B.O.P.P.A. Software Guide for Orbit Determination

Miles Puchner
Department of Mechanical and Aerospace Engineering, UC San Diego
MAE 182: Spacecraft Guidance & Navigation
Professor Rosengren
21 March 2024

I. Abstract

The need for a high-precision solution to the orbit-determination (OD) problem is paramount within the spacecraft guidance and navigation field. As such, the purpose of the following guide is to develop the necessary theory and derivations behind the problem and then exemplify the process of obtaining a high-accuracy solution via the Batch Orbit Prediction and Precision Algorithm, or BOPPA. To do this, the BOPPA software will be used to estimate the state of a LEO satellite given instantaneous range and range rate data from 3 different tracking stations over roughly 5 hours. The governing equations used to model the spacecraft's dynamics will include the two-body equation of motion, the effects of Earth's oblateness, and atmospheric drag; all of which are consistent with the primary perturbing forces felt by LEO spacecraft. The BOPPA software will do this by way of statistical OD in which the Batch Processor Algorithm is employed. The high accuracy of this software is accomplished through careful consideration of parameter uncertainties, ease of user ability to modify aspects of both the dynamics and iterative Batch procedure, and integration of advanced numerical methods and computational techniques. Overall, it is shown that given a real-world OD problem and necessary data, a user can effortlessly utilize the BOPPA software to find a precise solution.

II. Introduction

As common orbits within the immediate region above the Earth's atmosphere become overwhelmed by defunct satellites and debris, catastrophic collisions with operational spacecraft become inevitable. As a result, modern space operations are in an ever-increasing need to swiftly and accurately determine the orbits of any Earth-orbiting object so that maneuvers can be made to avoid such collisions. The ability to do this with accuracy is no easy task as countless sources of error riddle the OD problem from start to finish. Such uncertainties include the inherent errors in observation instruments, numerical integration, and uncertainties in specific spacecraft parameters or even the governing equations of motion themselves.

To solve this problem, two schools of thought have been developed, deterministic and stochastic OD. While often quick and computationally inexpensive, the deterministic approach lacks accuracy as it uses the minimum amount of required information to create a best estimate. As a result, errors within the observations that the algorithm uses are pervasive throughout the estimation. On the other hand, Stochastic OD uses an abundance of information along with corresponding statistical techniques to determine an estimated trajectory and refine it. The BOPPA

software is one such stochastic model that employs a minimum-variance criterion with *a priori* information to determine a best estimate of the actual trajectory. Such a criterion allows for the use of the statistical profiles of the measurement and *a priori* errors to aid in the accuracy of the estimation. Similarly, by weighting the observations, the errors can be appropriately scaled to match the accuracy of the site, instrument, or differing environmental conditions from which the observation was taken.

To demonstrate this, an in-depth overview of the OD problem will be provided along with a solution using the BOPPA software. By the end of this guide, the user will come to understand the nuances of the OD problem and see just how valuable the BOPPA software can be for any agency, company, or researcher involved in spacecraft guidance and navigation.

III. Problem Formulation

The OD problem is defined by the site observations and the orbits of the spacecraft/body in question which are governed by the dynamics. The dynamical models used in BOPPA can be broken down into three main categories: the two-body equation of motion, the effects of Earth's oblateness, and atmospheric drag. It is important to note that while this is a simplification of the actual dynamics felt by any planetary orbiting body, the Batch algorithm is primarily used for LEO satellites whose observation data can be reliably and systematically obtained in batches. This simplification is both necessary and sufficient in describing the dynamics as the above three accelerations are the primary perturbing forces that govern LEO; see Fig. 1.

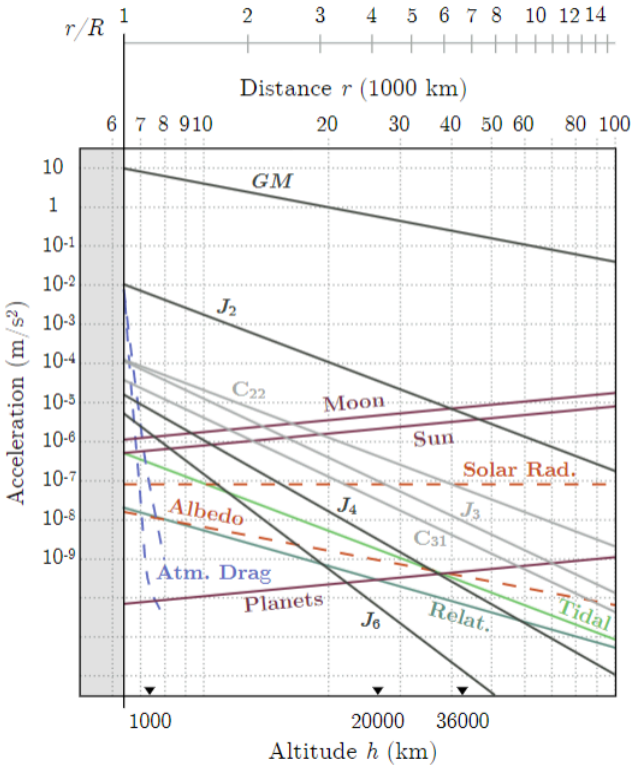

Figure 1[1]: Perturbing accelerations felt by Earth-orbiting spacecraft as a function of altitude above the Earth's surface (LEO: h < 1000 km)

