are examined. And each, in turn, is varied by the option of estimating force model parameters. The critical aspect of these scenarios is that they all possess exact analytic solutions. Hence, the exact dynamics of the idealized satellite or oscillator are defined and known at all points in time. Further, perfect synthetic data can be computed. These perfect data then feed the existing DC procedures, under the situation of imperfect knowledge of the state vector, to see if and how well the original, idealized state vector can be recovered. The key is that even these four simplified scenarios and their variants have sufficient complexity to require the implementation of variational equations in the DC. Any shortcomings in the mathematics (not to mention software "bugs") of the variational equations become immediately evident in the results.

In this study this author reviews the broad aspects of DC as an instance of nonlinear least squares computation. Differential equation (DE) integration is given a quick sketch, since no assertions have been raised regarding integration procedures. Variational equations and existing DC practice is inspected. Then, the scenarios and variants are defined, with special detail on the parameters, the force models, the variational equations, and the initial conditions. The results of the computations include comparison against the original, defining parameters. As described above, Sections 14 and 15 mathematically dissect the simple and forced harmonic oscillators. And, Section 11 displays derivations of the general case of initial conditions for variational equations. The study ends with discussion and conclusions.

2. Nonlinear Least Squares

Differential correction (DC) is a term used in orbit determination (OD) to refer to a batch, iterated, nonlinear weighted least squares solution. This is in marked contrast to sequential estimation by Kalman filters (KF) or extended Kalman filters (EKF). The DC mathematical setup is broad enough to encompass differential equations (DE) of force models.

The necessary ingredients for iterated least squares are n measured observations, $\mathbf{L}_b$, as an n x 1 column vector; an associated observation dispersion matrix, $\mathbf{D}$, as an n x n matrix (also known as a variance-covariance matrix); a derived weight matrix, $\mathbf{W}$, as an n x n matrix, where $\mathbf{W} = \mathbf{D}^{-1}$; an observation model vector, $\mathbf{F}(\mathbf{X}_a)$, as an n x 1 column vector of functions; which are evaluated by iteratively refined estimates of u unknown parameters, $\mathbf{X}_a$, as a u x 1 column vector; and, finally, starting estimates of the unknown parameters, $\mathbf{X}_0$, as a u x 1 column vector. (Note: the subscript of $\mathbf{L}_b$ denotes the German word "beobachtung", which means "observation".) We assume the observation set is sufficient to resolve the model parameters. As a minimum, $n \geq u$.

Due to measurement error and observation model shortcomings, we expect misfits between the observations and observation models. The misfit is denoted by an observation residual vector, $\mathbf{V} = \mathbf{F}(\mathbf{X}_a) - \mathbf{L}_b$, an n x 1 column vector. One may choose any set of values for $\mathbf{X}_a$ and get an associated set of values for $\mathbf{V}$. To resolve this multiplicity, a condition is imposed to minimize the scalar: $\lambda = \mathbf{V}^t\mathbf{W}\mathbf{V}$. The minimized scalar, λ, is the sum of squares of weighted residuals. Hence the name "least squares."

The derivation of the least squares "recipe" is not given in this study. It can be found in numerous texts, such as Ghilani and Wolf (2006) and Mikhail (1976). Suffice to say that when the observation parametric models, $\mathbf{F}(\mathbf{X}_a)$, are nonlinear, the derivation truncates a series expansion. This leads to an improved, but not ideal, estimate of the parameters, $\mathbf{X}_a$. The truncation also generates a requirement to have starting estimates of the parameters, $\mathbf{X}_0$, and the requirement to iterate the solution to convergence.

Given the ingredients detailed above, the iterated least squares process is:

$\mathbf{L}_0 = \mathbf{F}(\mathbf{X}_0)$	computed observations, n x 1
$\mathbf{L} = \mathbf{L}_0 - \mathbf{L}_b$	observation misclosures, n x 1
$\mathbf{A} = \partial\mathbf{F}/\partial\mathbf{X}$	design matrix (Jacobian), n x u, evaluated with $\mathbf{X}_0$
$\mathbf{N} = \mathbf{A}^t\mathbf{W}\mathbf{A}$	normal equations, left hand, u x u
$\mathbf{U} = \mathbf{A}^t\mathbf{W}\mathbf{L}$	normal equations, right hand, u x 1
$\mathbf{X} = -\mathbf{N}^{-1}\mathbf{U}$	corrections to unknowns, u x 1
$\mathbf{X}_a = \mathbf{X}_0 + \mathbf{X}$	adjusted unknowns, u x 1

If the solution $\mathbf{X}_a$ has not converged, overwrite $\mathbf{X}_0$ with $\mathbf{X}_a$ and repeat the process. If the starting estimates, $\mathbf{X}_0$, are sufficiently near the stationary point, $\mathbf{X}_a$, then one may expect convergence. The signs above must be rigorously followed. A sign error can lead to divergence, rather than convergence. If the iterated series of $\mathbf{X}_a$ get sufficiently close to the stationary point, then the solution may actually display quadratic convergence. Of course, at some stage this will be limited by the number of bits that represent a floating-point number.

It should come as no surprise that if one wants the highest accuracy, then iteration should always be an option in the DC process. However, it must be recognized that there are a variety of OD applications; some of which have less stringent requirements. One example is in the surveillance and monitoring efforts that lead to the creation and regular updating of the two-line element set (TLE). The formats there only support accuracies of 6-35 m (Vallado, 2013, section 2.4.2). One should expect a variety of DC implementation procedures in the real world.

One of the rules of least squares is that to obtain the highest accuracy of adjusted results, one should use the best possible observation models, $\mathbf{F}(\mathbf{X}_0)$. As seen above, quality observation models will lead to quality misclosures, $\mathbf{L}$. The least squares projection matrix, $(\mathbf{A}^t\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^t\mathbf{W}$, will project the misclosures into the corrections to the unknowns, $\mathbf{X}$. And these lead to quality adjusted unknowns, $\mathbf{X}_a$.

In the case of DC, the DEs of the satellite force models will need to be integrated to obtain state vectors at all observation times, *t*. To get higher accuracy here, one must have the best possible force models, and the best possible integration software. Figures 4.5, 4.6, 4.8, and 4.9 in Montenbruck and Gill (2000) compare a variety of integrators for achievable accuracy and computational effort in terms of number of function calls. These show upper bounds of 13 decimal digits.

These considerations must be tempered by the fact that the misclosures, $\mathbf{L}$, are also formed from the observations, $\mathbf{L}_b$. And the best modeling in the world will not help with observations or orbital configurations of indifferent quality. The least squares process above shows that there are *two* targets, $\mathbf{L}_0$ and $\mathbf{L}_b$, in the quest for greater accuracies.

For the reasons of real-world accuracy requirements and available observation accuracy, one will see approximations in force models in DC practice. Figure 3.1 (ibid., pg. 55) is particularly instructive. It shows typical force model perturbations plotted on a logarithmic scale against satellite orbit radius about the geocenter. The figure quantitatively illustrates some physical effects, such as atmospheric drag and certain geopotential coefficients, which become negligible at geosynchronous altitudes.

A companion rule of least squares is that one need not do the best possible job in computing the design matrix (Jacobian), $\mathbf{A}$, and yet one may still iterate to the correct stationary point. Performance may degrade to super-linear or even linear convergence (or worse); increasing the number of iterations. But the correct solution is ultimately obtained. Of course, one can not populate $\mathbf{A}$ with random numbers. This aspect of DC gets considerable attention because an exact $\mathbf{A}$ may be expensive to compute. This is highlighted in the following quote (ibid., pg. 242):

"Since accuracy requirements for the partial derivatives are generally more relaxed than that for the trajectory itself, it is common to apply a simplified force model in the solution of the variational equations."

(Here, variational equations refer to auxiliary DEs that are not part of the state vector force model DEs, but which are needed to compute the partials in **A**.)

3. Differential Equation Integration

One of the unique aspects of DC is that, in general, one does not have an exact analytic expression for the satellite trajectory. Rather, the satellite position, **r**, and velocity, $\dot{\mathbf{r}}$, is described by DEs for one or more force models. These force models describe the acceleration of the satellite, **a**, caused by the forces. Thus, one does not only need a matrix computation subsystem for the least squares problem, but also a DE integrator for the force models (and associated variational equations).

Happily, considerable research and development has gone into the field of DE integration. The DC practitioner can largely treat the various DE solvers as interchangeable subsystems that display different performance characteristics. It is not necessary to become expert in the fine details to get good results. The overview of numerical integration in Chapter 4 by Montenbruck and Gill (2000) is recommended. For a first exposure to numerical integration, many numerical analysis textbooks are available (e.g., Gerald and Wheatley, 1989).

Since no issues regarding the internal mechanics of DE solvers have been raised, the selection of a solver simplifies. A set of numerical tests on DC process will be conducted. We want the most accurate DE solver. Figures 4.6 and 4.9 in Montenbruck and Gill (2000) shows the "DE" code (Shampine and Gordon, 1975) provides 13 digits of relative accuracy (for eccentricity of 0.1).

The "DE" code is part of a solver package called DEPAC (Shampine and Watts, 1980). And one specific member of the package is called DDEABM. The leading "D" refers to double precision, and the trailing "ABM" refer to the Adams-Bashforth-Moulton predictor-corrector formulas of orders one through twelve. Fortran 77 source can be found through the Netlib repository at https://netlib.org, under the SLATEC library at https://netlib.org/slatec/src/. Searching that link for "ddeabm.f plus dependencies" will lead to https://netlib.org/cgi-bin/netlibfiles.pl?filename=/slatec/src/ddeabm.f (which offers download options). A C++ version of "DE" can be found on the CD with Montenbruck and Gill (2000).

The simplest, textbook DE solver, Runge-Kutta, typically uses a fixed order, a fixed step size, and is a single-step method. By contrast, DDEABM is a variable order, variable step, multistep predictor-corrector method. Further, considerable care taken in the package to automatically adjust order and step size. In fact, the DDEABM interface is presented as interval based, with no user specification of "step".

Instructions for invocation of DDEABM can be found in (Shampine and Watts, 1980) as well as in an extended set of comment records at the beginning of the subroutine. This study will not describe each subroutine parameter, but will provide some tips for usage.

The automation of DDEABM still needs some user guidance. This is provided by relative and absolute error tolerances, RTOL and ATOL, respectively. These can be placed in either scalar variables or 1-D arrays. Since modern Fortran compilers tend to check typing across subroutine interfaces, compiler warning can be suppressed by declaring RTOL and ATOL as 1-D arrays (also see INFO(2) in the documentation). This allows flexibility, allowing the tolerances to be specifically set for each individual DE. In addition, my experience found that only RTOL was needed for the state vector equations, but that both RTOL and ATOL were needed for the variational equations.

Other user guidance to DDEABM is provided through an INFO array. In particular, DDEABM needs to be instructed if it is the first call for each new problem (so that it may reset certain internal variables). This is signaled through the INFO(1) array element. It is set to 0 for the first invocation, and 1 for the remainder. This detail must be remembered, because the DE integration will be iterated as part of the DC least squares process. Thus, DDEABM will need to be signaled to reset when each new iteration occurs.

RPAR and IPAR are arrays that can be used to transfer numbers between main routines and a user-created subroutine that supplies derivatives to DDEABM. RPAR and IPAR don't have to be used, but they must be specified. In this study's scenarios, RPAR and IPAR were not used. Instead, certain force model parameters were transferred by a Fortran, named COMMON block. It should be noted that if one wishes to count the number of calls to the derivative routine, then an element of IPAR could be used to hold the count (where the derivative routine would increment the count in each invocation).

4. Variational Equations -- Motivation

When we consider the DC process, we must ask how one can solve for corrections to unknown parameters, $\mathbf{X}$, when there is no general orbital analytic formulation to compute those parameters. We saw that DE integration allows us to compute satellite position and velocity, $(\mathbf{r}, \dot{\mathbf{r}})$, at any desired time, t. However, the integration requires us to provide $(\mathbf{r}, \dot{\mathbf{r}})$ at some starting epoch, t_0. This is described as the initial value problem (IVP) of ordinary differential equations (Shampine and Gordon, 1975). In essence, the DEs alone provide a landscape of all possible solutions. And any individual initial value set will "select" a specific trajectory from that landscape.

Thus, the DE integration machinery, driven by user-defined, physical force models, substitutes for a general analytic formulation. And, we use the least squares unknown parameters, $\mathbf{X}_0$ (which become $\mathbf{X}_a$ at convergence) to establish the best initial values for DE integration. It must be emphasized that frequently there are other parameters in a least squares DC problem. But, without loss of generality, we currently focus in Sections 4 and 5 on the trajectory initial values (state vector initial values) as unknown parameters. This study will refer to these values as *state parameters*. This study defines state parameters as those unknown parameters that are initial values for the dynamical elements in the DE equations of motion. We handle the cases of additional parameter types in Section 6.

This study focuses on the Cowell formulation (3-D position and velocity) for the state vector and state parameters (Cappellari et al., 1976 and Long et al., 1989). This state vector of 6 elements entails integration of a system of 6 DEs to establish estimated $(\mathbf{r}, \dot{\mathbf{r}})$, at any point in time, t, by means of physical force models. This, in turn, suffices for an observation model $\mathbf{F}(t_t, t_r, \mathbf{r}, \dot{\mathbf{r}}, \mathbf{o}, \dot{\mathbf{o}})$, where $(\mathbf{o}, \dot{\mathbf{o}})$ represents observational site position and velocity, and (t_t, t_r) for transmit and receipt times. For simplicity of notation, we will now suppress the distinction between transmit and receipt times of an observation, and just use time, t.

In this setup we have satisfied the least squares requirement for an observation model vector, $\mathbf{F}(\mathbf{X}_a)$, seen in Section 2. We now address the design matrix (Jacobian), $\mathbf{A} = \partial\mathbf{F}/\partial\mathbf{X}$. In the general case, the denominator of the partials, $\partial\mathbf{X}$, are with respect to the current, iterated values of *all* the unknown parameters, $\mathbf{X}_0$. The denominator of the partials, $\partial\mathbf{X}$, does not refer to the unknown parameter correction column vector of Section 2.

For the remainder of Sections 4 and 5, we will only consider the case where all the unknown parameters are state parameters. Typical measurements, such as range, range rate, elevation, azimuth, and their rates, have straightforward representations in the $\mathbf{F}(t, \mathbf{r}, \dot{\mathbf{r}}, \mathbf{o}, \dot{\mathbf{o}})$ canonical form. However, partial derivatives of the canonical model only provide partials where the observations and unknowns are at the same epoch, t. That is, $\partial\mathbf{F}(t)/\partial\mathbf{X}(t)$. What is required for DC is *not* $\partial\mathbf{F}(t)/\partial\mathbf{X}(t)$, but $\mathbf{A}=\partial\mathbf{F}(t)/\partial\mathbf{X}(t_0)$. Recall, the unknown parameters are state parameters; and, by definition, are DE initial values referenced to time, t_0.

By the chain rule: $\partial\mathbf{F}(t)/\partial\mathbf{X}(t_0) = \partial\mathbf{F}(t)/\partial\mathbf{X}(t) \times \partial\mathbf{X}(t)/\partial\mathbf{X}(t_0)$. Both right hand terms are matrices. Define $\mathbf{H} = \partial\mathbf{F}(t)/\partial\mathbf{X}(t)$ and call it *incomplete partials*. It is an n x u matrix, it is a Jacobian, it is evaluated with $\mathbf{X}_0$ in iteration, but it is *not* the least squares design matrix, $\mathbf{A}$. The last term above we denote $\mathbf{\Phi} = \partial\mathbf{X}(t)/\partial\mathbf{X}(t_0)$, or sometimes, $\mathbf{\Phi}(t,t_0)$ (Montenbruck and Gill, 2000, Eq. 7.1). This is called the transition matrix or the state transition matrix (STM). For our 6 state parameters, the STM is 6x6, and it is not generally symmetric. Just to be explicit, the STM contents for the Cowell formulation are

$$\mathbf{\Phi}(t,t_0) = \begin{pmatrix} \frac{\partial X(t)}{\partial X(t_0)} & \frac{\partial X(t)}{\partial Y(t_0)} & \frac{\partial X(t)}{\partial Z(t_0)} & \frac{\partial X(t)}{\partial \dot{X}(t_0)} & \frac{\partial X(t)}{\partial \dot{Y}(t_0)} & \frac{\partial X(t)}{\partial \dot{Z}(t_0)} \\ \frac{\partial Y(t)}{\partial X(t_0)} & \frac{\partial Y(t)}{\partial Y(t_0)} & \frac{\partial Y(t)}{\partial Z(t_0)} & \frac{\partial Y(t)}{\partial \dot{X}(t_0)} & \frac{\partial Y(t)}{\partial \dot{Y}(t_0)} & \frac{\partial Y(t)}{\partial \dot{Z}(t_0)} \\ \frac{\partial Z(t)}{\partial X(t_0)} & \frac{\partial Z(t)}{\partial Y(t_0)} & \frac{\partial Z(t)}{\partial Z(t_0)} & \frac{\partial Z(t)}{\partial \dot{X}(t_0)} & \frac{\partial Z(t)}{\partial \dot{Y}(t_0)} & \frac{\partial Z(t)}{\partial \dot{Z}(t_0)} \\ \frac{\partial \dot{X}(t)}{\partial X(t_0)} & \frac{\partial \dot{X}(t)}{\partial Y(t_0)} & \frac{\partial \dot{X}(t)}{\partial Z(t_0)} & \frac{\partial \dot{X}(t)}{\partial \dot{X}(t_0)} & \frac{\partial \dot{X}(t)}{\partial \dot{Y}(t_0)} & \frac{\partial \dot{X}(t)}{\partial \dot{Z}(t_0)} \\ \frac{\partial \dot{Y}(t)}{\partial X(t_0)} & \frac{\partial \dot{Y}(t)}{\partial Y(t_0)} & \frac{\partial \dot{Y}(t)}{\partial Z(t_0)} & \frac{\partial \dot{Y}(t)}{\partial \dot{X}(t_0)} & \frac{\partial \dot{Y}(t)}{\partial \dot{Y}(t_0)} & \frac{\partial \dot{Y}(t)}{\partial \dot{Z}(t_0)} \\ \frac{\partial \dot{Z}(t)}{\partial X(t_0)} & \frac{\partial \dot{Z}(t)}{\partial Y(t_0)} & \frac{\partial \dot{Z}(t)}{\partial Z(t_0)} & \frac{\partial \dot{Z}(t)}{\partial \dot{X}(t_0)} & \frac{\partial \dot{Z}(t)}{\partial \dot{Y}(t_0)} & \frac{\partial \dot{Z}(t)}{\partial \dot{Z}(t_0)} \end{pmatrix}$$

Now our chain rule: $\partial\mathbf{F}(t)/\partial\mathbf{X}(t_0) = \partial\mathbf{F}(t)/\partial\mathbf{X}(t) \times \partial\mathbf{X}(t)/\partial\mathbf{X}(t_0)$, can be written in matrix form as $\mathbf{A} = \mathbf{H}\ \mathbf{\Phi}$, where the standard rules of matrix multiplication apply. See also Vallado (2013, Eq. 10-16). For n_s state parameters, the design matrix, **A**, is an n x n_s matrix; the incomplete partials, **H,** is an n x n_s matrix; and the STM, **Φ,** is an n_s x n_s matrix. Simply put, when we take partials of our observation model, **F**, (which refer to $(\mathbf{r}, \dot{\mathbf{r}})$ at time, t), they need to be transformed by an STM, **Φ**, into the design matrix partials, **A**, taken with respect to state parameters, $(\mathbf{r}_0, \dot{\mathbf{r}}_0)$. We shall see in the next section how to obtain $\mathbf{\Phi}(t,t_0)$.

As an implementation detail, the STM transformation could be written as:

$$\mathbf{A}_t = \mathbf{H}_t\ \mathbf{\Phi}(t,t_0)$$

where the matrix subscript, "t", refers to only those observations at time, t. The STM evolves through time, and is suitable to transform only those partials of observations at a contemporary epoch. One should *not* store the entire **H** matrix, and then transform using the final STM at the end of the observation series. Process the observations in temporal order, and transform the partials row by row using the current, correct version of the STM.

5. Variational Equations – The State Transition Matrix

It happens that the STM is the solution to a set of DEs known as variational equations. Therefore, we will need to integrate a DE for *each* element in $\mathbf{\Phi}(t,t_0)$. For a simple 6 state parameter DC we integrate 6 x 6 = 36 additional DEs simultaneously with the state vector DE integration of 6 DEs.

One definition of variational equations on the Internet is found at https://encyclopediaofmath.org/wiki/Variational_equations :

“The variational equation of order k is a linear differential (difference) equation whose solution is the k-th derivative with respect to a parameter of the solution of a differential (difference) equation.”