A. Orbit Dynamics Formulation

The above 3 dynamical models can be expressed via the following equations. The potential function for Two-Body motion with included effects from Earth's oblateness (J_2) is

$$U' = \frac{\mu}{r} - \frac{3\mu J_2 R_e^2}{2}\frac{z^2}{r^5} + \frac{\mu J_2 R_e^2}{2}\frac{1}{r^3} \tag{1}$$

Where the gradient of the potential gives the components of the absolute acceleration, that is

$$\frac{\partial U'}{\partial x} = -\frac{\mu x}{r^3}[1 - \frac{3}{2}J_2(\frac{R_e}{r})^2(5(\frac{z}{r})^2 - 1)] \tag{2.a}$$

$$\frac{\partial U'}{\partial y} = -\frac{\mu y}{r^3}[1 - \frac{3}{2}J_2(\frac{R_e}{r})^2(5(\frac{z}{r})^2 - 1)] \tag{2.b}$$

$$\frac{\partial U'}{\partial z} = -\frac{\mu z}{r^3}[1 - \frac{3}{2}J_2(\frac{R_e}{r})^2(5(\frac{z}{r})^2 - 3)] \tag{2.c}$$

The atmospheric drag can be modeled via

$$\ddot{\boldsymbol{r}}_{drag} = -\frac{1}{2}C_D(\frac{A}{m})\rho_A v_A \boldsymbol{v}_A \tag{3}$$

where

$$\boldsymbol{v}_A = \begin{bmatrix}\dot{x} + \dot{\theta}y & \dot{y} - \dot{\theta}x & \dot{z}\end{bmatrix}^T \tag{3.a}$$

$$v_A = \sqrt{(\dot{x} + \dot{\theta}y)^2 + (\dot{y} - \dot{\theta}x)^2 + (\dot{z})^2} \tag{3.b}$$

$$\rho_A = \rho_0 exp(\frac{-(r - r_0)}{H}) \tag{3.c}$$

The total governing dynamics in component form are therefore just the sum of the potential gradients and the atmospheric drag acceleration components

$$\ddot{x} = -\frac{\mu x}{r^3}[1 - \frac{3}{2}J_2(\frac{R_e}{r})^2(5(\frac{z}{r})^2 - 1)] - \frac{1}{2}C_D(\frac{A}{m})\rho_A v_A(\dot{x} + \dot{\theta}y) \tag{4.a}$$

$$\ddot{y} = -\frac{\mu y}{r^3}[1 - \frac{3}{2}J_2(\frac{R_e}{r})^2(5(\frac{z}{r})^2 - 1)] - \frac{1}{2}C_D(\frac{A}{m})\rho_A v_A(\dot{y} - \dot{\theta}x) \tag{4.b}$$

$$\ddot{z} = -\frac{\mu z}{r^3}[1 - \frac{3}{2}J_2(\frac{R_e}{r})^2(5(\frac{z}{r})^2 - 3)] - \frac{1}{2}C_D(\frac{A}{m})\rho_A v_A(\dot{z}) \tag{4.c}$$

where x, y, and z are the spacecraft's position components in the ECI frame, $\dot{x}$, $\dot{y}$, and $\dot{z}$ are the spacecraft's velocity components in the ECI frame, r is the norm of the position vector, μ is the gravitational parameter, J_2 is the Earth-oblateness constant, R_e is the Earth's radius, C_D is the satellite's drag coefficient, A is the satellite's cross-sectional area, m is the satellite's mass, r_0 is the reference position norm, ρ_0 is the reference atmospheric density, H is the reference altitude, and $\dot{\theta}$ is the Earth's rotation rate.

B. Observation-State Relationship Formulation

To accurately describe the observations from the integrated dynamics, the range and range rate are defined respectively as

$$\rho = (x^2+y^2+z^2+x_{si}^2+y_{si}^2+z_{si}^2-2(xx_{si}+yy_{si})cos(\theta)+2(xy_{si}-yx_{si})sin(\theta)-2zz_{si})^{\frac{1}{2}} \quad (5.a)$$

$$\dot{\rho} = \frac{1}{\rho}(x\dot{x}+y\dot{y}+z\dot{z}-(\dot{x}x_{si}+\dot{y}y_{si})cos(\theta)+\dot{\theta}(xx_{si}+yy_{si})sin(\theta)+(\dot{x}y_{si}-\dot{y}x_{si})sin(\theta)+\dot{\theta}(xy_{si}-yx_{si})cos(\theta)-\dot{z}z_{si}) \quad (5.b)$$

with

$$\theta = \dot{\theta}t \quad (5.c)$$

where the new variables x_{si}, y_{si}, and z_{si} are the observation site position components in the ECEF frame and t is the difference in time between the current and initial observation.