If we restrict the definition to a first order system and add some emphasis:

The variational equation is a linear DE whose solution is the first derivative with respect to a parameter of the solution of *a differential equation*.

A key takeaway of this definition is that there must be one or more underlying, fundamental DEs. And, the variational DEs are DEs of the solution to the fundamental DEs. This is emphasized in Beutler (2005, eq. 5.2) where he discusses both primary and variational equations. Variational equations are auxiliary to primary DEs.

The derivation of the STM solution in Section 7.2.1 of Montenbruck and Gill (2000, pg. 240) is so straightforward, this author must copy it. Notation is changed slightly and terminology is altered to provide reading continuity.

Our state vector of unknown state parameters is (ibid., 7.37)

$$\mathbf{X}(t) = \begin{pmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{pmatrix}$$

which obeys a set of primary differential equations (DE) (ibid., 7.38)

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{X}(t) = \mathbf{f}(t, \mathbf{X}) = \begin{pmatrix} \mathbf{v}(t) \\ \mathbf{a}(t, \mathbf{r}, \mathbf{v}) \end{pmatrix}$$

and, derivatives with respect to the parameters of the solution (to the primary DEs) are (ibid., 7.39)

$$\frac{\partial}{\partial \mathbf{X}(t_0)} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{X}(t) = \frac{\partial \mathbf{f}\big(t, \mathbf{X}(t)\big)}{\partial \mathbf{X}(t_0)} = \frac{\partial \mathbf{f}\big(t, \mathbf{X}(t)\big)}{\partial \mathbf{X}(t)} \quad \frac{\partial \mathbf{X}(t)}{\partial \mathbf{X}(t_0)}$$

The state transition matrix (STM) (ibid., 7.40)

$$\mathbf{\Phi}(t, t_0) = \frac{\partial \mathbf{X}(t)}{\partial \mathbf{X}(t_0)}$$

can be obtained from the DE (ibid., 7.41)

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{\Phi}(t, t_0) = \frac{\partial \mathbf{f}\big(t, \mathbf{X}(t)\big)}{\partial \mathbf{X}(t)} \ \mathbf{\Phi}(t, t_0)$$

or (ibid., 7.42)

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}(t,t_0) = \begin{pmatrix} \mathbf{0}_{3\text{x}3} & \mathbf{I}_{3\text{x}3} \\ \dfrac{\partial\mathbf{a}(t,\mathbf{r},\mathbf{v})}{\partial\mathbf{r}(t)} & \dfrac{\partial\mathbf{a}(t,\mathbf{r},\mathbf{v})}{\partial\mathbf{v}(t)} \end{pmatrix}_{6\text{x}6} \boldsymbol{\Phi}(t,t_0)$$

Now that we have defined the variational DE for a simple 6 state parameter problem, we need initial conditions to permit computation: $\boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}_{6\text{x}6}$ (ibid.). These initial conditions should come as no surprise. Our STM is defined as the Jacobian matrix $\partial\mathbf{X}(t)/\partial\mathbf{X}(t_0)$. So, if $t = t_0$, $\boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}$. Also, in general, $\boldsymbol{\Phi}(t, t) = \mathbf{I}$.

6. Variational Equations – The Sensitivity Matrix

State parameters were defined in Section 4 as those unknown parameters that are initial values for a DE set. However, there are two other disjoint sets of unknown parameters in a DC problem: *force model parameters* and *observation model parameters*. Observation model parameters are unknowns to be solved for that provide a better model of the observations. Examples of these would be offsets of an antenna to the center of mass of a satellite, or a tropospheric refraction term. Force model parameters are unknowns to be solved for that provide a better model of the forces involved. Examples of these would be solar radiation pressure, or one or more terms in the Earth's geopotential. Note that neither force model nor observation model parameters are described by a DE. To be sure, force model parameters will participate in the DE solution of state parameters and in the variational DEs above that solves the STM. But force model parameters are not dependent variables of a DE. They are a solution of the least squares DC problem. If a force model parameter is found to be described by a DE, then that parameter would be elevated to a state parameter, and would become part of the DE state vector. The force model would be altered to describe the DE governing the newly minted state parameter. And, that altered force model might require other new force model parameters to portray the new, larger, state vector.

This study will follow the mathematical setup of DC and will maintain a strict distinction between state parameters, force model parameters, and observation model parameters. This study will *not* use the term, dynamic parameter; which occasionally appears in OD literature. For a given DC problem, there are n_s state parameters, n_p force model parameters, and n_q observation model parameters. These distinctions are woven into the display of partial derivative categories found in Montenbruck and Gill (2000, pg. 233-234).

While the least squares adjustment unknown parameters can be in any order, it is useful to consider parameters clustered into disjoint sub-vectors:

- State Parameters, $\mathbf{X}_s$ — length, n_s
- Force Model Parameters, $\mathbf{X}_p$ — length, n_p
- Observation Model Parameters, $\mathbf{X}_q$ — length, n_q

The least squares parameter vectors $\mathbf{X}_0$, $\mathbf{X}_a$, and $\mathbf{X}$ can each be partitioned, $(\mathbf{X}_s \mid \mathbf{X}_p \mid \mathbf{X}_q)$. Note that the state parameters, $\mathbf{X}_s$, *always* refer to initial conditions of the DE state vector. The total number of unknowns, u, is found by $u = n_s + n_p + n_q$. A given DC problem may have n_p and/or n_q equal to 0. If there are no state parameters, there is no DE, no force model to inform the DE, and problem collapses to simple parametric least squares, $u = n_q$.

In DC with force model parameters, $n_p > 0$, we know that we will solve for corrections to our state and force parameters (and, if needed, n_q observation model parameters), all contained in an iteratively improved $\mathbf{X}_0$. The state vector, $\mathbf{X}(t)$, now also changes depending on the iterated estimate of the force parameters. We now need to compute partial derivatives of our observation model with respect to force parameters. We follow Montenbruck and Gill (2000, pg. 241), with modifications for continuity.

Denote the force model parameters, $\mathbf{p}$, as an n_p x 1 column vector. Recall the STM, where the state vector derivative was denoted as $\mathbf{f}(t, \mathbf{X})$. With the addition of force model parameters, our derivative is now $\mathbf{f}(t, \mathbf{X}, \mathbf{p})$. We now have (ibid., 7.43)

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathbf{X}(t)}{\partial \mathbf{p}} = \frac{\partial \mathbf{f}(t, \mathbf{X}(t), \mathbf{p})}{\partial \mathbf{X}(t)}\frac{\partial \mathbf{X}(t)}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}(t, \mathbf{X}(t), \mathbf{p})}{\partial \mathbf{p}}$$

Define the sensitivity matrix

$$\mathbf{S}(t)_{6\mathrm{x}n_p} = \frac{\partial \mathbf{X}(t)}{\partial \mathbf{p}}$$

and compute from the DE (ibid., 7.44)

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{S}(t) = \begin{pmatrix} \mathbf{0}_{3\mathrm{x}3} & \mathbf{I}_{3\mathrm{x}3} \\ \dfrac{\partial \mathbf{a}(t, \mathbf{r}, \mathbf{v}, \mathbf{p})}{\partial \mathbf{r}(t)} & \dfrac{\partial \mathbf{a}(t, \mathbf{r}, \mathbf{v}, \mathbf{p})}{\partial \mathbf{v}(t)} \end{pmatrix}_{6\mathrm{x}6} \mathbf{S}(t) + \begin{pmatrix} \mathbf{0}_{3\mathrm{x}n_p} \\ \dfrac{\partial \mathbf{a}(t, \mathbf{r}, \mathbf{v}, \mathbf{p})}{\partial \mathbf{p}} \end{pmatrix}_{6\mathrm{x}n_p}$$

To quote Montenbruck and Gill (2000, pg. 241),

"Since the state vector at t_0 does not depend on any force model parameter, the initial value of the sensitivity matrix is given by $\mathbf{S}(t_0) = \mathbf{0}$."

And, in the quote, "state vector at t_0", we call them *state parameters*; described earlier in this Section.

The initial value of the sensitivity matrix, $\mathbf{S}(t_0) = \mathbf{0}$, is not only provided in (ibid. pg. 241). It is found in (Beutler, 2005, pg. 179, eq. 5.10),

"… if parameter p is one of the dynamical parameters (see parameter definition (5.2))

$$\mathbf{Z}^{(i)}(t_0) = \mathbf{0}, \qquad i = 0, 1, \ldots, n - 1, \qquad p \in \{p_{nd+1}, p_{nd+2}, \ldots, p_{nd+m}\}$$ "

and in (Jäggi and Arnold 2017, pg. 55),

"For $P_i \in \{a, e, i, \Omega, \omega, u_0\}$ the variational equations (2.22) are a linear, homogeneous differential equation system of second order in time with initial values $\mathbf{z}_{P_i}(t_0) \neq \mathbf{0}$ and $\dot{\mathbf{z}}_{P_i}(t_0) \neq \mathbf{0}$. For $P_i \in \{Q_1, \ldots, Q_d\}$ (2.22) are inhomogeneous, but have zero initial values because the initial satellite state does not depend on the force model parameters."

In addition, the general mathematical derivation of, $\mathbf{S}(t_0) = \mathbf{0}$, is found in Coddington and Levinson (1955, Chapter 1, Section 7), and is detailed in Section 11 of this study.

It is seen the sensitivity matrix performs the same function as the STM. They both relate the evolving state vector, $\mathbf{X}(t)$, to unknown parameters (either state or force). It is natural to combine the STM and sensitivity matrix; and, thus, make a combined form of the variational equations (Montenbruck and Gill, 2000, Eq. 7.45)

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\Phi}|\mathbf{S}) = \begin{pmatrix} \mathbf{0}_{3\mathrm{x}3} & \mathbf{I}_{3\mathrm{x}3} \\ \dfrac{\partial \mathbf{a}}{\partial \mathbf{r}} & \dfrac{\partial \mathbf{a}}{\partial \mathbf{v}} \end{pmatrix}_{6\mathrm{x}6} (\boldsymbol{\Phi}|\mathbf{S}) + \begin{pmatrix} \mathbf{0}_{3\mathrm{x}6} & \mathbf{0}_{3\mathrm{x}n_p} \\ \mathbf{0}_{3\mathrm{x}6} & \dfrac{\partial \mathbf{a}}{\partial \mathbf{p}} \end{pmatrix}_{6\mathrm{x}(6+n_p)}$$

Just to be explicit, the combined matrix, $(\boldsymbol{\Phi}|\mathbf{S})$, is a 6 x $(6+n_p)$ matrix when n_s=6.

As with the STM, the chain rule is implemented as a matrix product. The incomplete partials, **H**, are still the same as before; an n x 6 matrix, where n (no subscript) is the number of observations. The new matrix product chain rule, $\mathbf{A} = \mathbf{H}\,(\boldsymbol{\Phi}|\mathbf{S})$, generates a design matrix, **A**; an n x $(6+n_p)$ matrix. This makes perfect sense, since our number of unknown parameters, u, were increased from 6 to $(6+n_p)$ when we added force parameters. As if by magic, this transformation of our incomplete partials, **H**, allows our measurements, through the design matrix, **A**, to solve for force parameters; even though force model parameters do not explicitly appear in the observation models, $\mathbf{F}(\mathbf{X}_a)$.

We now summarize some implementation details from Montenbruck and Gill (2000, pg. 242-243). They note that variational equations must be integrated simultaneously with the state vector. Thus, for $n_s = 6$, we integrate a total of $(n_s + (n_s \text{ x } (n_s + n_p)))$ first-order DEs. We quoted (ibid.) earlier that it is common to apply simplified force models for variational equation solution. They also relay warnings by several authors that one should use consistent models in the simultaneous integration of state vector and variational equations. This leaves the DC practitioner with two choices. One, integrate the variational equations with the same rigorous model that is required for the state equations. Or, do two integrations in parallel. One integration would be rigorous for the state equations. And the other integration would use a simplified force model for both the state and the variational equations simultaneously. The state vector results from the simplified force model would then be discarded (ibid.). This second approach can be faster. Recall there are only n_s state DEs, and n_s x $(n_s + n_p)$ variational DEs. However, for this study, since the best possible accuracy is desired, a suitably rigorous force model is used for both the state and variational equations.

In closing this section, we explicitly consider the case where we have state, force, and observation model parameters participating as unknown parameters. That is, n_s, n_p, and n_q are greater than 0. Now the unknown parameters are partitioned as ($\mathbf{X}_s$ | $\mathbf{X}_p$ | $\mathbf{X}_q$). Similarly, our design matrix partitions as $\mathbf{A}$ = ($\mathbf{A}_s$ | $\mathbf{A}_p$ | $\mathbf{A}_q$). The incomplete partials, **H**, is still an n x n_s matrix, as seen near the end of Section 4 and above in Section 6. The full expression for the design matrix, **A**, is now written as

$$\mathbf{A} = (\mathbf{A}_s \mid \mathbf{A}_p \mid \mathbf{A}_q) = (\mathbf{A}_{sp} \mid \mathbf{A}_q) = (\mathbf{H}\ (\mathbf{\Phi} | \mathbf{S}) \mid \mathbf{A}_q)$$

where $\mathbf{A}_{sp}$ represents the design matrix columns corresponding to the state and force parameters. The combined variational equations do not transform the incomplete partials for the observation model unknown parameters, $\mathbf{A}_q$. Those partial derivatives are merely appended to the transformed, design matrix partials for the state and force unknown parameters.

7. Uniform Gravity Model -- I

Recall that this study is performed to computationally test the existing theory of differential correction (DC), including the use of variational equations. Simple models that possess exact analytic solutions are implemented in the most rigorous fashion possible, to the limits of computational precision, in the DC setup. While more elaborate models could be used, exact analytic formulas allow creation of synthetic data which are perfectly in agreement with the observation models and the defining state parameters. Those perfect data then feed an iterated least squares DC solution (which includes DE integration) where the initial values of the unknown parameters are imperfectly known. Behavior is then analyzed.

This author is highly enthusiastic about Section 1.2 of Tapley, et.al (2004) which describes their uniform gravity field model (UGFM). It is simply a 2-D trajectory of an object in a uniform gravity field, with associated measurement types from a fixed tracking station. It is idealized ballistic motion. It is simple enough to provide an exact analytic solution, yet complex enough to perform OD by either DC or various Kalman filters. Tapley et al. (2004) also provide Exercise 1, pg. 16, that requires parameters to be recovered from range observations by means of Newton-Raphson iteration of the UGFM analytic formulas.

The UGFM equations of motion where gravity is constant are (ibid., 1.2.1)

$$\ddot{X}(t) = 0$$
$$\ddot{Y}(t) = -g$$

Integrate to obtain a system of first-order DEs where reference time $t_0 = 0$ (ibid., 1.2.2)

$$
\begin{aligned}
X(t) &= X_0 + \dot{X}_0 t \\
Y(t) &= Y_0 + \dot{Y}_0 t - ½ g t^2 \\
\dot{X}(t) &= \dot{X}_0 \\
\dot{Y}(t) &= \dot{Y}_0 - g t
\end{aligned}
$$

Note the UGFM is in an inertial frame of reference. There are no rotations or fictious forces. There is no atmosphere, so no velocity dependent forces. There is no Sun, no Moon, no direct or indirect tidal effects. The DC unknown parameters are $(X_0, Y_0, \dot{X}_0, \dot{Y}_0)^t$. Also, the DE state vector is $\mathbf{X}(t) = (X, Y, \dot{X}, \dot{Y})^t$.

There is a tracking station with known, constant coordinates (X_s, Y_s). Observations from station to object are range, ρ, which is a nonlinear function (ibid., 1.2.6)

$$\rho = \sqrt{(X - X_s)^2 + (Y - Y_s)^2}$$

And, our analytic expression for range at time, *t*, becomes (ibid., 1.2.7)

$$\rho_t = \sqrt{(X_0 + \dot{X}_0 t - X_s)^2 + (Y_0 + \dot{Y}_0 t - ½ g t^2 - Y_s)^2}$$

Note the UGFM speed of light is infinite. There is no distinction between signal transmit time and receipt time. There are no signal delays or refraction effects. There is no terrain or actual surface of an Earth, no signal blockage can occur.

Now, following Tapley, et.al (2004, pg. 16) we establish the Uniform Gravity Scenario:

$$
\begin{aligned}
X_s &= 1.0 \\
Y_s &= 1.0 \\
X_0 &= 1.0 \\
Y_0 &= 8.0 \\
\dot{X}_0 &= 2.0 \\
\dot{Y}_0 &= 1.0 \\
g &= 0.5
\end{aligned}
$$

Note that per Exercise 1, the quantities do not have specific units. However, X, Y, and ρ have length, $\dot{X}$ and $\dot{Y}$ have velocity, *g* has acceleration, and *t* has time. Lack of specific units will not inhibit the test in any way.

The exact analytic equation above and the defining scenario parameters are sufficient to generate ten epochs of data at times 0 through 9, inclusive. The times and ranges are shown in Table 7.1. Computations were performed in Fortran 77, in double precision (64-bit floating point), per the IEEE 754 standard. Since my CPU includes an integrated x87 capability, register arithmetic is performed to 80-bit floating point (double-extended) precision (18 decimal digits). However, variable storage is still 64-bit. Thus, the x87 capability provides guard digits for some of the arithmetic.

Table 7.1 – Perfect Range Data for Uniform Gravity Model Scenario

0	7.0000000000000000
1	8.0039052967910607
2	8.9442719099991592
3	9.8011478919563295
4	10.630145812734650
5	11.535271995059327
6	12.649110640673518
7	14.108951059522463
8	16.031219541881399
9	18.494931738181680

Note that these ranges carry more significant digits than the data displayed in Exercise 1 (ibid., pg. 16). Also, the t = 1 range corrects an error in Tapley, et.al (2004, pg. 16), which dropped the "2". And, our data are extended to 10 epochs, to support an overdetermined, least squares problem.

As described in our Section 2, least-squares requires observations, $\mathbf{L}_b$; an observation dispersion matrix, **D**; a weight matrix, $\mathbf{W} = \mathbf{D}^{-1}$; an observation model vector, $\mathbf{F}(\mathbf{X}_a)$; and, finally, a starting estimate of the unknown parameters, $\mathbf{X}_0$. We have $\mathbf{L}_b$ from Table 7.1., the number of observations, n = 10. We will treat our observations as uncorrelated. Since our data have no specific units, we arbitrarily choose σ = 0.000001. So, the observation dispersion matrix, **D**; is diagonal, where each diagonal element is the variance, σ^2. The weight matrix, $\mathbf{W} = \mathbf{D}^{-1}$, is trivially computed.

We adopt the perturbed initial conditions found in Exercise 1 (ibid.), as the starting estimate of the unknown parameters, $\mathbf{X}_0$,

$$X_0 = 1.5$$
$$Y_0 = 10.0$$
$$\dot{X}_0 = 2.2$$
$$\dot{Y}_0 = 0.5$$

We identify all of these unknown parameters as *state parameters*, as described earlier. Therefore, $u = n_s = 4$. Note, however, that the force model for X, $\ddot{X}(t) = 0$, is trivial. We could declare X_0 and $\dot{X}_0$ as *observation model parameters*, since they have immediate analytic forms. For the purposes of this test, we will treat them as state parameters.

For the observation model, $\mathbf{F}(\mathbf{X}_a)$ (or $\mathbf{F}(\mathbf{X}_0)$ if you like), one should use

$$\rho = \sqrt{(X - X_s)^2 + (Y - Y_s)^2}$$

where X and Y at time, t, are obtained by DE integration. But, as a trial, I chose the analytic form for the observation model to obtain computed observations, $\mathbf{L}_0 = \mathbf{F}(\mathbf{X}_0)$. The computed range is

$$\rho_t = \sqrt{(X_0 + \dot{X}_0 t - X_s)^2 + (Y_0 + \dot{Y}_0 t - ½gt^2 - Y_s)^2}$$

avoiding the state vector integration. (The variational equations were, of course, integrated by DDEABM.) Note that in the least squares iteration, this expression for range is not exact, since it uses the current iterates, $\mathbf{X}(t_0) = (X_0, Y_0, \dot{X}_0, \dot{Y}_0)^t$, and not the scenario defining parameters.

For the observation partial derivatives with respect to the unknown parameters (which are state parameters), consider a range at a single epoch, *i*, and compute the partials:

$$\mathbf{H}_\mathrm{i} = \left(\frac{(X_i - X_s)}{\sqrt{(X_i - X_s)^2 + (Y_i - Y_s)^2}} \quad \frac{(Y_i - Y_s)}{\sqrt{(X_i - X_s)^2 + (Y_i - Y_s)^2}} \quad 0 \quad 0 \right)$$

Note, the partials are derived from the earlier range model and not from the analytic form.

Since this scenario has no force model parameter, the variational DE consists solely of the state transition DE (ibid., Eq. 7.42). Since there is no atmospheric drag, there is no spatial dependence of **a** on **v**; the lower right quadrant is $\mathbf{0}_{2x2}$. Further, since gravity is uniform in this scenario, there is no spatial dependence of **a** on **r**; the lower left quadrant is $\mathbf{0}_{2x2}$. These are the set of 16 variational DEs integrated in time by DDEABM.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{\Phi}(t, t_0) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{\Phi}(t, t_0)$$

As per DC standard practice, the STM initial conditions are: $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{4x4}$.

Then the row of partials, $\mathbf{H}_\mathrm{i}$, which are partial derivatives with respect to the evolving state vector contents, must be converted to a row of partial derivatives with respect to the state parameters (which are established at t_0). This is standard matrix multiplication

$$\mathbf{A}_\mathrm{i} = \mathbf{H}_\mathrm{i}\, \mathbf{\Phi}(t, t_0)$$

Remember that this scenario treats the uniform gravity, *g*, as known. There are no force parameters, and no sensitivity matrix. This scenario is designated toyorb3, and was cycled for 5 loops, 4 iterations. The results are now displayed.

```
program toyorb3 -- 2022jul14
 l.s. solve toy orbit by differential correction
 ddeabm() solve state transition matrix
 xsta,ysta =    1.0000000000000000        1.0000000000000000
 grav=   0.50000000000000000
 xsg =    1.0000000000000000        8.0000000000000000        2.0000000000000000
1.0000000000000000
 xsi =    1.5000000000000000        10.000000000000000        2.2000000000000002
0.50000000000000000
 elb =   (ranges, unitless, perfect)
```

```
   7.0000000000000000
   8.0039052967910607
   8.9442719099991592
   9.8011478919563295
   10.630145812734650
   11.535271995059327
   12.649110640673518
   14.108951059522463
   16.031219541881399
   18.494931738181680
 range sigmas =    9.9999999999999995E-007

 r ave/std/rms ->   1.6530917671396215       0.61241145335437730
1.7522144275072200
 r max/iep     ->   2.8978824759417172                9
 r min/iep     ->   1.0098713693832533                4
 x()=  -0.70065173134553760       -1.9751114500013178      -0.16047969703490095
0.52188400492563991
 finished iteration:            0
 r ave/std/rms ->   5.6115577639856438E-002   3.9370209349390797E-002
6.7408976402613688E-002
 r max/iep     ->  0.10413753301993012                8
 r min/iep     ->   6.7813090443795687E-003           1
 x()=   0.19659179133315585       -2.2165197214466095E-002  -3.8834659708808283E-002  -
2.1985584441933104E-002
 finished iteration:            1
 r ave/std/rms ->   1.1897182112654114E-003   6.8244927769086294E-004
1.3544713866210953E-003
 r max/iep     ->   2.7245296914477635E-003           0
 r min/iep     ->   5.3467904840154290E-004           4
 x()=    4.0592446112681160E-003  -2.7221576847883304E-003  -6.8553827157700250E-004
1.0148933535111728E-004
 finished iteration:            2
 r ave/std/rms ->   4.3945505536768790E-007   3.8144661589409232E-007
5.6927332133832454E-007
 r max/iep     ->   1.1950994620590905E-006           0
 r min/iep     ->   1.0912178893818236E-007           7
 x()=    6.9540120508809584E-007  -1.1950993927309184E-006  -1.0498472470905077E-007
9.0180931737262950E-008
 finished iteration:            3
 r ave/std/rms ->   2.5046631435543531E-014   1.5653680953819926E-014
2.9118167884184328E-014
 r max/iep     ->   4.4408920985006262E-014           1
 r min/iep     ->   3.5527136788005009E-015           9
 x()=   -8.4825778391485483E-014  -3.5633663706037350E-014   1.0243623586190599E-014
9.5437657783815004E-015
 finished iteration:            4
 xa()=    1.0000000000000067        8.0000000000000000        1.9999999999999993
0.99999999999999922
 err.=    6.6613381477509392E-015   0.0000000000000000       -6.6613381477509392E-016  -
7.7715611723760958E-016

 end of processing
```

We test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters of our scenario. This comparison is provided in Table 7.2.

Table 7.2 – Adjusted Unknowns, Scenario Parameters, and Error (`toyorb3`)

X_0	1.0000000000000067	1.0000000000000000	6.6613381477509392E-015
Y_0	8.0000000000000000	8.0000000000000000	0.0000000000000000
$\dot{X}_0$	1.9999999999999993	2.0000000000000000	-6.6613381477509392E-016
$\dot{Y}_0$	0.99999999999999922	1.0000000000000000	-7.7715611723760958E-016

The first scenario test shows the DC process does indeed recover the unknown scenario parameters to the limits of 64-bit machine precision. This validates the use of least squares to solve an overdetermined problem, the DC setup as a special subproblem of least squares, and the theory and implementation of the state transition matrix as a solution of the variational equations of the underlying differential equations of motion.

It is instructive to track the progress of the iteration. The iterated corrections to the unknown parameters, **X** (denoted in the computer output as "`x()`"), track the differences between successive values of the iterations of $\mathbf{X}_0$. It is seen that not only do the corrections become successively smaller, but that **X** magnitudes seem to be the square of the prior magnitudes. This is quadratic convergence. And, this is the fastest convergence to be expected from the standard iterated least squares method.

The output results of "`r`" are some diagnostic reports on the statistics of the range misclosures, **L**, for each iteration. They are seen to decrease. And as $\mathbf{X}_0$ iterates to $\mathbf{X}_a$, **L** will iterate to the range residuals, **V**. This is expected. But, unlike a real-world problem (which has noisy data), the perfect scenario data should ideally iterate to misclosures of zero magnitude. This is seen to occur within the limits of machine precision.

8. Uniform Gravity Model - - II

In this section, we explore a variant of the scenario of the prior section. The magnitude of the uniform gravity, g, is still considered constant. But now g is treated as an unknown parameter to be solved. The number of unknowns become $u = 5$. The DC unknown *state parameters* are still $(X_0, Y_0, \dot{X}_0, \dot{Y}_0)^t$. The number of state parameters remains $n_s = 4$. However, uniform gravity, g, is not described by a DE. Certainly, it participates in the force model and the associated DEs. But, uniform gravity, g, has no kinematics, no trajectory; it is a *force parameter*. The number of force parameters is $n_p = 1$.

We retain the UFGM equations of motion, and the first-order DE system of the prior section. We retain the equations defining range observations, the defining quantities of the Uniform Gravity Scenario, and the perfect range data of Table 7.1. We keep the range sigma, dispersion, and weight matrices.

We now completely adopt the perturbed quantities found in Exercise 1 (Tapley, et.al, 2004, pg. 16), as the starting estimate of the unknown parameters, $\mathbf{X}_0$,

$$X_0 = 1.5$$
$$Y_0 = 10.0$$
$$\dot{X}_0 = 2.2$$

$\dot{Y}_0 = 0.5$
$g = 0.3$

Note that *g* is a force parameter, and all force model and state parameters are defined as unknown parameters in the least squares problem.

Since this variant has a force model parameter, the variational DE is now the combined form holding both the state transition matrix and the sensitivity matrix (Montenbruck and Gill, 2000, Eq. 7.45) reproduced above. We integrate both matrices as the combined matrix, $(\mathbf{\Phi}|\mathbf{S})$; which is a n_s x $(n_s + n_p)$ matrix where $n_s = 4$ where $n_p = 1$. Note that the 4x4 matrix in the homogenous part is unchanged from the variational DE in the prior scenario. The fact that gravity is now an unknown has no bearing on the partials $\partial\mathbf{a}/\partial\mathbf{r}$ and $\partial\mathbf{a}/\partial\mathbf{v}$. Note the 2x1 submatrix in the lower right-hand corner of the inhomogenous term becomes

$$\frac{\partial\mathbf{a}}{\partial\mathbf{p}} = \begin{pmatrix} \frac{\partial\ddot{X}}{\partial g} \\ \frac{\partial\ddot{Y}}{\partial g} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

since g is now a force parameter. The full DE system is now written as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{\Phi}|\mathbf{S}) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}(\mathbf{\Phi}|\mathbf{S}) + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

These are the 20 variational DEs integrated in time by DDEABM. As per DC standard practice, the STM initial conditions are: $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{4x4}$, and the 4x1 sensitivity matrix initial conditions are $\mathbf{S}(t_0) = \mathbf{0}$.

The observation partial derivatives are unchanged from the prior section. Note that any given row of $\mathbf{H}_i$ is a 1x4 row. Those partials are converted to a row of partial derivatives with respect to the full set of unknowns, both state parameters and force model parameters, by matrix multiplication with the combined matrix:

$$\mathbf{A}_i = \mathbf{H}_i\,(\mathbf{\Phi}|\mathbf{S})$$

Note that where $\mathbf{H}_i$ is a 1x4 row, $\mathbf{A}_i$ is now a 1x5 row. We have gained the needed partial derivatives with respect to our force model parameters by the multiplication.

This variant scenario is designated toyorb4, and was cycled for 5 loops, 4 iterations. The results are now displayed.

```
program toyorb4 -- 2022jul17
 l.s. solve toy orbit+g by differential correction
 unknown g, ddeabm() variational equations
```

```
   perfect g=   0.50000000000000000
 imperfect g=   0.29999999999999999

 xsta,ysta=    1.0000000000000000        1.0000000000000000
   perfect xsg=    1.0000000000000000        8.0000000000000000        2.0000000000000000
1.0000000000000000
 imperfect xsi=    1.5000000000000000        10.000000000000000        2.2000000000000002
0.50000000000000000
 elb=   (ranges, unitless, perfect)
   7.0000000000000000
   8.0039052967910607
   8.9442719099991592
   9.8011478919563295
   10.630145812734650
   11.535271995059327
   12.649110640673518
   14.108951059522463
   16.031219541881399
   18.494931738181680
 range sigmas =    9.9999999999999995E-007

 r ave/std/rms ->   2.0945109394094805       0.35124721684114563
2.1208520037736651
 r max/iep     ->   2.6042921924618749                7
 r min/iep     ->   1.6561997776192747                2
 x()=   -1.2681644065778528       -1.9473834175392994       -8.4221430338402570E-003
0.66284223246339025       0.28751686699550305
 finished iteration:           0
 r ave/std/rms ->  0.32908934588554234       0.49538044995383879
0.57372590036723559
 r max/iep     ->   1.4731041259975086                9
 r min/iep     ->   1.0899767531142857E-003           3
 x()=   0.77112820968162055       -1.1409953827655528E-002 -0.18667541604578020      -
0.18174614096616182       -8.8534223321750538E-002
 finished iteration:           1
 r ave/std/rms ->   2.3748552944493361E-002   2.2083941383146576E-002
3.1668978311120692E-002
 r max/iep     ->   6.6134189328998616E-002           9
 r min/iep     ->  -2.5425219835817359E-004           4
 x()=   -2.6590379423758748E-003  -4.1205612638026121E-002  -4.9188535253674726E-003
1.8817629448968587E-002   9.9763412143549779E-004
 finished iteration:           2
 r ave/std/rms ->   8.2736522622717240E-005   7.4288585146776035E-005
1.0868416016374334E-004
 r max/iep     ->   1.7675112804127480E-004           8
 r min/iep     ->  -9.5148967460545464E-007           1
 x()=   -3.0478626098633457E-004  -1.0093474615796127E-006   1.6415389564219084E-005
8.6283156534648024E-005   1.9722876376153842E-005
 finished iteration:           3
 r ave/std/rms ->   3.2166761698704248E-009   2.3960968331073695E-009
3.9388015451500872E-009
 r max/iep     ->   6.6475589477477115E-009           0
 r min/iep     ->   5.3201887340037501E-011           8
 x()=    2.1099600711723380E-008  -6.6475588681354719E-009  -2.7845772098874676E-009  -
4.1027335159886004E-009  -6.7156462810898473E-010
 finished iteration:           4
 xa()=    1.0000000000000062        8.0000000000000000        1.9999999999999993
0.99999999999999822       0.49999999999999961
 err.=    6.2172489379008766E-015   0.0000000000000000       -6.6613381477509392E-016  -
1.7763568394002505E-015
 grav=   0.49999999999999961
 err.=   -3.8857805861880479E-016

 end of processing
```

As before, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters of this scenario variant that includes a force parameter and a sensitivity matrix. This comparison is provided in Table 8.1.

Table 8.1 – Adjusted Unknowns, Scenario Parameters, and Error (`toyorb4`)

X_0	1.0000000000000062	1.0000000000000000	6.2172489379008766E-015
Y_0	8.0000000000000000	8.0000000000000000	0.0000000000000000
$\dot{X}_0$	1.9999999999999993	2.0000000000000000	-6.6613381477509392E-016
$\dot{Y}_0$	0. 99999999999999822	1.0000000000000000	-1.7763568394002505E-016
g	0. 49999999999999961	0. 50000000000000000	-3.8857805861880479E-016

This second scenario test again shows the DC process does recover the unknown scenario parameters, which now include a force parameter, *g*, to the limits of 64-bit machine precision. This further validates the DC setup extended by force parameters, and the theory and implementation of the state transition matrix combined with a sensitivity matrix as a solution of the full variational equations (Montenbruck and Gill, 2000, Eq. 7.45) of the underlying differential equations of motion.

Convergence progress of both the parameters and the misclosures show rapid progress to machine precision limits.

9. Kepler Model - - I

Given the results of the Uniform Gravity Scenario of Sections 7 and 8, a desire to test in a more common situation, with Newtonian gravitation and realistic quantities, is irresistible. Let us construct a Kepler Scenario, which addresses the classic central force problem (Montenbruck and Gill, 2000, Ch. 2). This is an Earth, modeled as a point mass, and a satellite whose mass is negligible when compared to the Earth. Newton solved Kepler's problem, and provided an exact analytic solution for the satellite orbit under a central force. Again, we will create synthetic data which are perfectly in agreement with the observation models and the defining state parameters. Those data will be processed in an iterated least squares DC solution where the initial values of the unknown parameters are imperfectly known.

The equations of motion have no perturbing accelerations

$$\ddot{X} + GM\frac{X}{r^3} = 0$$
$$\ddot{Y} + GM\frac{Y}{r^3} = 0$$
$$\ddot{Z} + GM\frac{Z}{r^3} = 0$$

where *r* is the geometric distance between the geocenter and the object, and where GM is the standard gravitational parameter of the Earth. Note this is an inertial frame; no coordinate system rotations, no accelerations of the origin.

Integrate to obtain a system of six first-order DEs

$$\dot{X} - V_X = 0$$
$$\dot{Y} - V_Y = 0$$
$$\dot{Z} - V_Z = 0$$
$$\dot{V}_X + GM\frac{X}{r^3} = 0$$
$$\dot{V}_Y + GM\frac{Y}{r^3} = 0$$
$$\dot{V}_Z + GM\frac{Z}{r^3} = 0$$

where the DC unknown state parameters are $(X_0, Y_0, Z_0, \dot{X}_0, \dot{Y}_0, \dot{Z}_0)^t$.

There are four tracking stations with known, constant coordinates ($\mathbf{X}_s$, $\mathbf{Y}_s$, $\mathbf{Z}_s$). Their mutual positions form a regular tetrahedron. The idealized Earth is a circumsphere, of radius R=6371 km, where the Earth surface encloses the tetrahedron and touches each vertex. The spherical coordinates (latitude, longitude) of the stations are chosen as

$$(b_s, \ell_s)_1 = (0°, \Theta)$$
$$(b_s, \ell_s)_2 = (0°, -\Theta)$$
$$(b_s, \ell_s)_3 = (\Theta, 180°)$$
$$(b_s, \ell_s)_4 = (-\Theta, 180°)$$

Where Θ is the equatorial angle between a station and the prime meridian. By means of the expression for tetrahedron edge length, *a*, the circumsphere radius is R = (¼√6) *a*, (https://en.wikipedia.org/wiki/Tetrahedron). By plane trigonometry sin Θ = 2/√6, and Θ ≈ 54.735610317245360 degrees. The tracking station Cartesian coordinates in meters become

$(X_s, Y_s, Z_s)_1$ = (3678298.5650071050 5201899.7170905434 0.0000000000000000)
$(X_s, Y_s, Z_s)_2$ = (3678298.5650071050 -5201899.7170905434 0.0000000000000000)
$(X_s, Y_s, Z_s)_3$ = (-3678298.5650071050 0.0000000000000000 5201899.7170905434)
$(X_s, Y_s, Z_s)_4$ = (-3678298.5650071050 0.0000000000000000 -5201899.7170905434)

The interstation chord distance is 10403799.434181085 meters. The tracking stations and the idealized Earth surface are considered stationary in the inertial frame. There are no Earth rotations, no precession, no nutation, no polar motion, no plate tectonics, no subsidence.

There are two types of observations in this scenario. The first type is range, ρ, from station to satellite

$$\rho = \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}$$

The second type is range rate, ρ, (following Tapley et al., 2004, Eq. 1.2.6)

$$\dot{\rho} = \frac{1}{\rho}\left[(X - X_s)(\dot{X} - \dot{X}_s) + (Y - Y_s)(\dot{Y} - \dot{Y}_s) + (Z - Z_s)(\dot{Z} - \dot{Z}_s)\right]$$

Since all the tracking station velocities are zero, the scenario range rate simplifies to

$$\dot{\rho} = \frac{1}{\rho}\left[\dot{X}\,(X - X_s) + \dot{Y}\,(Y - Y_s) + \dot{Z}\,(Z - Z_s)\right]$$

Note the scenario speed of light is infinite. There is no distinction between signal transmit time and receipt time. There are no signal delays or refraction effects. There is no terrain or actual surface of an Earth, no signal blockage can occur. Yes, this does mean that a given range may pass through the idealized Earth to a satellite on the other side.

This author selects the scenario defining parameters (Kepler elements) from a set of linked problem sets in Satellite Geodesy (Course 777) and Advanced Satellite Geodesy (Course 873), given at the Department of Geodetic Science and Surveying at The Ohio State University, through the years of 1978-1986, and taught by Prof. Ivan Mueller. This experience included writing validated software that would compute a state vector for a Kepler orbit (among many other exercises).

Following the problem sets, define the Kepler Scenario as

semi-major axis	12267692.6 m.
eccentricity	0.003845 deg.
inclination	109.85396 deg.
ascending node	43.95923 deg.
argument of perigee	245.07169 deg.
mean anomaly	55.20345 deg.
GM	3.98603E14 m^3/s^{-2}

Note that newer values of GM are known, but the above replicates an old assignment. The mean anomaly is referred to 18 August 1976, 0^hUT. The start time, t_0, is 17 August, 1976, 12^hUT. The scenario t_0 inertial state vector (Cartesian) in meters and meters/second:

X_0 = -7856436.2041079123
Y_0 = -3154149.8830321119
Z_0 = -8815210.5483046416
$\dot{X}_0$ = 2296.0662918987064
$\dot{Y}_0$ = 3944.6910449260236
$\dot{Z}_0$ = -3449.9126329548199

The exact analytic orbit solution (which includes an iterative solution to Kepler's equation) allows computation of Cartesian elements at any point in time. This, in turn, allows generation of perfect synthetic data at any point in time

The scenario observation data set spans 24 hours, at 120 second intervals, for a total of 721 epochs. The data at each epoch are 4 ranges, and 4 range rates; one from each of the 4 tracking stations to the satellite. This gives a total of n=5768 perfect synthetic observations. The satellite has a period of just over 225 minutes. The satellite will perform 3¾ orbital revolutions in 24 hours.

Least-squares requires, $\mathbf{L}_b$, $\mathbf{D}$, $\mathbf{W} = \mathbf{D}^{-1}$, $\mathbf{F}(\mathbf{X}_a)$, and starting estimate, $\mathbf{X}_0$. We arbitrarily choose our range and range rate $\sigma = 0.001$ (m and m/s) to compute **D** and **W**.

We generate perturbed initial conditions by multiplying the defining Keplerian elements by the factor 1.0000001. The perturbed Kepler elements become

semi-major axis	12267693.8 m.
eccentricity	0.003845 deg.
inclination	109.85397 deg.
ascending node	43.95923 deg.
argument of perigee	245.07171 deg.
mean anomaly	55.20346 deg.

(Note that the computer output above is not displaying the full precision of the perturbed elements.)

Converting the perturbed Kepler elements into a Cartesian inertial frame by exact analytic formulas generates the perturbed initial values displayed in Table 9.1. These are taken as the starting estimate of the unknown parameters, $\mathbf{X}_0$, for this scenario.