C. Orbit-Determination Problem Formulation

For the general OD problem, the nonlinear expressions

$$\dot{\boldsymbol{X}} = F(\boldsymbol{X}, t), \qquad \boldsymbol{X}(t_k) \equiv \boldsymbol{X}_k \quad (6.a)$$

$$\boldsymbol{Y}_i = G(\boldsymbol{X}_i, t_i) + \boldsymbol{\epsilon}_i, \qquad i = 1, ..., l \quad (6.b)$$

represent the state-space form of the dynamics and the observations respectively. Specifically, G is a matrix that holds equations 5.a and 5.b. The value $\boldsymbol{X}_k$ is the unknown n-dimensional state vector at time t_k and $\boldsymbol{Y}_i$ is a p-dimensional observation vector that will be used to obtain $\hat{\boldsymbol{X}}$, a best estimate of the unknown state vector. The OD problem can therefore be framed in the following manner; the amount of unknowns created by the unknown state $\boldsymbol{X}_k$ and the unknown errors in the observations $\boldsymbol{\epsilon}_i$, is greater than that of the known observations for any given observation index. In other words, p knowns is less than n + p unknowns for any i, and thus the system is underdetermined. However, should the state and observation relationships be linearized, certain techniques from linear estimation theory can be employed to create a solvable system.

To do this, the nonlinearities within the dynamics and observations are voided via a truncated Taylor series expansion about an *a priori* reference trajectory $\boldsymbol{X}^*$. Of course, this series truncation to the linear term approximates the actual dynamics and therefore assumes that the reference trajectory remains sufficiently close to the true trajectory. It should be noted that serious error will occur should the reference fail to remain within this linear envelope throughout the entire time interval.

Expansion of equation set 6 within a Taylor series about $\boldsymbol{X}^*$ yields

$$\dot{\boldsymbol{X}} = F(\boldsymbol{X}, t) = F(\boldsymbol{X}^*, t) + \left[\frac{\partial F(t)}{\partial \boldsymbol{X}(t)}\right]^* (\boldsymbol{X}(t) - \boldsymbol{X}^*(t)) + h.o.t. \quad (7.a)$$

$$\dot{\boldsymbol{Y}}_i = G(\boldsymbol{X}_i, t) + \boldsymbol{\epsilon}_i = G(\boldsymbol{X}_i^*, t_i) + \left[\frac{\partial G}{\partial \boldsymbol{X}}\right]^* (\boldsymbol{X}(t) - \boldsymbol{X}^*(t)) + \boldsymbol{\epsilon}_i + h.o.t. \quad (7.b)$$

where $[\,]^*$ denotes an evaluation at the reference and $\boldsymbol{X}^*$ is determined through integration of equation 6.a. Furthermore, let

$$\boldsymbol{x}(t) = \boldsymbol{X}(t) - \boldsymbol{X}^*(t) \qquad \& \qquad \boldsymbol{y}_i(t) = \boldsymbol{Y}(t) - \boldsymbol{Y}^*(t) \tag{8}$$

denote an nx1 state deviation vector and a px1 observation deviation vector respectively. If higher order terms are eliminated, equation set 7 can then be written in terms of equation 8 as

$$\dot{\boldsymbol{x}} = A(t)\boldsymbol{x}(t) \qquad where \qquad A(t) = \left[\frac{\partial F(t)}{\partial \boldsymbol{X}(t)}\right]^* \tag{9.a}$$

$$\boldsymbol{y}_i(t) = \tilde{H}_i \boldsymbol{x}_i + \boldsymbol{\epsilon}_i \qquad where \qquad \tilde{H}_i = \left[\frac{\partial G}{\partial \boldsymbol{X}}\right]^* \tag{9.b}$$

Finally, let the state transition matrix Φ be an nxn matrix that maps the state deviation vector from an arbitrary time t_k to time t; that is

$$\boldsymbol{x}(t) = \Phi(t, t_k)\boldsymbol{x}_k \tag{10}$$

A full solution to the state transition matrix is outlined in Tapley *et al.*[2] (2004), however, for the purposes of this guide, a brief summary is sufficient. Below is the STM differential equation.

$$\dot{\Phi}(t, t_k) = A(t)\Phi(t, t_k) \qquad with \;\; I.C. \qquad \Phi(t_k, t_k) = I \tag{11}$$

It can be shown that equation 11 has independent columns such that integration of each column with respect to the dynamics will result in a solution to the STM at the corresponding epoch. Equation 9.b can now be written in terms of an arbitrary initial state deviation by using equation 10

$$\boldsymbol{y}_i(t) = H_i \boldsymbol{x}_k + \boldsymbol{\epsilon}_i \qquad where \qquad H_i = \tilde{H}_i \Phi(t_i, t_k) \qquad (i = 1, ..., l) \tag{12}$$