Table 9.1 – Perturbed Initial Values and Perturbation (m and m/s)

X_0	-7856420.4697193988	15.734388513490558
Y_0	-3154119.6024935995	30.280538512393832
Z_0	-8815237.0415215995	-26.493216957896948
$\dot{X}_0$	2296.0784583470122	1.2166448305833910E-002
$\dot{Y}_0$	3944.6967362488972	5.6913228736448218E-003
$\dot{Z}_0$	-3449.8975864829749	1.5046471844925691E-002

Note that it is not desirable to generate perturbed initial conditions by multiplication of the defined Cartesian state vector by a perturbing factor. This would introduce a scale perturbation, to be sure. But it would not perturb other aspects of the orbit, such as inclination or argument of perigee. So, instead, the defining Kepler elements are perturbed, converted to Cartesian, and then adopted as our perturbed, $\mathbf{X}_0$.

We identify these unknown parameters as *state parameters*. Therefore, $u = n_s = 6$.

In contrast to the Uniform Gravity Scenario, the equations of motion as well as the variational equations will be integrated by DDEABM in this scenario. The six first-order DEs were displayed earlier in this section.

Since this scenario has no force model parameter, the variational DE consists solely of the state transition DE (Montenbruck and Gill, 2000, Eq. 7.42). Since there is no atmospheric drag, there is no spatial dependence of **a** on **v**; the lower right quadrant is $\mathbf{0}_{3x3}$. Since this scenario has Newtonian point mass gravitation, there now *is* spatial dependence of **a** on **r**. The lower left 3x3 quadrant is found in (ibid., Eq. 7.57). These are the set of 36 variational DEs integrated in time along with the 6 motion DEs.

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}(t,t_0) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{\mathrm{GM}}{r^5}(3\mathrm{X}^2 - r^2) & \frac{\mathrm{GM}}{r^5}3\mathrm{XY} & \frac{\mathrm{GM}}{r^5}3\mathrm{XZ} & 0 & 0 & 0 \\ \frac{\mathrm{GM}}{r^5}3\mathrm{YX} & \frac{\mathrm{GM}}{r^5}(3\mathrm{Y}^2 - r^2) & \frac{\mathrm{GM}}{r^5}3\mathrm{YZ} & 0 & 0 & 0 \\ \frac{\mathrm{GM}}{r^5}3\mathrm{ZX} & \frac{\mathrm{GM}}{r^5}3\mathrm{ZY} & \frac{\mathrm{GM}}{r^5}(3\mathrm{Z}^2 - r^2) & 0 & 0 & 0 \end{pmatrix} \boldsymbol{\Phi}(t,t_0)$$

As per DC standard practice, the STM initial conditions are: $\boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}_{6x6}$.

Turn now to the observation partial derivatives with respect to the unknown parameters (which are state parameters). The range at a single epoch, *i*, from a given tracking station to the satellite generates a 1x6 row of partial derivatives with respect to the evolving state vector contents

$$\mathbf{H}_\mathrm{i} = \begin{pmatrix} \frac{(\mathrm{X}_i - \mathrm{X}_s)}{\rho} & \frac{(\mathrm{Y}_i - \mathrm{Y}_s)}{\rho} & \frac{(\mathrm{Z}_i - \mathrm{Z}_s)}{\rho} & 0 & 0 & 0 \end{pmatrix}$$

The range rate at a single epoch, *i*, has a 1x6 row of partial derivatives (oriented for page display)

$$\mathbf{H}_i^t = \begin{pmatrix} \dfrac{\left(\dot{X}_i((Y_i - Y_s)^2 + (Z_i - Z_s)^2) - (X_i - X_s)\left((Y_i - Y_s)\dot{Y}_i + (Z_i - Z_s)\dot{Z}_i\right)\right)}{\rho^3} \\ \dfrac{\left(\dot{Y}_i((X_i - X_s)^2 + (Z_i - Z_s)^2) - (Y_i - Y_s)\left((X_i - X_s)\dot{X}_i + (Z_i - Z_s)\dot{Z}_i\right)\right)}{\rho^3} \\ \dfrac{\left(\dot{Z}_i((X_i - X_s)^2 + (Y_i - Y_s)^2) - (Z_i - Z_s)\left((X_i - X_s)\dot{X}_i + (Y_i - Y_s)\dot{Y}_i\right)\right)}{\rho^3} \\ \dfrac{(X_i - X_s)}{\rho} \\ \dfrac{(Y_i - Y_s)}{\rho} \\ \dfrac{(Z_i - Z_s)}{\rho} \end{pmatrix}$$

As before, the computed range, not the observed range is used in evaluating the partials. Also, as before, the row of partials, $\mathbf{H}_i$, must be converted to a row of partial derivatives with respect to the state parameters (which are established at t_0).

$$\mathbf{A}_i = \mathbf{H}_i\, \mathbf{\Phi}(t, t_0)$$

This scenario is designated kepdc, and was cycled for 5 loops, 4 iterations. The results are now displayed.

```
program kepdc -- 2022jul04
 l.s. solve kepler orbit by differential correction

  imperfect initial statevector
  rtol (all) =    9.9999999999999998E-013
  nep,nobs =          721        5768
 tracking stations --
           1   3678298.5650071050        5201899.7170905434        0.0000000000000000
           2   3678298.5650071050       -5201899.7170905434        0.0000000000000000
           3  -3678298.5650071050        0.0000000000000000        5201899.7170905434
           4  -3678298.5650071050        0.0000000000000000       -5201899.7170905434

 perfect kepler elements --
 semi-major axis     =  12267692.6 m.
 eccentricity        =    0.003845 deg.
 inclination         =  109.85396 deg.
 ascending node      =   43.95923 deg.
 argument of perigee =  245.07169 deg.
 mean anomaly        =   55.20345 deg.
 kep. period s/m/h=    13522.404232887979        225.37340388146632
3.7562233980244386

 non-perfect kepler element factor=   1.0000001000000001
 non-perfect kepler elements --
 semi-major axis     =  12267693.8 m.
 eccentricity        =    0.003845 deg.
 inclination         =  109.85397 deg.
 ascending node      =   43.95923 deg.
 argument of perigee =  245.07171 deg.
 mean anomaly        =   55.20346 deg.
```

```
 perfect statevector --
           1  -7856436.2041079123
           2  -3154149.8830321119
           3  -8815210.5483046416
           4   2296.0662918987064
           5   3944.6910449260236
           6  -3449.9126329548199

 non-perfect statevector and diff. with perfect --
           1  -7856420.4697193988        15.734388513490558
           2  -3154119.6024935995        30.280538512393832
           3  -8815237.0415215995       -26.493216957896948
           4   2296.0784583470122        1.2166448305833910E-002
           5   3944.6967362488972        5.6913228736448218E-003
           6  -3449.8975864829749        1.5046471844925691E-002

 top of perfect sat pos. --
           0  -7856436.2041079123       -3154149.8830321119       -8815210.5483046416
           1  -7568705.4140116964       -2676077.8998624440       -9215129.1542818397
           2  -7257182.4227614542       -2189593.7004120979       -9586079.9795341603
           3  -6922843.8984359335       -1696225.7544806539       -9926893.5148273204
           4  -6566739.2481008712       -1197524.7150596599       -10236495.860428371
           5  -6189987.2274864251       -695058.38216115907       -10513912.248329705
           6  -5793772.3135582767       -190406.60605464038       -10758270.229454534
           7  -5379340.8531783465        314843.85137669696       -10968802.513910934

 top of perfect data --
   0.0000000000000000        16750570.382064076
   0.0000000000000000        14661218.318668425
   0.0000000000000000        14962783.023493638
   0.0000000000000000        6360936.3130651973
   0.0000000000000000       -1733.3644659799313
   0.0000000000000000        818.82219574949602
   0.0000000000000000        1759.1899813539981
   0.0000000000000000       -1504.4673676341772
 range and range rate sigmas =    1.0000000000000000E-003

 r ave/std/rms  ->   1.1378606047417359        6.5690501024098751
6.6657470178827678
 r max/iep/ista ->   22.071017637848854               22           4
 r min/iep/ista ->  -20.044524449855089               20           1
 v ave/std/rms  ->   4.8443095562271953E-006   3.5201484780293425E-003
3.5195414697167184E-003
 v max/iep/ista ->   2.1348927599888157E-002           4           4
 v min/iep/ista ->  -1.4177053372861792E-002         716           1
 x()=   -15.734433993023401       -30.280563480175601        26.493160650825985       -
1.2166433821766409E-002  -5.6912969333682284E-003  -1.5046491384449134E-002
 finished iteration:            0
 r ave/std/rms  ->   2.8073553825197701E-004   3.1329261288325005E-003
3.1449380333146487E-003
 r max/iep/ista ->   9.1433562338352203E-003         662           4
 r min/iep/ista ->  -8.9136958122253418E-003         699           4
 v ave/std/rms  ->  -1.8539267225337745E-009   1.6375078350257764E-006
1.6372249648120881E-006
 v max/iep/ista ->   2.6390590619485010E-006         662           1
 v min/iep/ista ->  -8.7512969813019481E-006         681           4
 x()=    4.5421285137292600E-005   2.4731565036240077E-005   5.6397807789765331E-005  -
1.4594736206087564E-008  -2.6005482394895682E-008   1.9438502163048217E-008
 finished iteration:            1
 r ave/std/rms  ->  -2.6644092575025624E-008   8.7921884715936952E-007
8.7947009580589568E-007
 r max/iep/ista ->   4.5960769057273865E-006         693           4
 r min/iep/ista ->  -4.8484653234481812E-006         399           1
 v ave/std/rms  ->  -6.3562520172513534E-012   4.5928186772611337E-010
4.5924622431661313E-010
 v max/iep/ista ->   4.0487861951987725E-009         717           1
 v min/iep/ista ->  -1.3609451343654655E-009         693           3
 x()=    6.5696778682710613E-007   1.1025025362950520E-006  -9.1893720588178329E-007
4.7564992986365169E-010   2.0104234258224037E-010   4.7791834623287621E-010
 finished iteration:            2
```

```
 r ave/std/rms  ->   3.1389736527791468E-007   3.1621067269156416E-006
3.1771030006590398E-006
 r max/iep/ista ->   1.1287629604339600E-005         700           1
 r min/iep/ista ->  -1.0674819350242615E-005         703           4
 v ave/std/rms  ->  -1.2514606857355160E-011   1.7006857608651167E-009
1.7004369390045967E-009
 v max/iep/ista ->   3.3797959986259229E-009         668           1
 v min/iep/ista ->  -1.1144948075525463E-008         715           1
 x()=   -7.0163403881567769E-007  -1.6870383828025779E-006   2.0921600959568701E-006  -
6.5109245811717384E-010  -1.3940865105861005E-010  -9.3852270581333044E-010
 finished iteration:           3
 r ave/std/rms  ->  -2.1744186211062205E-007   1.8934797298250685E-006
1.9055978834513035E-006
 r max/iep/ista ->   7.3760747909545898E-006         696           4
 r min/iep/ista ->  -8.3427876234054565E-006         693           1
 v ave/std/rms  ->   3.5631208262820980E-012   9.9043952755028737E-010
9.9027421012650949E-010
 v max/iep/ista ->   7.4449246767471777E-009         716           1
 v min/iep/ista ->  -2.0597781258402392E-009         720           4
 x()=    1.2115031596703811E-007   6.5914783072148174E-007  -1.1656453645907811E-006
2.0940997411814440E-010  -3.7001265869844031E-011   4.8211897819411499E-010
 finished iteration:           4
 xa()=   -7856436.2041078936       -3154149.8830322730       -8815210.5483045448
2296.0662918986295        3944.6910449259826       -3449.9126329548994
 err.   =    1.8626451492309570E-008  -1.6111880540847778E-007   9.6857547760009766E-008
-7.6852302299812436E-011  -4.0927261579781771E-011  -7.9580786405131221E-011
 relerr.=   -2.3708525097639458E-015   5.1081531120389518E-014  -1.0987547855976917E-014
-3.3471290690069875E-014  -1.0375276824892454E-014   2.3067478765968335E-014

 end of processing
```

As before, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters of this scenario. Since our unknown parameters have realistic values (m and m/s), it is useful to compute the relative error to see how many significant digits have been recovered. This comparison is provided in Table 9.2.

Table 9.2 – Adjusted Unknowns, Error, and Relative Error (`kepdc`)

X_0	-7856436.2041078936	1.8626451492309570E-008	-2.3708525097639458E-015
Y_0	-3154149.8830322730	-1.6111880540847778E-007	5.1081531120389518E-014
Z_0	-8815210.5483045448	9.6857547760009766E-008	-1.0987547855976917E-014
$\dot{X}_0$	2296.0662918986295	-7.6852302299812436E-011	-3.3471290690069875E-014
$\dot{Y}_0$	3944.6910449259826	-4.0927261579781771E-011	-1.0375276824892454E-014
$\dot{Z}_0$	-3449.9126329548994	-7.9580786405131221E-011	2.3067478765968335E-014

This third scenario test shows the DC process does indeed recover the unknown scenario parameters almost to the limits of 64-bit machine precision. Only one to maybe one and a half digits have been lost. Realistic units and models have been used, which are seen to entail more machine arithmetic. In addition, the number of observations has been extended from n=10 to n=5768. The results show initial positions are recovered with only 18 to 161 nanometers of error. This is a completely successful solution that does not show any evidence of mathematical difficulty. This scenario re-validates the findings of Section 8 regarding least squares, standard application of DC, and procedures for variational equations of the underlying differential equations of motion.

The output results of "`r`" and "`v`" show some statistics of the range and range rate misclosures, **L**, for each iteration. Of interest are the maximum and minimum instances. The epoch numbers tend to values in the high 600's to low 700's. This is showing extrema are occurring near the end of the 3¾ revolution orbit. This is evidence of minor error in the DDEABM integration of the equations of motion. (Recall our perfect data, $\mathbf{L}_b$, are from a Keplerian exact analytic model, while our iterated, computed data, $\mathbf{L}_0$, are from DDEABM state vector values.) Even so, the misclosure extrema iterate to the micrometer level, and the post-fit residual extrema are likely an order of magnitude better.

10. Kepler Model - - II

In this section, we explore a variant of the scenario of the prior section. The standard gravitational parameter of the Earth, GM, is still considered constant. But now GM is treated as an unknown parameter to be solved. The number of unknowns become $u = 7$. The DC unknown *state parameters* are still $(X_0, Y_0, Z_0, \dot{X}_0, \dot{Y}_0, \dot{Z}_0)^t$. The number of state parameters remains $n_s = 6$. However, the standard gravitational parameter, GM, is not described by a DE. Certainly, it participates in the force model and the associated DEs. But GM has no kinematics, no trajectory; it is a *force parameter*. The number of force parameters is $n_p = 1$.

We retain the central force equations of motion, and the first-order DE system of satellite motion of the prior section. We retain the equations defining range and range rate observations, the defining quantities of the Kepler Scenario, and the perfect range and range rate data. We keep the range and range rate sigmas, dispersion and weight matrices.

We retain the method of obtaining perturbed initial values for our state vector. We also perturb the defining value of GM = 398603000000000.00 by multiplying by the factor of 1.00000001. This gives the starting estimate of the unknown parameters, $\mathbf{X}_0$, displayed in Table 10.1.

Table 10.1 – Perturbed Initial Values and Perturbation

X_0	-7856420.4697193988	15.734388513490558
Y_0	-3154119.6024935995	30.280538512393832
Z_0	-8815237.0415215995	-26.493216957896948
$\dot{X}_0$	2296.0784583470122	1.2166448305833910E-002
$\dot{Y}_0$	3944.6967362488972	5.6913228736448218E-003
$\dot{Z}_0$	-3449.8975864829749	1.5046471844925691E-002
GM	398603003986030.00	3986030.00

Note that GM is a force parameter, and all force model and observation model parameters are defined as unknown parameters in the least squares problem.

Since this variant has a force model parameter, the variational DE is now the combined form holding both the state transition matrix and the sensitivity matrix (Montenbruck and Gill, 2000, Eq. 7.45) reproduced above. We integrate both matrices as the combined matrix, $(\mathbf{\Phi}|\mathbf{S})$; which is a n_s x $(n_s + n_p)$ matrix where $n_s = 6$ where $n_p = 1$. Note that the leftmost 6x6 matrix in the homogenous part is unchanged from the variational DE in the prior scenario. The fact that the standard gravitational parameter, GM, is now an unknown has no bearing on the partials ∂**a**/∂**r** and ∂**a**/∂**v**. We retain the quantities established in Section 9. Note the 3x1 submatrix in the lower right-hand corner of the inhomogenous term becomes

$$\frac{\partial \mathbf{a}}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial \ddot{X}}{\partial GM} \\ \frac{\partial \ddot{Y}}{\partial GM} \\ \frac{\partial \ddot{Z}}{\partial GM} \end{pmatrix} = \begin{pmatrix} -\frac{X}{\rho^3} \\ -\frac{Y}{\rho^3} \\ -\frac{Z}{\rho^3} \end{pmatrix}$$

since GM is now a force parameter. The full variational DE system is now written as

$$\frac{d}{dt}(\mathbf{\Phi}|\mathbf{S}) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{GM}{r^5}(3X^2 - r^2) & \frac{GM}{r^5}3XY & \frac{GM}{r^5}3XZ & 0 & 0 & 0 \\ \frac{GM}{r^5}3YX & \frac{GM}{r^5}(3Y^2 - r^2) & \frac{GM}{r^5}3YZ & 0 & 0 & 0 \\ \frac{GM}{r^5}3ZX & \frac{GM}{r^5}3ZY & \frac{GM}{r^5}(3Z^2 - r^2) & 0 & 0 & 0 \end{pmatrix}(\mathbf{\Phi}|\mathbf{S})$$
$$+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{X}{\rho^3} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{Y}{\rho^3} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{Z}{\rho^3} \end{pmatrix}$$

These are the 42 variational DEs integrated along with the 6 state vector DEs. As per DC standard practice, the STM initial conditions are: $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{6x6}$, and the 6x1 sensitivity matrix initial conditions are $\mathbf{S}(t_0) = \mathbf{0}$.

The observation partial derivatives are unchanged from the prior section. Note that any given row of $\mathbf{H}_i$ is a 1x7 row. Those partials are converted to a row of partial derivatives with respect to the full set of unknowns, both state parameters and force model parameters, by matrix multiplication with the combined matrix:

$$\mathbf{A}_i = \mathbf{H}_i\,(\mathbf{\Phi}|\mathbf{S})$$

Note that where $\mathbf{H}_i$ is a 1x6 row, $\mathbf{A}_i$ is now a 1x7 row. We have gained the needed partial derivatives with respect to our force model parameters by the conversion.

This variant scenario is designated kepdc2. It incorporates Newtonian gravitation, solves for a point mass, and has familiar units for the observations and parameters. This variant scenario is cycled for 5 loops, 4 iterations. The results are now displayed.

```
program kepdc2 -- 2022jul17
 l.s. solve Kepler+GM orbit by differential correction
 unknown GM, ddeabm() variational equations

  imperfect initial statevector
  rtol (all) =    9.9999999999999998E-013
  nep,nobs =          721        5768
 tracking stations --
          1   3678298.5650071050        5201899.7170905434        0.0000000000000000
          2   3678298.5650071050       -5201899.7170905434        0.0000000000000000
          3  -3678298.5650071050        0.0000000000000000        5201899.7170905434
          4  -3678298.5650071050        0.0000000000000000       -5201899.7170905434

   perfect GM=    398603000000000.00
 imperfect GM=    398603003986030.00

 perfect kepler elements --
 semi-major axis     =  12267692.6 m.
 eccentricity        =    0.003845 deg.
 inclination         =  109.85396 deg.
 ascending node      =   43.95923 deg.
 argument of perigee =  245.07169 deg.
 mean anomaly        =   55.20345 deg.
 kep. period s/m/h=    13522.404232887979        225.37340388146632
3.7562233980244386

 non-perfect kepler element factor=   1.0000001000000001
 non-perfect kepler elements --
 semi-major axis     =  12267693.8 m.
 eccentricity        =    0.003845 deg.
 inclination         =  109.85397 deg.
 ascending node      =   43.95923 deg.
 argument of perigee =  245.07171 deg.
 mean anomaly        =   55.20346 deg.

 perfect statevector --
          1  -7856436.2041079123
          2  -3154149.8830321119
          3  -8815210.5483046416
          4   2296.0662918987064
          5   3944.6910449260236
          6  -3449.9126329548199

 non-perfect statevector and diff. with perfect --
          1  -7856420.9656723104        15.238435601815581
          2  -3154120.4545480874        29.428484024479985
          3  -8815236.2963436693       -25.748039027675986
          4   2296.0780993349022        1.1807436195795162E-002
          5   3944.6966072306241        5.5623046005166543E-003
          6  -3449.8980194407363        1.4613514083521295E-002

 top of perfect sat pos. --
          0  -7856436.2041079123       -3154149.8830321119       -8815210.5483046416
          1  -7568705.4140116964       -2676077.8998624440       -9215129.1542818397
          2  -7257182.4227614542       -2189593.7004120979       -9586079.9795341603
          3  -6922843.8984359335       -1696225.7544806539       -9926893.5148273204
          4  -6566739.2481008712       -1197524.7150596599       -10236495.860428371
          5  -6189987.2274864251       -695058.38216115907       -10513912.248329705
          6  -5793772.3135582767       -190406.60605464038       -10758270.229454534
          7  -5379340.8531783465        314843.85137669696       -10968802.513910934

 top of perfect data --
   0.0000000000000000        16750570.382064076
```

```
   0.0000000000000000        14661218.318668425
   0.0000000000000000        14962783.023493638
   0.0000000000000000        6360936.3130651973
   0.0000000000000000       -1733.3644659799313
   0.0000000000000000        818.82219574949602
   0.0000000000000000        1759.1899813539981
   0.0000000000000000       -1504.4673676341772
 range and range rate sigmas =    1.0000000000000000E-003

 r ave/std/rms  ->   1.1372279918445183        6.3681006140452707
6.4677609468103192
 r max/iep/ista ->   21.470123112201691               22           4
 r min/iep/ista ->  -19.453779188916087               20           1
 v ave/std/rms  ->   4.8251823124220039E-006   3.4118523403745484E-003
3.4112641877374534E-003
 v max/iep/ista ->   2.0736164939933133E-002           4           4
 v min/iep/ista ->  -1.3601585696648044E-002         716           1
 x()=   -15.238479356883758       -29.428510395357719        25.747989359535953       -
1.1807423612165979E-002  -5.5622802259434993E-003  -1.4613534299959124E-002
 delta GM:   -3986015.5650691986
 finished iteration:           0
 r ave/std/rms  ->   2.6461857709460137E-004   2.9528123218199604E-003
2.9641356805530619E-003
 r max/iep/ista ->   8.6164940148591995E-003         662           4
 r min/iep/ista ->  -8.3984583616256714E-003         699           4
 v ave/std/rms  ->  -1.7286273849769573E-009   1.5433248792909753E-006
1.5430582576231316E-006
 v max/iep/ista ->   2.4868159016477875E-006         662           1
 v min/iep/ista ->  -8.2461547918910583E-006         681           4
 x()=    4.3618838642567326E-005   2.5675056770396720E-005   5.0506915530892059E-005  -
1.2862164660081644E-008  -2.4446691143077627E-008   1.9778900396086492E-008
 delta GM:   -13.870271699503064
 finished iteration:           1
 r ave/std/rms  ->  -2.4604806146906088E-008   1.0953503769937341E-006
1.0954368206641677E-006
 r max/iep/ista ->   4.1350722312927246E-006         505           1
 r min/iep/ista ->  -3.8333237171173096E-006         580           1
 v ave/std/rms  ->  -2.0600985087960339E-012   5.3313496034592555E-010
5.3304650346541461E-010
 v max/iep/ista ->   3.1060807259564172E-009         568           4
 v min/iep/ista ->  -1.3527596820495091E-009         504           4
 x()=   -1.3107690211991505E-007  -1.3877875958492620E-007   4.8447909602093755E-008  -
7.9232322238528933E-011  -6.6708365390814019E-011   1.9379327939532491E-012
 delta GM:   -6.2805873062634419
 finished iteration:           2
 r ave/std/rms  ->  -9.9619545672869055E-008   9.4216384683994084E-007
9.4725338530353797E-007
 r max/iep/ista ->   4.0698796510696411E-006         668           4
 r min/iep/ista ->  -5.2712857723236084E-006         399           1
 v ave/std/rms  ->  -6.1597124380645019E-012   4.9194891010240025E-010
4.9190218154940237E-010
 v max/iep/ista ->   1.6482051279353982E-009         399           4
 v min/iep/ista ->  -3.3682852063066093E-009         715           1
 x()=   -9.4389705912624179E-007  -1.4082136073961525E-006   1.0178985580172899E-006  -
5.8353550599933299E-010  -1.9225945152530135E-010  -7.2658698726726295E-010
 delta GM:    21.900605641877519
 finished iteration:           3
 r ave/std/rms  ->   5.9651469246028359E-008   6.5821788935013959E-007
6.6080168128453716E-007
 r max/iep/ista ->   3.3508986234664917E-006         399           3
 r min/iep/ista ->  -5.2917748689651489E-006         399           1
 v ave/std/rms  ->  -1.2168548685253907E-011   3.6280498940366619E-010
3.6294613005751154E-010
 v max/iep/ista ->   1.6499956956295136E-009         399           4
 v min/iep/ista ->  -1.6985950423986651E-009         713           1
 x()=    6.4740667751572142E-007   8.3654785291528591E-007  -4.4048129482054058E-007
3.4727733673194651E-010   1.8369417687146956E-010   2.8542620084413858E-010
 delta GM:   -5.1697455368587271
 finished iteration:           4
 xa()=   -7856436.2041084776       -3154149.8830335182       -8815210.5483031757
2296.0662918981129        3944.6910449258758       -3449.9126329556966
```

```
 err.   =   -5.6531280279159546E-007  -1.4062970876693726E-006   1.4659017324447632E-006
-5.9344529290683568E-010  -1.4779288903810084E-010  -8.7675289250910282E-010
 relerr.=    7.1955373671335748E-014   4.4585613868085649E-013  -1.6629231082026602E-013
-2.5846174171917864E-013  -3.7466277423222751E-014   2.5413770891878258E-013
 GM  =    398603000000011.00
 err.   =    11.000000000000000
 relerr.=    2.7596380358401719E-014

 end of processing
```

Once again, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters of this scenario variant that now includes a force parameter and a sensitivity matrix. This comparison is provided in Table 10.2.

Table 10.2 – Adjusted Unknowns, Error, and Relative Error (`kepdc2`)

X_0	-7856436.2041084776	-5.6531280279159546E-007	7.1955373671335748E-014
Y_0	-3154149.8830335182	-1.4062970876693726E-006	4.4585613868085649E-013
Z_0	-8815210.5483031757	1.4659017324447632E-006	-1.6629231082026602E-013
$\dot{X}_0$	2296.0662918981129	-5.9344529290683568E-010	-2.5846174171917864E-013
$\dot{Y}_0$	3944.6910449258758	-1.4779288903810084E-010	-3.7466277423222751E-014
$\dot{Z}_0$	-3449.9126329556966	-8.7675289250910282E-010	2.5413770891878258E-013
GM	398603000000011.00	11.000000000000000	2.7596380358401719E-014

This fourth scenario test again shows the DC process does recover the unknown scenario parameters, which now includes a standard gravitational parameter, GM. The initial positions are recovered to 0.5 to 1.4 micrometers of accuracy. All quantities are near the limits of machine precision. These results demonstrate that the standard application of variational equations in orbit determination (OD), including force parameters and the sensitivity matrix, is valid.

11. Ordinary Differential Equation Background

This author now briefly relates ordinary differential equation (ODE) mathematical theory. An expanded version of this material can be found in Milbert and Jekeli (2023, Section 4). The context for the mathematics is orbit determination (OD). Thus, it is reasonable to expect benign models that support conditions for existence and uniqueness of ODE solutions.

An advanced text used for graduate studies in mathematics is the classic work by Coddington and Levinson (1955), which is now denoted CL55. We closely follow their exposition and notation, and describe their work. CL55 is important, since they derive the initialization of the state transition and sensitivity matrices.

They begin with the first order ODE

$$x' = f(t, x)$$

where t is the independent variable, and x is a variable of function f. This is expanded to an n-dimensional *system* of first order ODE's:

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$$

or, with indices:

$$x'_i = f_i\,(t, x_1, \ldots, x_n) \qquad\qquad (i = 1, \ldots, n)$$

General solutions, φ and $\boldsymbol{\varphi}$, are all sets of all points, $(t, \varphi(t))$ and $(t, \boldsymbol{\varphi}(t))$. The general solutions vary due to unresolved constants of integration, and form an infinite set of integral curves. In those sets there is only one solution that passes through a point, (t, x) and $(t, \boldsymbol{x})$. That specific point is named (τ, ξ) and $(\tau, \boldsymbol{\xi})$, where $\varphi(\tau) = \xi$ and $\boldsymbol{\varphi}(\tau) = \boldsymbol{\xi}$. The initial value problem (IVP) is defined as the ODE, $x' = f(t, x)$ and $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$, *and* the initial value, $x(\tau) = \xi$ and $\boldsymbol{x}(\tau) = \boldsymbol{\xi}$, respectively. (Earlier in this report, t_0 was used for τ, the initial time.) The initial value, ξ and $\boldsymbol{\xi}$, is a defining quantity of an IVP. CL55 establish conditions for existence in Chapter 1, Section 1 (single ODE) and Section 5 (system). Also, CL55 establish conditions for uniqueness (Lipschitz condition) in Chapter 1, Section 2 (single ODE) and Section 5 (system). The CL55 uniqueness theorems prove an IVP has only one solution passing through a specific (t, x) and $(t, \boldsymbol{x})$. The defining initial value, ξ and $\boldsymbol{\xi}$, at τ, resolve any constants of integration, and collapse the infinite sets of general solutions (the set of integral curves) down to a specific orbit.

When first order ODE system variables are defined as derivatives of another system variable, it allows the ODE system to represent higher order ODE's. CL55 (Chapter 1, Section 6) show that arbitrarily high order derivatives may be represented by such expressions. In OD, for a body subject to forces, the accelerations (x'') in a 3-D space are mapped into $n = 2 \times 3 = 6$ variables.

To address partial derivatives, CL55 (Chapter 1, Section 7) elaborate their notation to distinguish between n-dimensional general solutions, $\boldsymbol{\varphi}$, and n-dimensional specific solutions, $\boldsymbol{\varphi}(t, \tau, \boldsymbol{\xi})$. The initial value of a specific n-dimensional solution is denoted, $\boldsymbol{\varphi}(\tau, \tau, \boldsymbol{\xi})$. In that section they also prove existence and continuity of $\partial\boldsymbol{\varphi}/\partial\boldsymbol{\xi}_i$. Further, (CL55, Chapter 1, eq. 7.12) shows

$$\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\xi}_j}(\tau, \tau, \boldsymbol{\xi}) = e_j$$

where e_j is the vector with all components zero, except the j-th component is 1. The full system

$$\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\xi}}(\tau, \tau, \boldsymbol{\xi}) = \mathbf{I}_{n \times n}$$

equals an $n \times n$ identity matrix, $\mathbf{I}_{n \times n}$. The matrix $\partial\boldsymbol{\varphi}/\partial\boldsymbol{\xi}$ is the state transition matrix (STM) of Section 5 in this report. When the first parameter is τ, the equation above refers to initial time. The indexed result above is the CL55 expression of the initial condition of an STM, $\boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}_{n \times n}$, (Montenbruck and Gill, 2000, eq. 7.42).

After the notation change, CL55 (Chapter 1, Section 7) extend the n-dimensional system of first order ODE's to

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{\mu})$$

where $\boldsymbol{\mu}$ is a k dimensional parameter vector. The solutions satisfy $\boldsymbol{\varphi}(\tau, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}) = \boldsymbol{\xi}$. The IVP is defined as the ODE, $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{\mu})$ and the initial values $\boldsymbol{x}(\tau) = \boldsymbol{\xi}$. The parameter vector, $\boldsymbol{\mu}$, can, for example, represent force model parameters in OD. Note that the dependent variables, $\boldsymbol{x}$, and parameters, $\boldsymbol{\mu}$, are distinct. The IVP solution is $\boldsymbol{\varphi}(t, \tau, \boldsymbol{\xi}, \boldsymbol{\mu})$. Existence and uniqueness proofs for the parametric IVP are in CL55 (Chapter 1, Theorem 7.4). Their uniqueness result proves the presence of $\boldsymbol{\mu}$ does not alter the fact that a unique ODE system solution is still established by n initial values, $\boldsymbol{\xi}$, at an initial time, τ. And, those defining initial values, $\boldsymbol{\xi}$, do not depend on $\boldsymbol{\mu}$.

CL55 take the IVP: $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{\mu})$ with initial values, $\boldsymbol{x}(\tau) = \boldsymbol{\xi}$. By the Second Fundamental Theorem of Calculus, CL55 write the parametric ODE solution,

$$\boldsymbol{\varphi}(t, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}) = \boldsymbol{\xi} + \int_{\tau}^{t} \boldsymbol{f}(s, \boldsymbol{\varphi}(s, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}), \boldsymbol{\mu})\, \mathrm{d}s \tag{11.1}$$

Then they take the derivative with respect to $\boldsymbol{\mu}_j$ (where j ranges from 1 to k),

$$\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\mu}_j}(t, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}) = \int_{\tau}^{t} \left[\boldsymbol{f}_{\boldsymbol{x}}(s, \boldsymbol{\varphi}(s, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}), \boldsymbol{\mu}) \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\mu}_j}(s, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\mu}_j}(s, \boldsymbol{\varphi}(s, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}), \boldsymbol{\mu}) \right] \mathrm{d}s \tag{11.2}$$

where $\boldsymbol{f}_{\boldsymbol{x}}$ are partial derivatives, $\partial \boldsymbol{f} / \partial \boldsymbol{x}$. When writing (11.2), the initial value term, $\boldsymbol{\xi}$, in (11.1) becomes zero because $\boldsymbol{\xi}$ are defined quantities. They do not vary with the parameters, $\boldsymbol{\mu}$. CL55 then define $\boldsymbol{y} = \partial \boldsymbol{\varphi} / \partial \boldsymbol{\mu}_j$ as the j-th column of all the partials, and take the derivative with respect to time. This leads to the variational IVP:

$$\boldsymbol{y}' = \boldsymbol{f}_{\boldsymbol{x}}(t, \boldsymbol{\varphi}(t, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}), \boldsymbol{\mu}) \boldsymbol{y} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\mu}_j}(t, \boldsymbol{\varphi}(t, \tau, \boldsymbol{\xi}, \boldsymbol{\mu}), \boldsymbol{\mu}) \qquad \boldsymbol{y}(\tau) = \mathbf{0} \tag{11.3}$$

In (11.3) the initial condition, $\boldsymbol{y}(\tau) = \mathbf{0}$, is required because, at the initial time, (11.2) will integrate from τ to τ, and collapse to zero.

Each $\boldsymbol{y} = \partial \boldsymbol{\varphi} / \partial \boldsymbol{\mu}_j$, is a column of the sensitivity matrix, $\partial \boldsymbol{\varphi} / \partial \boldsymbol{\mu} = \mathbf{S}(t)$. For comparison with (11.3), equation (7.44) by Montenbruck and Gill (2000) is repeated

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{S}(t) = \begin{pmatrix} \mathbf{0}_{3\mathrm{x}3} & \mathbf{I}_{3\mathrm{x}3} \\ \dfrac{\partial \mathbf{a}(t, \mathbf{r}, \mathbf{v}, \mathbf{p})}{\partial \mathbf{r}(t)} & \dfrac{\partial \mathbf{a}(t, \mathbf{r}, \mathbf{v}, \mathbf{p})}{\partial \mathbf{v}(t)} \end{pmatrix}_{6\mathrm{x}6} \mathbf{S}(t) + \begin{pmatrix} \mathbf{0}_{3\mathrm{x}n_p} \\ \dfrac{\partial \mathbf{a}(t, \mathbf{r}, \mathbf{v}, \mathbf{p})}{\partial \mathbf{p}} \end{pmatrix}_{6\mathrm{x}n_p}$$

CL55 (Chapter 1, Section 7), derived the correct initial conditions, $\boldsymbol{y}(\tau) = \mathbf{0}$, for each $\partial \boldsymbol{\varphi} / \partial \boldsymbol{\mu}_j$ variational equation; and, by extension, zero for *all* the initial values of the sensitivity matrix, $\mathbf{S}(t_0) = \mathbf{0}$. CL55 prove the valid mathematical basis for IVP's with parameters, $\boldsymbol{\mu}$ and their associated variational equations. To emphasize, the initial values of the sensitivity matrix are all zeros because a parametric IVP unique solution has, by definition, specified initial values, $\boldsymbol{\xi}$, at the initial time, τ; and, because the integral in (11.2) collapses to zero at the initial time, $t = \tau$.

12. Orbit Determination Literature

As a cautionary admonition, if one is researching variational equations in DC, there are references that need to be read with care. By all means, read these references if you are so inclined. Critical examination, discussion, and testing are the engines that drive Scientific Progress.

Riley et al. (1967) suffers from inexact terminology. This reference is a reason why such rigor in expression was made in this study. They combine the *state parameters* and *force model parameters*, denote them β_k, and call them "initial values" or "one of the differential equation parameters" (ibid., pg. 12). To follow (ibid.), it is useful to refer to the equations of Cappellari et al. (1976, Eq. 4-2 to 4-8) or Long et al. (1989, Eq. 4-2 to 4-8). The clearest map of the variational equations is the block matrix notation of Montenbruck and Gill (2000, Eq. 7.45). By Riley et al. (1967, pg. 13 top):

"The initial values are determined by differentiating the initial values of Y with respect to β_k and in general will be zero if β_k is a differential equation parameter (i.e., a parameter occurring in the function F)."

The first "initial values" above is referring to initial values of the sensitivity matrix, $\mathbf{S}(t_0)$. The "initial values of Y" are, in fact, referring to the state vector, $\mathbf{X}(t)$. Once the partials $\partial\mathbf{X}(t)/\partial\mathbf{p}$ are formed, which is the sensitivity matrix, $\mathbf{S}(t)$, then it can be evaluated at t_0. The phrase "in general will be zero" is incorrect. There is nothing general about it. By standard DC practice, the initial values of the sensitivity matrix are $\mathbf{S}(t_0) = \mathbf{0}$.

Continuing (ibid.),

"On the other hand, if β_k denotes an initial value, then not all of the initial conditions on $Y_{\beta k}$ will be zero, …"

The phrase "…not all the initial conditions on $Y_{\beta k}$ will be zero,…" is correct as far as it goes. These "initial conditions" are referring to the initial conditions of the STM. By standard DC practice, the initial values of the sensitivity matrix are $\boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}$. Clearly, the Montenbruck and Gill (2000) terminology is more rigorous.

Continuing Riley et al. (1967),

"… but normally $\partial F/\partial\beta_k$ will be the null vector."

On its face, $\partial F/\partial\beta_k$ is frustrating, because (ibid.) earlier lumped the state and force model parameters together. But in the context of (ibid., second equation, pg. 12), those partials match Cappellari et al. (1976) and Long et al. (1989) equations 4-8c. The $C(t)$ in equation 4-8c is *not* the null vector if there exist any force model parameters at all.

To be fair, Riley et al. (1967), is a short 5¼ page note. Their description of variational equations is a too brief sketch supported by only 2 equations. The thrust of their paper was not to thoroughly describe variational equations, but to demonstrate efficient, two-part integrations of the state vector and the variational equations.

A series of references that must be read with care are Xu (2009), Xu (2015), Xu (2018), and Xu (2021). These assert that the standard application of variational equations in orbit determination (OD), as well as in other applications, does not have a sound mathematical basis. Xu (2021) goes so far as to state that solutions have been "incorrectly solved for 100 years." These assertions are based on a flawed proof that the sensitivity matrix, $\mathbf{S}(t_0) \neq 0$.

The completely general, mathematical proofs by Coddington and Levinson (1955, Chapter 1, Section 7) regarding the primary initial value problem (IVP) for a system of ODE's with parameters, and the associated IVP for the variational equation for columns of the sensitivity matrix, $\mathbf{S}(t)$, prove that, in fact, $\mathbf{S}(t_0) = 0$. (As related in Section 11 of this study, all initial values of the sensitivity matrix are zero because a parametric IVP unique solution has, by definition, specified initial values at the initial time, and because the integral expressing the columns of $\mathbf{S}(t)$ collapses to zero when $t = \tau$.)

The Xu (2018) claim (above) is based on a flawed proof by contradiction. The stated counter example equation (ibid., eq. 11) is

$$\ddot{y} + {p_1}^2 y - p_2 \cos(p_1 t) = 0 \qquad (12.1)$$

The proposed general solution is (ibid., eq.12)

$$y^?(t) = c_1 \sin(p_1 t) + \frac{c_2}{p_1} \cos(p_1 t) + \frac{p_1 p_2 t \sin(p_1 t) + p_2 \cos(p_1 t)}{2{p_1}^2} \qquad (12.2)$$

The fundamental structural flaw is that (ibid.) never forms an initial value problem (IVP). Recall that the differential correction (DC) method in orbit determination (OD) requires an IVP. A valid proof (or test) of the DC method must first encompass the specific solution developed by an IVP. Further, any supposed proof relying on contradiction is suspect, since an ODE general solution describes an infinite number of integral curves.