Now that the OD problem has been reduced to a linear estimation problem, the specific case for a LEO satellite whose range and range rate can be tracked by up to 3 stations can be formulated. The state vector of the specific case follows as

$$\boldsymbol{X} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \mu & J_2 & C_D & x_{s1} & y_{s1} & z_{s1} & x_{s2} & y_{s2} & z_{s2} & x_{s3} & y_{s3} & z_{s3} \end{bmatrix}^T \tag{13}$$

where the variables remain the same as those defined in sections A and B. Taking the time derivative of equation 13 gives us the state-space form as

$$\dot{\boldsymbol{X}} = F(\boldsymbol{X}, t) = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \ddot{x} & \ddot{y} & \ddot{z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{14}$$

where equation set 4 can be inputted for $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ respectively. The remaining terms in equation 13 are reduced to zeros as the site coordinates and dynamic parameters are constants. The A(t) matrix can then be determined by taking the derivative of equation 14 with respect to equation 13.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial y} & \frac{\partial \ddot{x}}{\partial z} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial \dot{y}} & \frac{\partial \ddot{x}}{\partial \dot{z}} & \frac{\partial \ddot{x}}{\partial \mu} & \frac{\partial \ddot{x}}{\partial J_2} & \frac{\partial \ddot{x}}{\partial C_D} \\ \frac{\partial \ddot{y}}{\partial x} & \frac{\partial \ddot{y}}{\partial y} & \frac{\partial \ddot{y}}{\partial z} & \frac{\partial \ddot{y}}{\partial \dot{x}} & \frac{\partial \ddot{y}}{\partial \dot{y}} & \frac{\partial \ddot{y}}{\partial \dot{z}} & \frac{\partial \ddot{y}}{\partial \mu} & \frac{\partial \ddot{y}}{\partial J_2} & \frac{\partial \ddot{y}}{\partial C_D} \\ \frac{\partial \ddot{z}}{\partial x} & \frac{\partial \ddot{z}}{\partial y} & \frac{\partial \ddot{z}}{\partial z} & \frac{\partial \ddot{z}}{\partial \dot{x}} & \frac{\partial \ddot{z}}{\partial \dot{y}} & \frac{\partial \ddot{z}}{\partial \dot{z}} & \frac{\partial \ddot{z}}{\partial \mu} & \frac{\partial \ddot{z}}{\partial J_2} & \frac{\partial \ddot{z}}{\partial C_D} \end{bmatrix}^* \tag{15}$$

Only the first 6 rows and 9 columns of the full 18x18 matrix are shown for conciseness as the remaining entries are zero. Similarly, by taking the derivative of equation set 5 with respect to equation 13, the $\tilde{H}_i$ matrices for sites 1, 2, and 3 are

$$\tilde{H}_{s1} = \begin{bmatrix} \frac{\partial \rho}{\partial x} & \frac{\partial \rho}{\partial y} & \frac{\partial \rho}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \rho}{\partial x_{s1}} & \frac{\partial \rho}{\partial y_{s1}} & \frac{\partial \rho}{\partial z_{s1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{\rho}}{\partial x} & \frac{\partial \dot{\rho}}{\partial y} & \frac{\partial \dot{\rho}}{\partial z} & \frac{\partial \dot{\rho}}{\partial \dot{x}} & \frac{\partial \dot{\rho}}{\partial \dot{y}} & \frac{\partial \dot{\rho}}{\partial \dot{z}} & 0 & 0 & 0 & \frac{\partial \dot{\rho}}{\partial x_{s1}} & \frac{\partial \dot{\rho}}{\partial y_{s1}} & \frac{\partial \dot{\rho}}{\partial z_{s1}} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^* \tag{16.a}$$

$$\tilde{H}_{s2} = \begin{bmatrix} \frac{\partial \rho}{\partial x} & \frac{\partial \rho}{\partial y} & \frac{\partial \rho}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \rho}{\partial x_{s2}} & \frac{\partial \rho}{\partial y_{s2}} & \frac{\partial \rho}{\partial z_{s2}} & 0 & 0 & 0 \\ \frac{\partial \dot{\rho}}{\partial x} & \frac{\partial \dot{\rho}}{\partial y} & \frac{\partial \dot{\rho}}{\partial z} & \frac{\partial \dot{\rho}}{\partial \dot{x}} & \frac{\partial \dot{\rho}}{\partial \dot{y}} & \frac{\partial \dot{\rho}}{\partial \dot{z}} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \dot{\rho}}{\partial x_{s2}} & \frac{\partial \dot{\rho}}{\partial y_{s2}} & \frac{\partial \dot{\rho}}{\partial z_{s2}} & 0 & 0 & 0 \end{bmatrix}^* \tag{16.b}$$