The proofs by Coddington and Levinson (1955, Chapter 1) are completely adequate on these points. But, to further emphasize the flawed proof, we will proceed by deriving the analytic solutions for the initial value problems for (12.1). In contrast to (Xu, 2018), derivations from the IVPs will show the initial value of the sensitivity matrix, $\mathbf{S}(t_0) = \mathbf{0}$. Scenarios will be constructed, and numerical solutions will be demonstrated that achieve the limits of computer precision, as done earlier in this study.

Please note the second part of (Xu, 2018) proposes a perturbation theory. Our report does not test nor judge that perturbation theory in any way, shape, or form. For analysis of that method, refer to Jekeli and Habana (2019) and Habana (2020).

13. Classic Harmonic Oscillator

The differential equation of interest (12.1) is

$$\ddot{y} + {p_1}^2 y - p_2 \cos(p_1 t) = 0 \tag{13.1}$$

Write (13.1) in an inhomogenous form

$$\ddot{y} + {p_1}^2 y = p_2 \cos(p_1 t) \tag{13.2}$$

This equation is an old friend; the classic harmonic, undamped, forced oscillator. There are countless references in mathematics, analytical mechanics, classical dynamics, and engineering that address this problem, such as Marion (1965, Chapter 6 and Appendix C), Fowles and Cassiday (2005, Chapter 3), and Feynman et al. (1963, Chapter 21.) (Yes, this is the Nobel laureate in Physics and admired educator, Prof. Richard Feynman). Of particular interest is that in (13.2) the *forcing function* on the right has the same frequency, p_1, as the frequency of the simple oscillator on the left. This will cause resonance. And, since there is no damping term, the steady-state oscillation, $t \to \infty$, will have an infinite amplitude. For example, see Feynman et al. (1963, pg. 21-11). Despite this issue, we can proceed with analysis of the transient response for small values of t.

The right-hand side of (13.2) is a form, $F(t) \neq 0$. This is a linear *inhomogenous* DE (Marion 1965, Appendix C.2). The general solution is a superposition of the solution to the reduced (or associated) *homogenous* DE (where $F(t) = 0$), which is called the *complementary solution*, $y_c(t)$; and any possible solution to (13.2), which is called the *particular solution*, $y_p(t)$. Two methods of finding a particular solution to an inhomogenous DE are Undetermined Coefficients and Variation of Parameters. Don't be confused by nomenclature here. Both the *complementary* and *particular* solutions are still referring to a general ODE solution with a multiplicity of integral curves (Section 11).

Consider the homogenous ($F(t) = 0$) DE form of (13.2). This is the simple (unforced) harmonic oscillator

$$\ddot{y} + {p_1}^2 y = 0 \tag{13.3}$$

The general solution of (13.3) is

$$y(t) = c_1 \sin(p_1 t) + c_2 \cos(p_1 t) \tag{13.4}$$

where c_1 and c_2 are constants of integration (Feynman et al., 1963, eq. 21.6c or Blanchard et al., 2012, pg. 415, last equation). This is the complementary solution, $y_c(t)$ of the simple (unforced) harmonic oscillator.

In a proposed general solution (Xu, 2018, eq.12)

$$y^{?}(t) = c_1 \sin(p_1 t) + \frac{c_2}{p_1}\cos(p_1 t) + \frac{p_1 p_2 t \sin(p_1 t) + p_2 \cos(p_1 t)}{2{p_1}^2} \tag{13.5}$$

we see a solution of the *homogenous part* of (13.5) as

$$y^{?}(t) = c_1 \sin(p_1 t) + \frac{c_2}{p_1}\cos(p_1 t) + \cdots \tag{13.6}$$

This does not conform a general solution to the homogenous part, as seen in (13.4). The term

$$\ldots + c_2 \frac{\cos(p_1 t)}{p_1} + \cdots$$

has an issue. In a physical context, variable, t, has units of time; so, the parameter, p_1, must have units of inverse time (or frequency). When p_1 appears in the denominator, this prevents c_2 from being arbitrary. The dimensional units of c_2 (above) now depend upon the dimensional units of p_1. The solution posed in (ibid., eq.12) is denoted as $y^{?}(t)$ due to its lack of correspondence with the general solution of the classic harmonic oscillator (13.4) (Feynman et al., 1963, eq. 21.6c or Blanchard et al., 2012, pg. 415, last equation). Rather than further examining a nonstandard general solution at this point, it is more constructive to provide detailed solutions in the following two sections. Equation (12.2) will be addressed in Section 15.

14. Simple Harmonic Oscillator - - Theory

The simple harmonic oscillator has a DE

$$\ddot{y} + {p_1}^2 y = 0 \tag{14.1}$$

where a model parameter, p_1, has units of inverse time (or frequency), and the variable, y, has units of length. We may immediately write a system of first-order DEs of motion by inspection

$$\begin{aligned} \dot{\mathrm{Y}}(t) &= \mathrm{V}_y \\ \dot{\mathrm{V}}_y(t) &= -{p_1}^2 y \end{aligned} \tag{14.2}$$

where $\dot{\mathrm{V}}_y$ is acceleration of y. This specifies two state parameters, $(y, \dot{y})$.

The oscillator DE (14.1) is solved by the Maxima computer algebra system (Maxima.sourceforge.io, 2022). In fact, all subsequent derivations are obtained through Maxima. The general solution of the simple harmonic oscillator (14.1) is

$$y(t) = c_1 \sin(p_1 t) + c_2 \cos(p_1 t) \tag{14.3}$$

where c_1 and c_2 are constants of integration. Note that (14.3) obeys the accepted general solution form (Feynman et al., 1963, eq. 21.6c, Blanchard et al., 2012, pg. 415, last equation); and, so, also qualifies as a complementary solution. The velocity of y is $\partial y/\partial t$

$$\dot{y}(t) = c_1\, p_1 \cos(p_1 t) - c_2\, p_1 \sin(p_1 t) \qquad (14.4)$$

Following the requirements of the differential correction (DC) method for OD, we now find the specific solution to the initial value problem (IVP) for the ODE (14.1). We define initial values at t_0=0 for our two state parameters, and denote them as (y_0, $\dot{y}_0$). The procedure is quite simple. Following Feynman et al. (1963, section 21-4), y_0 and $\dot{y}_0$ are set to the left-hand sides of 14.3 and 14.4 respectively; time t is set to zero; and c_1 and c_2 are solved

$$c_1 = \frac{\dot{y}_0}{p_1} \qquad c_2 = y_0 \qquad (14.5)$$

This immediately provides us with exact analytic solutions to the initial value problem of the simple harmonic oscillator DE. The position equation is

$$y(t) = \frac{\dot{y}_0}{p_1}\sin(p_1 t) + y_0 \cos(p_1 t) \qquad (14.6)$$

and the velocity equation (after a p_1 cancellation) is

$$\dot{y}(t) = \dot{y}_0 \cos(p_1 t) - y_0 p_1 \sin(p_1 t) \qquad (14.7)$$

Equation 14.6 will later be used to compute error-free ranges from a tracking station to the oscillator at time, t, as was done in the scenarios in Sections 7, 8, 9, and 10.

The equations 14.6 and 14.7 can be used to provide exact analytic solutions for the state transition matrix (STM), **Φ**, and the sensitivity matrix, **S**. For the STM

$$\boldsymbol{\Phi}(t, t_0) = \begin{pmatrix} \frac{\partial y(t)}{\partial y(t_0)} & \frac{\partial y(t)}{\partial \dot{y}(t_0)} \\ \frac{\partial \dot{y}(t)}{\partial y(t_0)} & \frac{\partial \dot{y}(t)}{\partial \dot{y}(t_0)} \end{pmatrix} = \begin{pmatrix} \cos(p_1 t) & \frac{1}{p_1}\sin(p_1 t) \\ -p_1 \sin(p_1 t) & \cos(p_1 t) \end{pmatrix} \qquad (14.8)$$

For the sensitivity matrix, **S**, the parameter, p_1, is treated as unknown, and is a force parameter

$$\mathbf{S}(t) = \begin{pmatrix} \frac{\partial y(t)}{\partial p_1} \\ \frac{\partial \dot{y}(t)}{\partial p_1} \end{pmatrix} = \begin{pmatrix} -y_0 t \sin(p_1 t) + \frac{\dot{y}_0}{p_1} t \cos(p_1 t) - \frac{\dot{y}_0}{{p_1}^2}\sin(p_1 t) \\ -y_0 \sin(p_1 t) - y_0 p_1 t \cos(p_1 t) - \dot{y}_0 t \sin(p_1 t) \end{pmatrix} \qquad (14.9)$$

The initial conditions for **Φ** and **S** are found by setting $t = 0$.

$$\boldsymbol{\Phi}(t_0, t_0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (14.10)$$

and

$$\mathbf{S}(t_0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{14.11}$$

Note that these derived initial conditions (14.10) and (14.11) conform to Coddington and Levinson (1955, Chapter 1), and Montenbruck and Gill (2000, pg. 240-241).

15. Forced Harmonic Oscillator - - Theory

The forced harmonic oscillator has a DE

$$\ddot{y} + {p_1}^2 y = p_2 \cos(p_1 t) \tag{15.1}$$

where a model parameter, p_1, has units of inverse time, and the variable, y, has units of length. Model parameter, p_2, has units of acceleration. We may immediately write a system of first-order DEs of motion by inspection

$$\begin{aligned} \dot{\mathrm{Y}}(t) &= \mathrm{V}_y \\ \dot{\mathrm{V}}_y(t) &= -{p_1}^2 y + p_2 \cos(p_1 t) \end{aligned} \tag{15.2}$$

where $\dot{\mathrm{V}}_y$ is acceleration of y. This specifies two state parameters, $(y, \dot{y})$.

The oscillator DE (15.1) is solved by Maxima (Maxima.sourceforge.io, 2022). The general solution of the forced harmonic oscillator is

$$y(t) = c_1 \sin(p_1 t) + c_2 \cos(p_1 t) + \frac{p_1 p_2 t \sin(p_1 t) + p_2 \cos(p_1 t)}{2{p_1}^2} \tag{15.3}$$

where c_1 and c_2 are constants of integration. As described in Section 13, the general solution (15.3) is a superposition of the complementary solution (14.3) and a particular solution. The general solution for the velocity of y is $\partial y/\partial t$

$$\dot{y}(t) = c_1\, p_1 \cos(p_1 t) - c_2\, p_1 \sin(p_1 t) + \frac{1}{2}\, p_2 t \cos(p_1 t) \tag{15.4}$$

We now provide initial values at t_0=0 for our two state parameters, and denote them as $(y_0, \dot{y}_0)$. Once again, following Feynman et al. (1963, section 21-4) and Section 14, y_0 and $\dot{y}_0$ are set to the left-hand sides of 15.3 and 15.4 respectively; time t is set to zero; and c_1 and c_2 are solved

$$\begin{aligned} c_1 &= \frac{\dot{y}_0}{p_1} \\ c_2 &= y_0 - \frac{p_2}{2{p_1}^2} \end{aligned} \tag{15.5}$$

This provides us with exact analytic solutions to the initial value problem of the forced harmonic oscillator DE. The position equation is

$$y(t) = \frac{\dot{y}_0}{p_1}\sin(p_1 t) + \left(y_0 - \frac{p_2}{2{p_1}^2}\right)\cos(p_1 t) + \frac{p_1 p_2 t \sin(p_1 t) + p_2 \cos(p_1 t)}{2{p_1}^2} \quad (15.6)$$

and the velocity equation is

$$\dot{y}(t) = \dot{y}_0 \cos(p_1 t) - \left(y_0 - \frac{p_2}{2{p_1}^2}\right) p_1 \sin(p_1 t) + \frac{1}{2} p_2 t \cos(p_1 t) \quad (15.7)$$

Equation 15.6 will later be used to compute error-free ranges from a tracking station to the oscillator, as was done in the scenarios in Sections 7, 8, 9, and 10.

The equations 15.6 and 15.7 can be used to provide exact analytic solutions for the state transition matrix (STM), $\mathbf{\Phi}$, and the sensitivity matrix, $\mathbf{S}$. For the STM

$$\mathbf{\Phi}(t, t_0) = \begin{pmatrix} \frac{\partial y(t)}{\partial y(t_0)} & \frac{\partial y(t)}{\partial \dot{y}(t_0)} \\ \frac{\partial \dot{y}(t)}{\partial y(t_0)} & \frac{\partial \dot{y}(t)}{\partial \dot{y}(t_0)} \end{pmatrix} = \begin{pmatrix} \cos(p_1 t) & \frac{1}{p_1}\sin(p_1 t) \\ -p_1 \sin(p_1 t) & \cos(p_1 t) \end{pmatrix} \quad (15.8)$$

Note that this result is identical to 14.8 for the simple harmonic oscillator. This is because there is no explicit functional dependence between $(y(t_0) \quad \dot{y}(t_0))$ and the particular solution of the inhomogenous part of 14.6 and 14.7.

For the sensitivity matrix, $\mathbf{S}$, the parameters, p_1 and p_2, are unknown force parameters

$$\mathbf{S}(t) = \begin{pmatrix} \frac{\partial y(t)}{\partial p_1} & \frac{\partial y(t)}{\partial p_2} \\ \frac{\partial \dot{y}(t)}{\partial p_1} & \frac{\partial \dot{y}(t)}{\partial p_2} \end{pmatrix} \quad (15.9a)$$

where

$$\frac{\partial y(t)}{\partial p_1} = -\frac{\dot{y}_0 \sin(p_1 t)}{{p_1}^2} + \frac{\dot{y}_0 t \cos(p_1 t)}{p_1} - \left(y_0 - \frac{p_2}{2{p_1}^2}\right) t \sin(p_1 t) - \frac{p_2 \cos(p_1 t)}{{p_1}^3} + \frac{p_2 \cos(p_1 t)}{{p_1}^3} - \frac{p_1 p_2 t \sin(p_1 t)}{{p_1}^3} \quad (15.9b)$$

$$\frac{\partial y(t)}{\partial p_2} = \frac{p_1 t \sin(p_1 t)}{2p_1} + \frac{\cos(p_1 t)}{2{p_1}^2} - \frac{\cos(p_1 t)}{2{p_1}^2} \quad (15.9c)$$

$$\frac{\partial \dot{y}(t)}{\partial p_1} = -\dot{y}_0 \sin(p_1 t) - \left(y_0 - \frac{p_2}{2{p_1}^2}\right)\sin(p_1 t) - \left(y_0 - \frac{p_2}{2{p_1}^2}\right) p_1 t \cos(p_1 t) - \frac{p_2 t^2 \sin(p_1 t)}{2} - \frac{p_2 \sin(p_1 t)}{{p_1}^2} \quad (15.9d)$$

$$\frac{\partial \dot{y}(t)}{\partial p_2} = \frac{1}{2}\left(\frac{\sin(p_1 t)}{p_1}\right) + t \cos(p_1 t) \quad (15.9e)$$

The initial conditions for $\mathbf{\Phi}$ and $\mathbf{S}$ are found by setting $t = 0$.

$$\mathbf{\Phi}(t_0, t_0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{15.10}$$

and

$$\mathbf{S}(t_0) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{15.11}$$

Note that the initial conditions (15.10) and (15.11) conform to Coddington and Levinson (1955, Chapter 1), and Montenbruck and Gill (2000, pg. 240-241). The derivations above contradict the arguments of Xu (2018) using the ODE proposed by Xu (2018). Thanks to a valuable contribution by Prof. Chris Jekeli (Milbert and Jekeli, 2023), he points out that $\partial c_1/\partial p_1$ and $\partial c_2/\partial p_1$ of Xu are not zero, as was assumed in Xu (2018). Further, Dr. Jekeli shows that when those partials are established, (12.2) does, in fact, lead to (15.11).

16. Simple Harmonic Oscillator - - I - - Known Force Parameter

In this section we create a new computational scenario based on the simple harmonic oscillator (SHO) theory of Section 14. The structure of the scenario is inspired by the Uniform Gravity Model scenarios of Sections 7 and 8. The oscillator is aligned along the Y-axis, where a fixed X=0. A tracking station with known, constant coordinates (X_s, Y_s) is defined. Observations from station to oscillator are range, ρ

$$\rho = \sqrt{(X - X_s)^2 + (Y - Y_s)^2}$$

As in the uniform gravity field model (UGFM, Section 7 and 8), the range is idealized (infinite speed of light, etc.).

We define the Simple Harmonic Oscillator Scenario:

$X_s = -1.0$
$Y_s = -1.0$
$Y_0 = 0.4$
$\dot{Y}_0 = 0.2$
$p_1 = 0.6$

As in the UGFM scenario, the quantities do not have specific units. However, Y, has length, $\dot{Y}$ has velocity, p_1 has frequency (inverse time), and t has time. Lack of specific units will not inhibit the test in any way.

The exact analytic theory of Section 14 and the defining scenario parameters are sufficient to generate ten epochs of range data at times 0 through 9, inclusive.

Table 16.1 – Perfect Range Data for Simple Harmonic Oscillator Scenario

0	1.7204650534085253
1	1.7867005209798366
2	1.8180709213909316

3	1.8108986696461278
4	1.7660231389977032
5	1.6887149431200152
6	1.5881127625767222
7	1.4761291700172752
8	1.3657475192551192
9	1.2687384331829761

Our least squares and differential correction (DC) setups give scenario characteristics. The number of observations, $n = 10$. We will treat our observations as uncorrelated. We arbitrarily choose $\sigma = 0.001$. There are two unknown parameters, (Y_0, $\dot{Y}_0$). We identify these unknowns as *state parameters*, as described earlier. Therefore, $u = n_s = 2$. For this first SHO scenario, p_1, is assumed known. This scenario has no unknown *force parameter*, $n_p = 0$.

As before, we begin with perturbed initial conditions as the starting estimate of the unknown parameters, $\mathbf{X}_0$:

$$Y_0 = 0.3$$
$$\dot{Y}_0 = 0.15$$

Since this scenario has no force model parameter, the variational DE consists solely of the state transition DE (Montenbruck and Gill, 2000, Eq. 7.42). Since there is no frictional drag, there is no spatial dependence of **a** on **v**; the lower right element is $\mathbf{0}_{1x1}$. However, there is spatial dependence of **a** on **r**; the lower left element is nonzero.

Although the SHO theory in Section 14 includes exact analytic forms for the state transition matrix (STM), $\mathbf{\Phi}(t, t_0)$, the variational equations will be numerically integrated. These are the system of 4 variational DEs integrated in time by DDEABM.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{\Phi}(t,t_0) = \begin{pmatrix} 0 & 1 \\ -{p_1}^2 & 0 \end{pmatrix} \mathbf{\Phi}(t,t_0)$$

As per DC standard practice, and as demonstrated in Section 14, the STM initial conditions are: $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{2x2}$.

This scenario is designated sho2, and was cycled for 5 loops, 4 iterations. The results are now displayed.

```
program sho2 -- 2022aug30
l.s. solve simple harmonic oscillator by differential correction
imperfect initial statevector
rtol,atol =    1.0000000000000000E-013   1.0000000000000000E-013
tfact =   0.50000000000000000
ns,np,nq,nsy=           2           0           0           6
nep,nobs =          10          10
xsta,ysta =   -1.0000000000000000       -1.0000000000000000
p1  =   0.59999999999999998
xag =   0.40000000000000002       0.20000000000000001
x0  =   0.30000000000000004       0.15000000000000002
perfect c1, c2 =   0.33333333333333337       0.40000000000000002
elb =   (ranges, unitless, perfect)
  1.7204650534085253
```

```
   1.7867005209798366
   1.8180709213909316
   1.8108986696461278
   1.7660231389977032
   1.6887149431200152
   1.5881127625767222
   1.4761291700172752
   1.3657475192551192
   1.2687384331829761
 range sigmas =    1.0000000000000000E-003

 r ave/std/rms ->  -5.6883479060797645E-002   5.1052572326614649E-002
7.4709161533964508E-002
 r max/iep     ->   3.4437052077616048E-002           9
 r min/iep     -> -0.10673870460340784                2
 x()=   0.10117981591776654        5.0773148429313311E-002
 finished iteration:           0
 r ave/std/rms ->   8.3597059056306211E-004   6.2177664130477067E-004
1.0231287312901403E-003
 r max/iep     ->   1.4532054162386121E-003           3
 r min/iep     ->  -3.1745532189231263E-004           9
 x()=   -1.1796453038345218E-003  -7.7300994031497829E-004
 finished iteration:           1
 r ave/std/rms ->   1.4335439402213268E-007   9.3440522870475387E-008
1.6854821407635834E-007
 r max/iep     ->   2.3915535485308226E-007           3
 r min/iep     ->  -3.4219407973878901E-008           9
 x()=   -1.7061392874558182E-007  -1.3848899151743730E-007
 finished iteration:           2
 r ave/std/rms ->   3.4638958368304883E-015   5.1387857322789174E-015
5.9803830387741573E-015
 r max/iep     ->   1.2878587085651816E-014           4
 r min/iep     ->  -3.7747582837255322E-015           9
 x()=   -3.7857999180465399E-015  -4.3787839367434787E-015
 finished iteration:           3
 r ave/std/rms ->  -8.6597395920762208E-016   3.6432310591584215E-015
3.5631068175681542E-015
 r max/iep     ->   5.9952043329758453E-015           4
 r min/iep     ->  -7.3274719625260332E-015           2
 x()=    8.7637142705308423E-017   1.6521952689216726E-017
 finished iteration:           4
 xa()=   0.39999999999999963       0.20000000000000245
 err.  =   -3.8857805861880479E-016   2.4424906541753444E-015
 relerr.=   -9.7144514654701197E-016   1.2212453270876722E-014

 end of processing
```

As before, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters. This comparison is shown below in Table 16.2.