$$\tilde{H}_{s3} = \begin{bmatrix} \frac{\partial \rho}{\partial x} & \frac{\partial \rho}{\partial y} & \frac{\partial \rho}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \rho}{\partial x_{s3}} & \frac{\partial \rho}{\partial y_{s3}} & \frac{\partial \rho}{\partial z_{s3}} \\ \frac{\partial \dot{\rho}}{\partial x} & \frac{\partial \dot{\rho}}{\partial y} & \frac{\partial \dot{\rho}}{\partial z} & \frac{\partial \dot{\rho}}{\partial \dot{x}} & \frac{\partial \dot{\rho}}{\partial \dot{y}} & \frac{\partial \dot{\rho}}{\partial \dot{z}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \dot{\rho}}{\partial x_{s3}} & \frac{\partial \dot{\rho}}{\partial y_{s3}} & \frac{\partial \dot{\rho}}{\partial z_{s3}} \end{bmatrix}^* \tag{16.c}$$

It should be noted that calculations of the partials for equations 15 and 16 can be quite involved. To make sure the user can avoid this cumbersome task, BOPPA employs the 'jacobian' function within Matlab's Symbolic Toolbox to do all the heavy lifting. The user need only define the state and state space form of the dynamics and the symbolic variables involved.

D. Software Input & Output Guide

Now that the fundamental equations have been established, BOPPA's inputs and outputs can be defined. The large number of inputs has been specifically designed so that there is great variability in how the user wishes to model the dynamics, statistical parameters, and observations. With this in mind, the 'MasterLiveScript' file has been formulated to streamline this variable input process. Within this file, the user will see the below input values.

1) '**rv0**': A 6x1 column array containing the spacecraft's initial position [m] (rows 1:3) and velocity [m/s] (rows 4:6) in the ECI frame.

2) '**site**': A 3x3 matrix containing the initial positions of up to three sites within the ECEF frame. Each column corresponds to a particular site with each row corresponding to the x, y, and z position components [m] respectively.

3) '**tspan**': A time array [s] of variable length that is used to integrate the dynamics.

4) '**x0_bar**': An 18x1 array of variable units that describes the *a priori* state deviation estimate of the spacecraft. It has the form of equation 13.

5) '**P0_bar**': An 18x18 initial error covariance matrix.

6) '**R**': A 2x2 matrix that describes the noise of the observation data.

7) '**obs_data**': a matrix of variable length containing the time stamp [s] for each observation in column 1, the site number for each observation in column 2, and the range [m] and range rate [m/s] observations in columns 3 and 4 respectively.

8) '**n**': The number of iterations desired for the Batch Algorithm (3 recommended).

9) '**const**': A structure array containing all the necessary dynamic and reference constants to be used in computation.
   - '**const.mu**': Standard Gravitational Parameter [m^3/s^2]
   - '**const.J2**': Effects of Earth's Oblateness
   - **'const.Re**': Earth's Radius [m]
   - **'const.thetadot**': Earth's Rotation Rate [rad/s]
   - **'const.Cd**': The Satellite's Drag Coefficient
   - **'const.Ar**': The Satellite's Cross-Sectional Area [m^2]
   - **'const.m**': The Satellite's Mass [kg]
   - **'const.r0**': Reference Position [m]
   - **'const.rho0'**: Atmospheric Density Reference Value [kg/m^3]
   - **'const.H**': Reference Altitude [m]

Once the above inputs are defined and the 'MasterLiveScript' is run, the following will be outputted.

1) '**RMS**': A 2x**n** matrix containing the RMS values of the range and range rate residuals. Each column corresponds to a Batch iteration and rows 1 and 2 contain the range [m] and range rate [m/s] residual RMS values respectively.

2) '**x0_hat**': An 18x(**n**+1) matrix holding the state updates mapped back to the initial epoch. The nth column corresponds to the nth batch iteration and the final column is the total state change from the initial state.

3) **Residuals Plots:** A range [m] and range rate [m/s] residuals plot for each iteration.

4) **Position & Velocity Error Ellipsoid Plots:** A 3D plot for the position and velocity deviation error ellipsoid corresponding to the batch state-error covariance matrix.

5) **ECI Orbit Plot:** A 3D plot displaying the best-fit orbit that was determined from the final iteration of the Batch Algorithm.

IV. Batch Processor Algorithm

The Batch Processor Algorithm can be implemented now that the OD problem has been reduced to a linear estimation problem containing the state and observation deviations. A full derivation of the algorithm can also be found in Tapley *et al.*[2] (2004). Simply put, however, the Batch algorithm for OD can be explained in the following steps:

1) A 'batch' of observation data, the initial spacecraft state, and corresponding statistical parameters that reflect the observations are passed to the algorithm.

2) The initial spacecraft state and STM are propagated forward via numerical integration to form an initial trajectory estimation (Equation set 4).