Table 16.2 – Adjusted Unknowns, Scenario Parameters, and Error (`sho2`)

Y_0	0.39999999999999963	0.4000000000000000	-3.8857805861880479E-016
$\dot{Y}_0$	0.20000000000000245	0.2000000000000000	2.4424906541753444E-015

The DC process again recovers the unknown scenario parameters to the limits of 64-bit machine precision.

17. Simple Harmonic Oscillator - - II - - Unknown Force Parameter

In this section we explore a variant of the scenario of the prior section. The frequency parameter, p_1, is still considered constant. But now p_1 is treated as an unknown parameter to be solved. The number of unknowns become $u = 3$. The DC unknown *state parameters* are still $(\mathrm{Y}_0, \dot{\mathrm{Y}}_0)^t$. The number of state parameters remains $n_s = 2$. However, frequency, p_1, is not described by a DE. Certainly, it participates in the force model and the associated DEs. But, frequency, p_1, has no trajectory; it is a *force parameter*. The number of force parameters is $n_p = 1$.

We retain the SHO equations of motion, and the first-order DE system of Sections 14 and 16. We retain the equations defining range observations, the defining quantities of the Simple Harmonic Oscillator Scenario, and the perfect range data of Table 16.1. We keep the range sigma, dispersion, and weight matrices.

We choose perturbed starting estimates of the unknown parameters, $\mathbf{X}_0$:

$$\mathrm{Y}_0 = 0.3$$
$$\dot{\mathrm{Y}}_0 = 0.15$$
$$p_1 = 0.45$$

Since this variant has a force model parameter, the variational DE is now the combined form holding both the state transition matrix and the sensitivity matrix (Montenbruck and Gill, 2000, Eq. 7.45) reproduced above. We integrate both matrices as the combined matrix, $(\mathbf{\Phi}|\mathbf{S})$; which is a n_s x $(n_s + n_p)$ matrix where $n_s = 2$ where $n_p = 1$. Note that the 2x2 matrix in the homogenous part is unchanged from the variational DE in the prior scenario. The fact that p_1 is now an unknown has no bearing on the partials $\partial\mathbf{a}/\partial\mathbf{r}$ and $\partial\mathbf{a}/\partial\mathbf{v}$. Note the 1x1 element in the lower right-hand corner of the inhomogenous term becomes

$$\frac{\partial \mathbf{a}}{\partial \mathbf{p}} = \left(\frac{\partial \ddot{y}}{\partial p_1}\right) = (-2\, p_1 y)$$

since p_1 is now a force parameter. The full variational DE system is now written as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{\Phi}|\mathbf{S}) = \begin{pmatrix} 0 & 1 \\ -p_1{}^2 & 0 \end{pmatrix}(\mathbf{\Phi}|\mathbf{S}) + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -2\, p_1 y \end{pmatrix}$$

These are the 6 variational DEs integrated in time by DDEABM. As per DC standard practice, and as demonstrated in Section 14, the STM initial conditions are: $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{2x2}$, and the 2x1 sensitivity matrix initial conditions are $\mathbf{S}(t_0) = \mathbf{0}$.

This scenario is designated sho3, and was cycled for 6 loops, 5 iterations. The results are now displayed.

```
program sho3 -- 2022aug28
l.s. solve simple harmonic oscillator by differential correction
unknown force parameter--frequency
imperfect initial statevector
rtol,atol =    1.0000000000000000E-013   1.0000000000000000E-013
ns,np,nq,nsy=           2           1           0           8
nep,nobs =          10          10
xsta,ysta =   -1.0000000000000000       -1.0000000000000000
  perfect p1=   0.59999999999999998
```

```
 imperfect p1=   0.45000000000000001
 xag =   0.40000000000000002       0.20000000000000001       0.59999999999999998
 x0  =   0.30000000000000004       0.15000000000000002       0.45000000000000001
 perfect c1, c2 =   0.33333333333333337       0.40000000000000002
 elb =   (ranges, unitless, perfect)
   1.7204650534085253
   1.8180709213909316
   1.7660231389977032
   1.5881127625767222
   1.3657475192551192
   1.1932547896898711
   1.1152705435822938
   1.1240770355998095
   1.2223466303252157
   1.4115919195461191
 range sigmas =    1.0000000000000000E-003

 r ave/std/rms ->   4.3717959699135632E-002  0.17383963061871688
0.17061493327532026
 r max/iep     ->  0.27195471803218707               5
 r min/iep     -> -0.26903760940637289               9
 x()=   0.12714065556563914       -9.3728933376620693E-002   3.1991367908506430E-002
 finished iteration:           0
 r ave/std/rms ->  -2.5635241491227111E-002   7.5104276036009304E-002
7.5721546849739571E-002
 r max/iep     ->   6.2873174254484132E-002           5
 r min/iep     -> -0.17047153343073229               9
 x()=   -3.1347272652276839E-002  0.15311051921044044       0.16777530402710034
 finished iteration:           1
 r ave/std/rms ->   1.2785993102832016E-002   7.8585854937214400E-002
7.5641553105533749E-002
 r max/iep     ->  0.16820069669824922               9
 r min/iep     ->  -6.7821319673996072E-002          4
 x()=   -4.6426699547637562E-003  -1.1681614258807094E-002  -4.8154588140340182E-002
 finished iteration:           2
 r ave/std/rms ->  -5.2291686300722786E-004   5.4694016292098644E-003
5.2150130209866674E-003
 r max/iep     ->   5.3974240508578575E-003           8
 r min/iep     ->  -8.2481664178473135E-003           1
 x()=    8.8396573156952450E-003   2.2988158325274108E-003  -1.6378023985129976E-003
 finished iteration:           3
 r ave/std/rms ->  -3.6111189779308716E-006   4.4776872463990342E-005
4.2632284212361688E-005
 r max/iep     ->   4.6634701482917151E-005           4
 r min/iep     ->  -9.6061979359296501E-005           9
 x()=    9.6258298569092138E-006   1.2130829789578420E-006   2.5720203845340609E-005
 finished iteration:           4
 r ave/std/rms ->   7.4846417952301178E-011   2.3192921891252745E-009
2.2015464153553630E-009
 r max/iep     ->   3.5876679405077994E-009           9
 r min/iep     ->  -3.1701830049968294E-009           0
 x()=    3.8958491178216498E-009  -4.9051690753770889E-010  -1.6005983048007833E-009
 finished iteration:           5
 xa()=   0.39999999999999991       0.20000000000000215       0.60000000000000064
 err.  =   -1.1102230246251565E-016   2.1371793224034263E-015
 relerr.=   -2.7755575615628914E-016   1.0685896612017132E-014
 p1  =   0.60000000000000064
 err.=    6.6613381477509392E-016

 end of processing
```

As before, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters of this scenario variant that includes a force parameter and a sensitivity matrix. This comparison is shown below in Table 17.1.

Table 17.1 – Adjusted Unknowns, Scenario Parameters, and Error (`sho3`)

Y_0	0.39999999999999991	0.4000000000000000	-1.1102230246251565E-016
$\dot{Y}_0$	0.20000000000000215	0.2000000000000000	2.1371793224034263E-015
p_1	0.60000000000000064	0.6000000000000000	6.6613381477509392E-016

This scenario test again shows the DC process does recover the unknown scenario parameters, which now include a force parameter, p_1, to the limits of 64-bit machine precision.

18. Forced Harmonic Oscillator - - I - - Known Force Parameters

We now create a new computational scenario based on the forced harmonic oscillator (FHO) theory of Section 15. Recall, the inhomogenous DE is 15.1

$$\ddot{y} + {p_1}^2 y = p_2 \cos(p_1 t)$$

where model parameter, p_2, has units of length per time squared.

We define the Forced Harmonic Oscillator Scenario:

$X_s = -1.0$
$Y_s = -1.0$
$Y_0 = 0.4$
$\dot{Y}_0 = 0.2$
$p_1 = 0.6$
$p_2 = 0.1$

Note, p_1 is selected to provide a nearly full oscillation with ten epochs of range data at times 0 through 9. This is to keep t as small as possible for this scenario. Recall in Section 13 that because of resonance with the forcing function on the right, 13.2 will develop a steady-state oscillation with infinite amplitude. Therefore, we analyze the transient response for small values of t.

The exact analytic theory of Section 15 and the defining scenario parameters are sufficient to generate ten epochs of range data at times 0 through 9, inclusive.

Table 18.1 – Perfect Range Data for Forced Harmonic Oscillator Scenario

0	1.7204650534085253
1	1.8575476424068613
2	1.8961014326396548
3	1.7838471962699893
4	1.5280172212056140
5	1.2263266673938802
6	1.0364711194491059
7	1.0000122571820356
8	1.0007537939094795
9	1.0833521633238039

The number of observations, $n = 10$. Observations are uncorrelated. We choose $\sigma = 0.001$. There are two unknown parameters, (Y_0, $\dot{Y}_0$). We identify these unknowns as *state parameters*. Therefore, $u = n_s = 2$. For this FHO scenario, p_1 and p_2 are assumed known. This scenario has no unknown *force parameters*, $n_p = 0$.

As before, we begin with perturbed initial conditions as the starting estimate of the unknown parameters, $\mathbf{X}_0$:

$$Y_0 = 0.3$$
$$\dot{Y}_0 = 0.15$$

Since there is no spatial dependence of the right-hand forcing function in 13.2, the variational DE (which does not include a sensitivity matrix) is identical to that of Section 16. A system of 4 variational DEs are integrated in time by DDEABM.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{\Phi}(t,t_0) = \begin{pmatrix} 0 & 1 \\ -{p_1}^2 & 0 \end{pmatrix}\mathbf{\Phi}(t,t_0)$$

As per DC standard practice, and as demonstrated in Section 14, the STM initial conditions are: $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{2x2}$.

This scenario is designated fho2, and was cycled for 5 loops, 4 iterations. The results are now displayed.

```
 program fho2 -- 2022aug31
 l.s. solve forced harmonic oscillator by differential correction
 imperfect initial statevector
 rtol,atol =    1.0000000000000000E-013   1.0000000000000000E-013
 ns,np,nq,nsy=           2           0           0           6
 nep,nobs =          10          10
 xsta,ysta =   -1.0000000000000000       -1.0000000000000000
   perfect p1=   0.59999999999999998
   perfect p2=   0.10000000000000001
 xag =   0.40000000000000002       0.20000000000000001
 x0  =   0.30000000000000004       0.15000000000000002
 perfect c1, c2 =   0.33333333333333337       0.26111111111111113
 elb =   (ranges, unitless, perfect)
   1.7204650534085253
   1.8575476424068613
   1.8961014326396548
   1.7838471962699893
   1.5280172212056140
   1.2263266673938802
   1.0364711194491059
   1.0000122571820356
   1.0007537939094795
   1.0833521633238039
 range sigmas =    1.0000000000000000E-003

 r ave/std/rms ->  -2.1212624407696092E-002   5.7451176829574797E-002
5.8485462992565504E-002
 r max/iep     ->   5.2477509740658679E-002           5
 r min/iep     -> -0.10781501675050120               1
 x()=    9.9174046262394497E-002   4.9928614169339046E-002
 finished iteration:           0
 r ave/std/rms ->  -6.8145108374317867E-005   3.9080643436813515E-004
3.7696214402049643E-004
 r max/iep     ->   4.6376310607110227E-004           5
 r min/iep     ->  -6.7203919597735862E-004           0
```

```
 x()=    8.2591322302375762E-004   7.1389542425807521E-005
 finished iteration:           1
 r ave/std/rms ->  -6.8976577782819957E-010   1.9370279968576339E-008
1.8389201947969077E-008
 r max/iep     ->   2.5748489118626594E-008           4
 r min/iep     ->  -3.2968070717842579E-008           0
 x()=    4.0514603990313684E-008  -3.7117761389760169E-009
 finished iteration:           2
 r ave/std/rms ->  -1.3544720900426909E-015   1.3649052143972925E-014
1.3019276348119634E-014
 r max/iep     ->   1.8207657603852567E-014           0
 r min/iep     ->  -2.9976021664879227E-014           1
 x()=    6.7028704949717713E-016  -2.6609160224593044E-016
 finished iteration:           3
 r ave/std/rms ->  -1.4432899320127035E-015   1.3523571728434144E-014
1.2910514276842041E-014
 r max/iep     ->   1.8873791418627661E-014           0
 r min/iep     ->  -2.9087843245179101E-014           1
 x()=   -4.8111597004979838E-016  -4.5222411816792907E-017
 finished iteration:           4
 xa()=   0.40000000000002250       0.19999999999998841
 err.   =    2.2482016248659420E-014  -1.1601830607332886E-014
 relerr.=    5.6205040621648550E-014  -5.8009153036664429E-014

 end of processing
```

As before, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters. This comparison is shown below in Table 18.2.

Table 18.2 – Adjusted Unknowns, Scenario Parameters, and Error (`fho2`)

Y_0	0.40000000000002250	0.4000000000000000	2.2482016248659420E-014
$\dot{Y}_0$	0.19999999999998841	0.2000000000000000	-1.1601830607332886E-014

As seen in the other scenarios and variants, the DC process recovers the unknown scenario parameters to the limits of 64-bit machine precision. It seems that the numerical integration of 15.1 has added some slight roundoff error, and an additional iteration might be indicated.

19. Forced Harmonic Oscillator - - II - - Two Unknown Force Parameters

This section explores a variant of the scenario in Section 18. The force parameters, p_1 and p_2, are still considered constant. But now p_1 and p_2 are treated as unknown parameters to be solved. The number of unknowns becomes $u = 4$. The DC unknown *state parameters* are still $(Y_0, \dot{Y}_0)^t$. The number of state parameters remains $n_s = 2$. The number of force parameters is $n_p = 2$.

We retain the FHO equations of motion, and the first-order DE system of Sections 15 and 18. We retain the equations defining range observations, the defining quantities of the Forced Harmonic Oscillator Scenario, and the perfect range data of Table 18.1. We keep the range sigma, dispersion, and weight matrices.

We choose perturbed starting estimates of the unknown parameters, $\mathbf{X}_0$:

$$Y_0 = 0.3$$

$\dot{Y}_0 = 0.15$
$p_1 = 0.45$
$p_2 = 0.075$

Since this variant has a force model parameter, the variational DE is now the combined form holding both the state transition matrix and the sensitivity matrix (Montenbruck and Gill, 2000, Eq. 7.45) reproduced above. We integrate both matrices as the combined matrix, $(\mathbf{\Phi}|\mathbf{S})$; which is a n_s x $(n_s + n_p)$ matrix where $n_s = 2$ where $n_p = 2$.

The full inhomogenous variational DE system is now written as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{\Phi}|\mathbf{S}) = \begin{pmatrix} 0 & 1 \\ -p_1{}^2 & 0 \end{pmatrix}(\mathbf{\Phi}|\mathbf{S}) + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -2\,p_1 y - p_2 t\sin(p_1 t) & \cos(p_1 t) \end{pmatrix}$$

These are the 8 variational DEs integrated in time by DDEABM. As per DC standard practice, and as demonstrated in Section 14, the STM initial conditions are: $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}_{2x2}$, and the 2x2 sensitivity matrix initial conditions are $\mathbf{S}(t_0) = \mathbf{0}$.

This scenario is designated fho3, and was cycled for 9 loops, 8 iterations. The results are now displayed.

```
 program fho3 -- 2022sep02
 l.s. solve forced harmonic oscillator by differential correction
 two unknown force parameters, S(t0)=0
 imperfect initial statevector
 rtol,atol =    1.0000000000000000E-013   1.0000000000000000E-013
 ns,np,nq,nsy=           2           2           0          10
 nep,nobs =          10          10
 xsta,ysta =   -1.0000000000000000       -1.0000000000000000
   perfect p1=   0.59999999999999998
 imperfect p1=   0.45000000000000001
   perfect p2=   0.10000000000000001
 imperfect p2=    7.4999999999999997E-002
 xag =   0.40000000000000002       0.20000000000000001       0.59999999999999998
0.10000000000000001
 x0  =   0.30000000000000004       0.15000000000000002       0.45000000000000001
7.4999999999999997E-002
 perfect c1, c2 =   0.33333333333333337       0.26111111111111113
 elb =   (ranges, unitless, perfect)
   1.7204650534085253
   1.8575476424068613
   1.8961014326396548
   1.7838471962699893
   1.5280172212056140
   1.2263266673938802
   1.0364711194491059
   1.0000122571820356
   1.0007537939094795
   1.0833521633238039
 range sigmas =    1.0000000000000000E-003

 r ave/std/rms ->  0.13750284822375430       0.22371384120141377
0.25288362489537958
 r max/iep     ->  0.49015573940180901             5
 r min/iep     ->  -9.5028363353665268E-002          1
 x()=   0.10056389115876141       9.8796616086885614E-002   5.3098685362056232E-002  -
9.7699915220765821E-002
 finished iteration:            0
 r ave/std/rms ->   6.1961071915903651E-002   6.6900162220754192E-002
8.8697480052224081E-002
```

```
 r max/iep     ->  0.15432677805188422                6
 r min/iep     ->  -2.1911981836467254E-002           3
 x()=   -2.1127238612745902E-002   2.7033710552844661E-002  0.12896206638213684
6.9600219922317394E-002
 finished iteration:            1
 r ave/std/rms ->   1.5020282554647778E-002   9.8019311405277809E-002
9.4194563301444309E-002
 r max/iep     ->  0.25054729238859919                9
 r min/iep     ->  -9.5054849298730648E-002           4
 x()=    1.0874388679331701E-002  -4.1562518398081671E-002  -4.2501566788292144E-002
1.5096688447220119E-002
 finished iteration:            2
 r ave/std/rms ->   9.9734578237975716E-003   1.4405790171881296E-002
1.6918746182613933E-002
 r max/iep     ->   3.3214057082444626E-002           9
 r min/iep     ->  -7.9939504894479541E-003           3
 x()=    8.2044486451957228E-003  -2.6967482377041330E-002   9.5461996901234775E-003
3.0649699511532541E-002
 finished iteration:            3
 r ave/std/rms ->   1.2169462903136364E-003   3.5426325738739228E-003
3.5743781657280524E-003
 r max/iep     ->   9.1688222430355015E-003           9
 r min/iep     ->  -2.9404215759105679E-003           4
 x()=    1.4000052112679955E-003  -7.0209449949782711E-003   8.7942896162676414E-004
7.1044038279066760E-003
 finished iteration:            4
 r ave/std/rms ->   3.9787920223943728E-005   1.2493402206230673E-004
1.2502294780377464E-004
 r max/iep     ->   3.3566469710821778E-004           9
 r min/iep     ->  -9.5220524107375581E-005           4
 x()=    8.4398070038359011E-005  -2.7908922594699538E-004   1.5183344221059636E-005
2.4866284700754079E-004
 finished iteration:            5
 r ave/std/rms ->   3.9029880238850011E-008   1.2940202380418434E-007
1.2881664077135296E-007
 r max/iep     ->   3.5380426854914049E-007           9
 r min/iep     ->  -9.3136452328224095E-008           4
 x()=    1.0684804269102540E-007  -2.9164339510184345E-007   3.0481202430049386E-009
2.4066454429371080E-007
 finished iteration:            6
 r ave/std/rms ->   3.9990233346998137E-014   1.2112014540747375E-013
1.2166470081466599E-013
 r max/iep     ->   3.3106850594322168E-013           9
 r min/iep     ->  -8.7929663550312398E-014           0
 x()=    1.0791087461707088E-013  -2.8610029864956942E-013   7.0512133410446710E-015
2.3669046790464206E-013
 finished iteration:            7
 r ave/std/rms ->  -4.4408920985006264E-017   8.4260003245840822E-016
8.0059320849734419E-016
 r max/iep     ->   1.5543122344752192E-015           2
 r min/iep     ->  -1.1102230246251565E-015           4
 x()=    1.8109478659338511E-016  -4.2409858914604050E-016  -6.1306794338266376E-016  -
4.8278036294486476E-016
 finished iteration:            8
 xa()=   0.40000000000000008       0.20000000000000040       0.59999999999999887
9.9999999999998937E-002
 err.   =    5.5511151231257827E-017   3.8857805861880479E-016
 relerr.=    1.3877787807814457E-016   1.9428902930940239E-015
 p1  =   0.59999999999999887
 err.=   -1.1102230246251565E-015
 p2  =    9.9999999999998937E-002
 err.=   -1.0685896612017132E-015

 end of processing
```

As before, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters. This comparison is shown below in Table 19.1.