3) The propagated state is transformed into the calculated observation variables at every epoch via the observation-state relationships (Equation set 5).

4) The error is quantified via the residuals (difference) between the 'batch' of observations and the calculated observations from step 3.

5) A minimum-variance criterion is introduced to adjust the orbital parameters and initial spacecraft state to minimize the residuals from step 4. This is the crucial state update that allows for convergence of the estimation.

6) The new parameters are then passed back through steps 1 through 5 until they are sufficiently small and convergence in the trajectory has occurred.

As discussed above, there exists a lengthy derivation of the procedure, however, the final result of the batch algorithm for which the crucial state update occurs is

$$(H^T W H + \bar{P}_0^{-1})\hat{\boldsymbol{x}}_0 = H^T W \boldsymbol{y} + \bar{P}_0^{-1}\bar{\boldsymbol{x}}_0 \tag{17}$$

where $\hat{\boldsymbol{x}}$ is the best estimate of the state deviation that minimizes the residuals between the weighted observed and the calculated observations. That is, $\hat{\boldsymbol{x}}$ is the solution corresponding to the minimum of the performance index given by

$$J(x) = (\hat{\boldsymbol{x}}_0 - \bar{\boldsymbol{x}}_0)^T \bar{P}_0^{-1}(\hat{\boldsymbol{x}}_0 - \bar{\boldsymbol{x}}_0) + \sum_{i=1}^{l} \hat{\boldsymbol{\epsilon}}_i^T R_i^{-1} \hat{\boldsymbol{\epsilon}}_i \qquad where \qquad \hat{\boldsymbol{\epsilon}}_i = \boldsymbol{y}_i - H_i \hat{\boldsymbol{x}}_0 \tag{18}$$

Within equation 17, the new variables W and $\bar{P}_0^{-1}$ are the observation weighting matrix and inverse of the *a priori* error covariance matrix respectively. It should be noted that W and R^{-1} within equations 17 and 18 are equivalent.

An edited flowchart for the full Batch Processor Algorithm from Tapley *et al.*[2] (2004) is provided in Fig. 2 below. The information matrix, Λ, and the normal matrix, *N*, have been introduced for convenience.

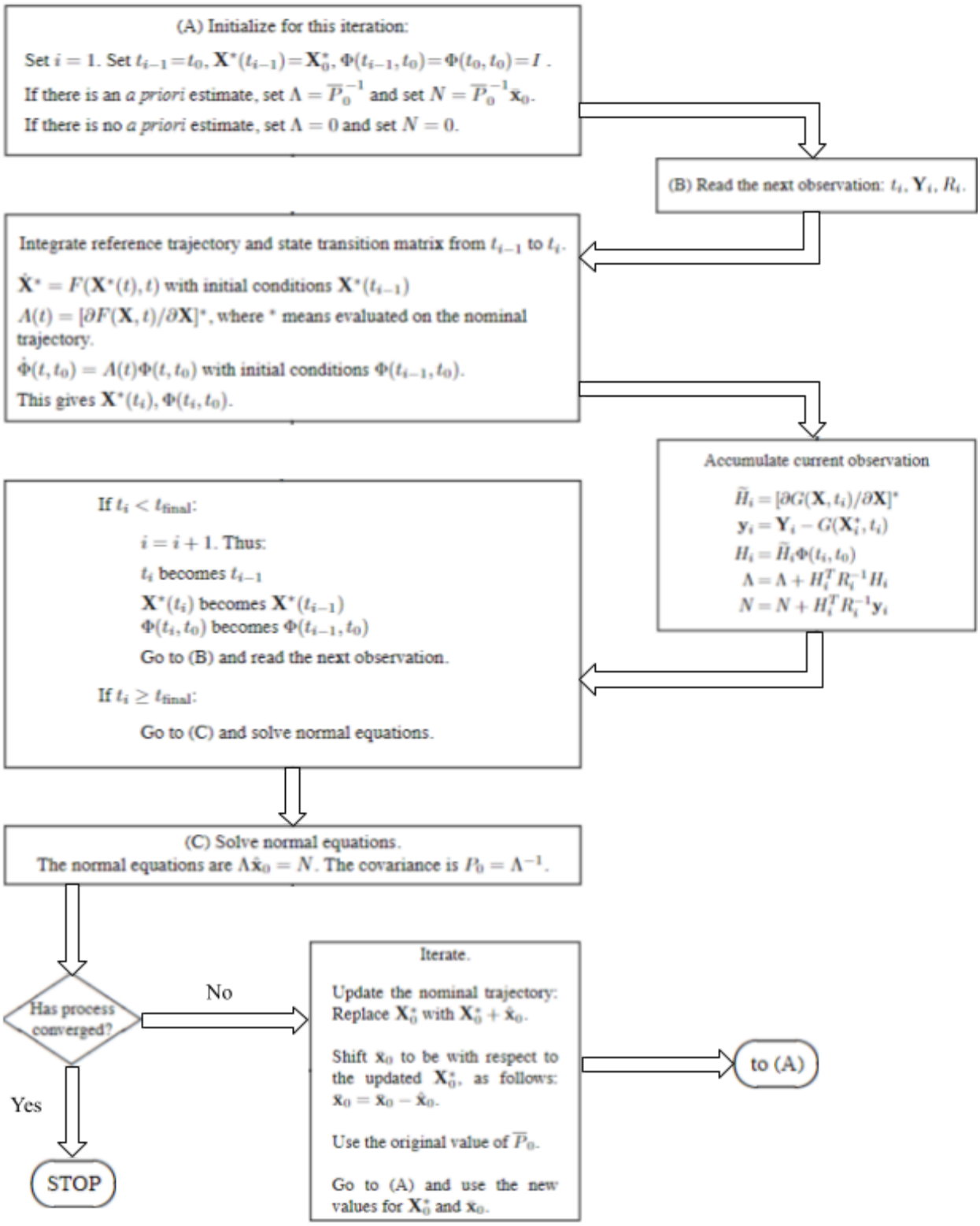

Figure 2[2]: Complete Batch Processor Algorithm Flowchart for the Orbit-Determination Problem

V. Results

To test and verify BOPPA's functionality and accuracy, the specific case of a LEO satellite is reintroduced. Modeled after NASA's QuikSCAT program, the occupied orbit is a sun-synchronous, retrograde, and nearly circular orbit with an inclination of 98.6 degrees. Over an interval of 18340 seconds, three ground sites took a total of 385 range and range rate observations with each observation time being limited to one site. The modeled governing dynamics follow what was mentioned in section III with rationale from Fig. 1. The following tables introduce the specific dynamical parameters, initial spacecraft state, and observation site coordinates used in the test.

Gravitational Parameter μ [m^3/s^2]	$3.986004415 \times 10^{14}$
Earth Oblateness Constant J_2	$1.082626925638815 \times 10^{-3}$
Earth's Radius [m]	6378136.3
Earth's Rotation Rate $\dot{\theta}$ [rad/s]	$7.2921158553 \times 10^{-5}$
Satellite Drag Coefficient C_D	2
Satellite Area [m^2]	3
Satellite Mass [kg]	970
Reference Range r_0 [m]	7078136.3
Reference Atmospheric Density ρ_0 [kg/m^3]	3.614×10^{-13}
Reference Altitude H [m]	88667

Table 1: Dynamical Paramters

	X	**Y**	**Z**
Position [m]	757700	5222607	4851500
Velocity [m/s]	2213.21	4678.34	-5371.3

Table 2: Initial Spacecraft ECI Position and Velocity

	X	**Y**	**Z**
Site 101 (fixed) Position [m]	-5127510	-3794160	0
Site 337 Position [m]	3860910	3238490	3988094
Site 394 Position [m]	549505	-1380872	6182197

Table 3: Observation Site ECEF Positions

An assumed 1 cm and 1 mm noise were introduced to the range and range rate data respectively. Thus, the weighting matrix follows as

$$W = R^{-1} = \begin{bmatrix} \frac{1}{\sigma_\rho^2} & 0 \\ 0 & \frac{1}{\sigma_{\dot\rho}^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{0.01^2} & 0 \\ 0 & \frac{1}{0.001^2} \end{bmatrix} \quad (19)$$

We also assume an accurate 18x1 *a priori* state deviation vector could not be found and was therefore set to

$$\bar{\boldsymbol{x}}_0 = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}^T \quad (20)$$

Lastly, the 18x18 *a priori* covariance matrix is defined as

$$\overline{P}_0 = \text{diag}\left(10^6, 10^6, 10^6, 10^6, 10^6, 10^6, 10^{20}, 10^6, 10^6, 10^{-10}, 10^{-10}, 10^{-10}, 10^6, 10^6, 10^6, 10^6, 10^6, 10^6\right) \quad (21)$$

where site 101's location is taken to be fixed. As a result, we associate near zero (10^{-10}) position covariance values. It should be noted that the covariance value associated with the gravitational parameter (μ) is considerably larger than the other values due to the parameter being on the order of 10^{14} itself. Furthermore, such drastic differences across the covariances cause the system to be poorly conditioned. Upon taking the required inverse per equation 17, certain elements may approach a divide-by-zero error causing the matrix to be close to singular. To get around this, the BOPPA software employs the '\' MATLAB command which is proven to provide smaller errors than the 'inv' command.

Should a different system be desired, the above statistical parameters must reflect the new corresponding symbolic state. For example, if a fourth observation site is introduced, equations 13, 14, 15, 16, 20, and 21 must all be expanded accordingly to include the three new site position components.

After inputting the data into the 'MasterLiveScript' file, the 'boppa' function is called and the results are outputted. The total state update mapped back to the initial epoch after 3 iterations is tabulated below.