Table 19.1 – Adjusted Unknowns, Scenario Parameters, and Error (`fho3`)

Y_0	0.40000000000000008	0.4000000000000000	5.5511151231257827E-017
$\dot{Y}_0$	0.20000000000000040	0.2000000000000000	3.8857805861880479E-016
P_1	0.59999999999999887	0.6000000000000000	-1.1102230246251565E-015
P_2	0.09999999999999893 7	0.1000000000000000	-1.0685896612017132E-015

This computation numerically integrated a proposed counter-example DE (12.2). The initial value of the sensitivity matrix was set to $\mathbf{S}(t_0) = \mathbf{0}$. Further, the initial value of the exact analytic solution of the sensitivity matrix derived in Section 15 confirmed $\mathbf{S}(t_0) = \mathbf{0}$. The DC process recovers the unknown scenario parameters, which now includes two force parameters, p_1 and p_2, to the limits of 64-bit machine precision.

20. Uniform Gravity Model - - III - - Perturbed Initial Condition

The claims that the standard application of variational equations in orbit determination (OD), as well as in other applications, has neither a sound mathematical nor physical basis rests upon an argument regarding the initial value of the sensitivity matrix. The results from sections 8, 10, 17, and 19 illustrate that, following standard DC practice (Montenbruck and Gill, 2000, pg. 241), and as proved by Coddington and Levinson (1955, Chapter 1), the initial conditions, $\mathbf{S}(t_0) = \mathbf{0}$, provide convergence to a solution with an accuracy that is only limited by machine precision.

An irresistible question naturally arises, "What happens if $\mathbf{S}(t_0) \neq \mathbf{0}$?".

Of course, by the general proof by Coddington and Levinson (1955, Chapter 1) we can expect a degraded result. And, by the derivations of the specific solutions to the IVP scenarios in Sections 14 and 15, we have analytic expressions that $\mathbf{S}(t_0) = \mathbf{0}$. And, our computational results match the theory. We expect trouble if $\mathbf{S}(t_0) \neq \mathbf{0}$.

We revisit the Uniform Gravity Scenario, which is simply a 2-D trajectory of an object in a uniform gravity field, with associated measurement types from a fixed tracking station. This section's variant is a modification of the variant of Section 8 (`toyorb4`). The magnitude of uniform gravity, g, is constant, and is treated as an unknown parameter to be solved. As before, g, is a *force parameter*, and $n_p = 1$. It is solved by a DC procedure with the *state parameters*, $(X_0, Y_0, \dot{X}_0, \dot{Y}_0)^t$, $n_s = 4$, and the number of unknowns, $u = 5$.

The complaint about OD practice does not offer any insight regarding alternative contents of $\mathbf{S}(t_0)$. For this experiment, $\mathbf{S}(t_0) = (0, 0, 0, 1)^t$, is adopted. This is the only change made from the toyorb4 program of Section 8.

This scenario is designated toyorb4x, and was cycled for 5 loops, 4 iterations. The results are now displayed.

```
program toyorb4x -- 2022aug03
l.s. solve toy orbit+g by differential correction
unknown g, ddeabm() variational equations
TEST INVALID INITIAL CONDITION FOR SENSITIVITY MATRIX
```

```
   perfect g=   0.50000000000000000
 imperfect g=   0.29999999999999999

 xsta,ysta=    1.0000000000000000        1.0000000000000000
   perfect xsg=    1.0000000000000000        8.0000000000000000        2.0000000000000000
1.0000000000000000
 imperfect xsi=    1.5000000000000000        10.000000000000000        2.2000000000000002
0.50000000000000000
 elb=   (ranges, unitless, perfect)
   7.0000000000000000
   8.0039052967910607
   8.9442719099991592
   9.8011478919563295
   10.630145812734650
   11.535271995059327
   12.649110640673518
   14.108951059522463
   16.031219541881399
   18.494931738181680
 range sigmas =    9.9999999999999995E-007

 in lsq2loop--SET INVALID INIT.COND. FOR SENS. MATRIX
 S()=    0.0000000000000000        0.0000000000000000        0.0000000000000000
1.0000000000000000
 r ave/std/rms ->   2.0945109394094805       0.35124721684114563
2.1208520037736651
 r max/iep     ->   2.6042921924618749                 7
 r min/iep     ->   1.6561997776192747                 2
 x()=   -1.2681644065633009       -1.9473834175396689       -8.4221430348065951E-003
0.37532536546609663       0.28751686699538936
 finished iteration:            0
 in lsq2loop--SET INVALID INIT.COND. FOR SENS. MATRIX
 S()=    0.0000000000000000        0.0000000000000000        0.0000000000000000
1.0000000000000000
 r ave/std/rms ->  0.15878707876686252        1.0268135202206254
0.98697757885394710
 r max/iep     ->   2.4311244406513062                 9
 r min/iep     -> -0.71868111502073084                 4
 x()=    1.3781694255209231        5.3730107878998012E-002 -0.24334156219401848
3.8755062195164669E-002 -0.11421477603713015
 finished iteration:            1
 in lsq2loop--SET INVALID INIT.COND. FOR SENS. MATRIX
 S()=    0.0000000000000000        0.0000000000000000        0.0000000000000000
1.0000000000000000
 r ave/std/rms ->  0.22158602291513452        8.4060086741375523E-002
0.23549915905586019
 r max/iep     ->  0.31082894324313060                 5
 r min/iep     ->   5.0197907003507680E-002            9
 x()=  -0.65377667069905954       -7.6916837310524677E-002   5.7629162099857467E-002
5.9942471480511017E-002   2.7292065858307346E-002
 finished iteration:            2
 in lsq2loop--SET INVALID INIT.COND. FOR SENS. MATRIX
 S()=    0.0000000000000000        0.0000000000000000        0.0000000000000000
1.0000000000000000
 r ave/std/rms ->  -1.9614677427194936E-002   4.4336477411890327E-002
4.6409982513878040E-002
 r max/iep     ->   6.2574072320632723E-002            9
 r min/iep     ->  -6.7357936863762902E-002            4
 x()=    4.7039957597242221E-002  -2.9277227730294786E-002  -6.1483104669726174E-003
2.5829341456137023E-002  -7.2192314130847812E-004
 finished iteration:            3
 in lsq2loop--SET INVALID INIT.COND. FOR SENS. MATRIX
 S()=    0.0000000000000000        0.0000000000000000        0.0000000000000000
1.0000000000000000
 r ave/std/rms ->   1.2227589617052280E-003   8.0714932218643146E-004
1.4427336912631733E-003
 r max/iep     ->   2.1111853419562010E-003            5
 r min/iep     ->  -2.1482581827214631E-004            9
 x()=   -3.2681376489300251E-003  -1.5186593982123402E-004   2.8286288016345085E-004
1.9740554971101054E-005   1.2774682627726512E-004
 finished iteration:            4
```

```
 xa()=    1.0000001682068749        8.0000007593586879        2.0000000092842236
0.99987198115288045       0.49999998050153538
 err.=    1.6820687487317798E-007   7.5935868792953443E-007   9.2842236121271071E-009  -
1.2801884711954870E-004
 grav=   0.49999998050153538
 err.=   -1.9498464620681943E-008

 end of processing
```

As before, we test to see how well the iterated, adjusted unknowns, $\mathbf{X}_a$ (denoted in the computer output as "`xa()`"), match the unperturbed defining parameters. This scenario variant includes a force parameter, a sensitivity matrix, and a variant set of sensitivity matrix initial conditions that contains one entry that is incorrect. This comparison is shown below in Table 20.1.

Table 20.1 – Adjusted Unknowns, Scenario Parameters, and Error (`toyorb4x`)

X_0	1.0000001682068749	1.0000000000000000	1.6820687487317798E-007
Y_0	8.0000007593586879	8.0000000000000000	7.5935868792953443E-007
$\dot{X}_0$	2.0000000092842236	2.0000000000000000	9.2842236121271071E-009
$\dot{Y}_0$	0.99987198115288045	1.0000000000000000	1.2801884711954870E-004
g	0.49999998050153538	0.50000000000000000	-1.9498464620681943E-008

These results should also be compared to Table 8.1, the variant scenario, toyorb4, with a correct initial condition for the sensitivity matrix.

This variant scenario test shows the experimental DC process does *not* completely recover the unknown scenario parameters when compared to Section 8. To be sure, 6 to 8 digits are obtained (with one important exception). But these results fall far short of the exquisite accuracy of toyorb4. Note that the worst result belongs to $\dot{Y}_0$, which has only half the correct digits as the other unknowns. Recall it was the $\frac{\partial \dot{Y}(t_0)}{\partial g}$ entry of the $\mathbf{S}(t_0)$ matrix that was set to non-zero in this experiment.

Convergence is retarded when compared to Section 8. Perhaps the scenario parameters could be recovered in time with more iterations. The experimental perturbation applied to $\mathbf{S}(t_0)$ creates perturbed values of the design matrix, $\mathbf{A}$; and degraded partials mean degraded convergence.

This numerical experiment was a failure in recovery of the scenario parameters. But this deliberate failure reinforces a key point. The initial values of the sensitivity matrix, $\mathbf{S}(t_0) = \mathbf{0}$, are mandatory for parametric IVPs; which are common in differential correction (DC).

21. Discussion

This author must again relate his enthusiasm about Section 1.2 of Tapley, et.al (2004), which describes their uniform gravity field model (UGFM), and which is found in Section 7 of this study. Their 2-D ballistic model with a companion set of perfect range data communicates the fundamental essence of the orbit determination (OD) problem. It has an exact analytic solution to its (idealized) theory with no question regarding its valid mathematical basis. Any numerical error in implementation becomes immediately evident. Yet, it is easily extended to overdetermined, least squares problems, and can include force parameter estimation. This study's eight exercises (and one exploratory experiment in Section 20), can be considered an educational "boot camp" that ensures a reader can take simple physical systems, and establish an initial value problem (IVP) for the physical model, and an associated variational IVP for the observation model partial derivatives.

A pleasant result in the scenario tests was the excellent performance of the DDEABM integrator. In Section 10, initial positions were recovered to 0.5 to 1.4 micrometers of accuracy over a 24-hour span of 3¾ orbital revolutions. Additional tests could look at longer time spans, lower orbits, faster data rates, and variation of the accuracy limits communicated to the software package. In addition, one can look at alternative DE integration software, perhaps with higher order algorithms.

Some mention must be made of an entire an entire body of expertise in orbit determination (OD) that is not batch least squares differential correction (DC). I speak of Kalman filters (KF), and the iterative form, extended Kalman filters (EKF). These are described in numerous references, such as Gelb (1974) and Brown and Hwang (2012). Also, some texts on OD address KF's, such as Tapley et al. (2004), Vallado (2013), and Montenbruck and Gill (2000). Having a separate methodology for orbit determination (OD) provides a powerful cross check on underlying theory.

This author must now advertise for iteration in orbit determination (OD). The measurements and dynamical motion are nonlinear. Yes, one may have very good initial conditions, and low accuracy requirements. But software systems will be updated due to changing requirements and new measurement technologies. And, there is always the possibility of bugs, both new and legacy. When a new software system is proposed, consider including iteration of the nonlinear problem. It doesn't add much to the design and coding effort. Such software does not need to be iterated on a routine basis. But it certainly can help detect bugs in force models and/or variational equations. And, maybe, one's state vector initial conditions were not as good as originally thought.

This study is somewhat long. However, this author felt there was a gap in the literature. Variational equations are more abstract than force models, and should have a reference that provides detailed, descriptive examples of their application in differential correction (DC). This author wrote the study he wishes was available when he first encountered variational equations.

22. Conclusions

This author has located the general mathematical proofs for the initial value problem (IVP) of a system of parametric ordinary differential equations (ODE) in Coddington and Levinson (1955, Chapter 1, Section 7). Further, their work includes a derivation of the associated IVP for the variational ODE and the companion initial conditions for the columns of the sensitivity matrix, $\mathbf{S}(t)$. The general character of their mathematical work supports all applications involving differential equation (DE) models with parameters. I conclude that, in contrast to Xu (2018), the standard application of variational equations in orbit determination (OD) has a sound basis, and that $\mathbf{S}(t_0) = \mathbf{0}$.

I have derived the exact analytic solutions to the IVPs of both the simple and the forced harmonic oscillator models. This, in turn, allowed derivation of the associated exact analytic solutions for the state transition (STM) and sensitivity matrices for both oscillator variational ODEs. Evaluation of those solutions at $t = 0$ confirm the initial values of the STM and sensitivity matrix used in standard OD practice. This result also invalidates a proposed counterexample using the forced harmonic oscillator model (Xu, 2018) as an argument against standard OD practice.

This author finds that the standard application of variational equations in OD computation, including the initial condition of the sensitivity matrix, $\mathbf{S}(t_0) = \mathbf{0}$, does recover scenario parameters to the limits of machine precision. I also find that experimental modification of the initial conditions of the sensitivity matrix to $\mathbf{S}(t_0) \neq \mathbf{0}$, had a decidedly detrimental effect on convergence to correct values. This reinforces the validity of standard OD practice.

It is found that nonlinear least squares, when iterating the unknown parameters, is successful in achieving highly accurate results. It was found that a sophisticated differential equation integrator (DDEABM) could process the classic, central force problem (Kepler) of 3¾ orbital revolutions, almost to the limits of 64-bit machine precision. Observation misclosure statistics based on error-free synthetic data provide evidence that extremely tiny amounts of force model integration error were passed to the misclosures through the computed observations.

I conclude the methodology of processing perfect, synthetic data derived from exact analytic models is an extremely powerful tool in debugging OD software.

ACKNOWLEDGEMENTS

Particular thanks are given to Prof. Chris Jekeli, who provided a reference copy of the DVDQ differential equation integrator by F. T. Krogh. Although DVDQ was ultimately not used in this study, it helped me past a hurdle to achieve a goal of a relatively simple differential correction software. I also greatly appreciate his conversations, emails, and perspectives on this entire problem.

I thank the reviewers, Dr. Rick Bennett and Dr. Phillip McFarland, for their very helpful comments and suggestions that improved the quality of this study.

A helpful comment by Prof. Artem Novozhilov crystalized the Coddington and Levinson proof regarding the initial condition of the sensitivity matrix.

Thank you to L. F. Shampine and H. A. Watts for writing the DDEABM differential equation integration software package. Thanks to the organizers of the SLATEC mathematical library and to NETLIB for hosting the Fortran 77 version of the DEPAC collection. I thank Bill Schelter and the cadre of workers who built, extend, and maintain the Maxima computer algebra system (and its Macsyma forerunner) and made it available on SourceForge. I thank Prof. Urho A. Uotila, who insured that all students in his least squares adjustment courses had a validated collection of matrix manipulation routines. A version of that software was used in this study.

REFERENCES

Beutler, Gerhard, 2005: *Methods of Celestial Mechanics, Volume I: Physical, Mathematical, and Numerical Principles*. Springer-Verlag, Berlin, 482 pp.

Blanchard, Paul, Robert L. Devaney, and Glen R. Hall, 2012: *Differential Equations, Fourth Ed*. Brooks/Cole, Boston, MA, 834 pp.

Brown, Robert Grover and Patrick Y. C. Hwang, 2012: *Introduction to Random Signals and Applied Kalman Filtering, Fourth Ed*. John Wiley & Sons, Inc., Hoboken, NJ, 383 pp.

Cappellari, J.O., Velez, C.E., and Fuchs, A.J., 1976: *Mathematical Theory of The Goddard Trajectory Determination System*, NASA-TM-X-71106, Goddard Space Flight Center, Greenbelt, Maryland, 598 pp.

Coddington, Earl A. and Norman Levinson, 1955: *Theory of Ordinary Differential Equations*. McGraw-Hill Inc., New York, 429 pp.

Feynman, Richard P., Robert B. Leighton, and Matthew Sands, 1963: *The Feynman Lectures on Physics, Vol. I*. Basic Books, New York, 968 pp.

Fowles, Grant R., and George L. Cassiday, 2005: *Analytical Mechanics, Seventh Ed*. Thomson Brooks/Cole, Belmont, CA, 574 pp.

Gelb, Arthur, ed., 1974: *Applied Optimal Estimation*. The MIT Press, Cambridge, MA, 374 pp.

Gerald, Curtis. F. and Patrick. O. Wheatley, 1989: *Applied Numerical Analysis, Fourth Ed*. Addison-Wesley Publishing Co., Reading, MA, 722 pp.

Ghilani, Charles D. and Paul R. Wolf, 2006: *Adjustment Computations, Fourth Ed*. John Wiley & Sons, Inc., Hoboken, NJ, 611 pp.

Habana, Nlingilili, 2020: *Gravity Recovery by Kinematic State Vector Perturbation from Satellite-to-Satellite Tracking for GRACE-like Orbits over Long Arcs*, Report No. 514, Geodetic Science, The Ohio State University, Columbus, Ohio, 127 pp.

Jäggi, A., Arnold, D. (2017): Precise Orbit Determination. In: Naeimi, M., Flury, J. (eds.), *Global Gravity Field Modeling From Satellite-to-Satellite Tracking Data*, pp.35-80. Lecture Notes in Earth System Sciences, Springer International Publishing AG, Cham, Switzerland, 2017.

Jekeli, Christopher and Nlingilili Habana, 2019: *On the Numerical Implementation of a Perturbation Method for Satellite Gravity Mapping*. In: IX Hotine-Marussi Symposium on Mathematical Geodesy, International Association of Geodesy Symposia 151, Springer Nature Switzerland AG, Switzerland, 175-180.

Long, A.C., Cappellari, J.O., Velez, C.E., and Fuchs, A.J., 1989: *Goddard Trajectory Determination System (GTDS) Mathematical Theory Revision 1*, Technical Report FDD/552-89/001 and CSC/TR-89/6001, Goddard Space Flight Center, Greenbelt, Maryland, 698 pp.

Maxima.sourceforge.io. *Maxima, a Computer Algebra System*. Version 5.46.0 (2022). https://maxima.sourceforge.io/

Marion, Jerry B., 1965: *Classical Dynamics of Particles and Systems*. Academic Press, New York, 576 pp.

Mikhail, E. M., 1976: *Observations and Least Squares*. IEP -- Dun-Donnelley, New York, 497 pp.

Milbert, Dennis and Christopher Jekeli, 2023: On the initialization of the sensitivity matrix in variational equations. *J Geod* 97, 88 (2023). https://doi.org/10.1007/s00190-023-01776-4.

Montenbruck, Oliver and Eberhard Gill, 2000: *Satellite Orbits: Models, Methods, Applications*. Springer-Verlag, Berlin, 369 pp.

Riley, James. D., Bennett, Morris M., and McCormick, Emily. 1967: Numerical integration of variational equations. *Math. Comput*., 21(1), 12-17.

Shampine, Lawrence. F. and M. K. Gordon, 1975: *Computer Solution of Ordinary Differential Equations*. W. H. Freeman and Co., San Francisco, 318 pp.

Shampine, Lawrence F. and Herman A. Watts, 1980: *DEPAC – Design of a User Oriented Package of ODE Solvers*. Report SAND79-2374. Sandia National Laboratories, Albuquerque, NM, 88 pp.

Tapley, Byron D., Bob E. Schutz, and George H. Born, 2004: *Statistical Orbit Determination*. Elsevier Academic Press, Burlington, MA, 547 pp.

Vallado, David A. 2013: *Fundamentals of Astrodynamics and Applications, Fourth Ed*. Microcosm Press, Hawthorne, CA, 1106 pp.

Xu, P. (2009): Zero initial partial derivatives of satellite orbits with respect to force parameters violate the physics of motion of celestial bodies. *Sci China Ser D*, 52:562–566.

Xu, P. (2015): Zero initial partial derivatives of satellite orbits with respect to force parameters nullify the mathematical basis of the numerical integration method for the determination of standard gravity models from space geodetic measurements. Geophysical Research Abstracts, Vol. 17, EGU2015-15243, 2015 EGU General Assembly.

Xu, P. (2018): Measurement-based perturbation theory and differential equation parameter estimation with applications to satellite gravimetry. *Commun Nonlinear Sci Numer Simulat*, 59:515-543.

Xu, P. (2021): Reconstruction of mathematical foundations for satellite gravimetry from tracking: solutions to problems incorrectly solved for 100 years. IAG 2021 Abstract Book, Scientific Assembly of the International Association of Geodesy, Beijing, June 28-July 2, 2021.