X [m]	Y [m]	Z [m]	$\dot{X}$ [m/s]	$\dot{Y}$ [m/s]	$\dot{Z}$ [m/s]	μ [m^3/s^2]	J_2	C_D
0.2904	-0.4222	-0.2619	0.0406	0.0327	-0.0144	-4.277x10^7	-6.3x10^{-7}	0.1887

X_{s1} [m]	Y_{s1} [m]	Z_{s1} [m]	X_{s2} [m]	Y_{s2} [m]	Z_{s2} [m]	X_{s3} [m]	Y_{s3} [m]	Z_{s3} [m]
~0	~0	~0	-10.01	10.00	5.976	-5.009	2.021	2.976

Table 4: Total initial state update after 3 Batch Algorithm passes

To gain insight into the accuracy of this initial-state update, the residual plots for each Batch Algorithm pass are shown here.

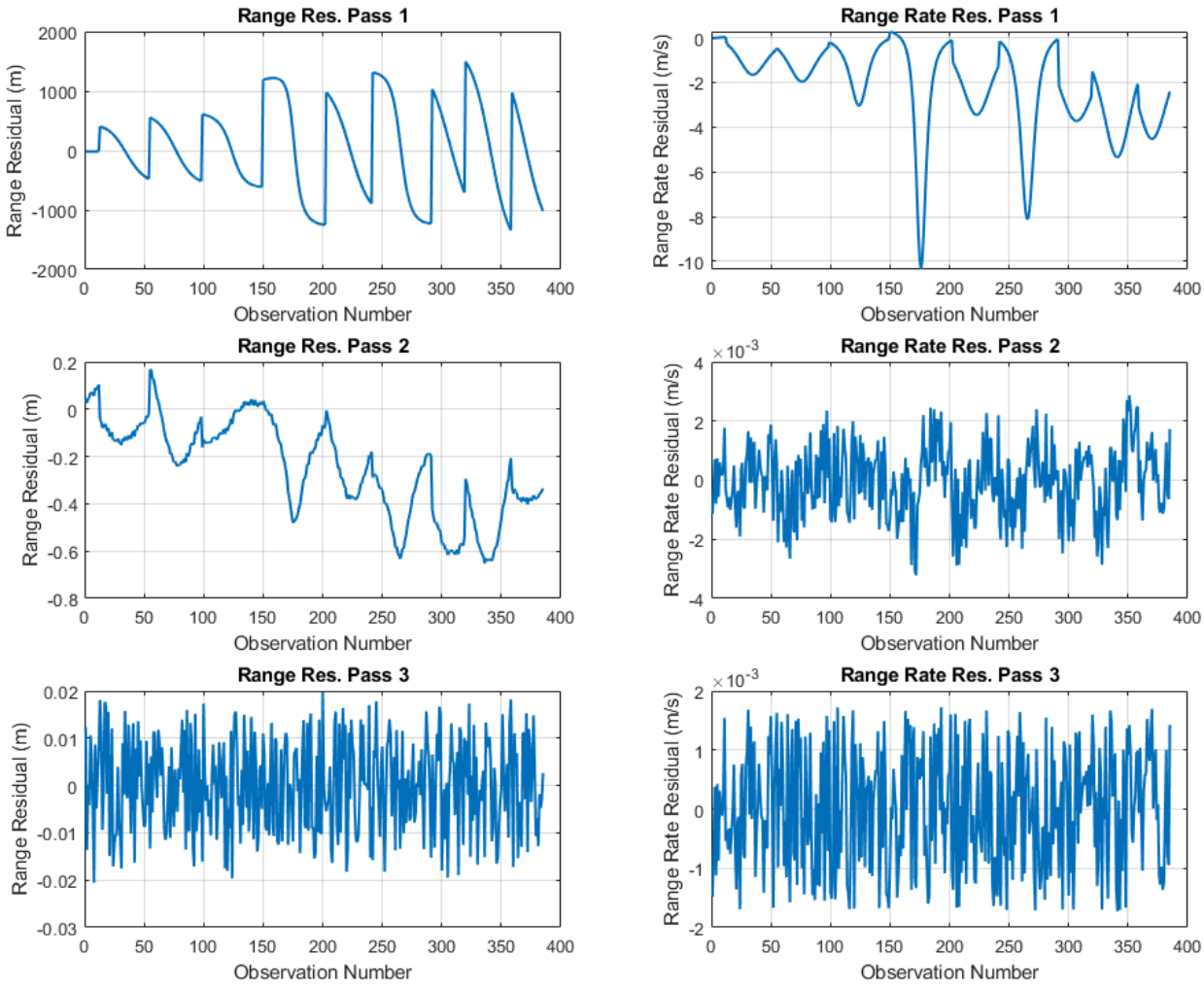

Figure 3: Range and range rate residuals for each pass through the Batch Algorithm

Pass 1 represents the residuals of the propagation from the inputted initial state before any state updates have been applied. For the range, a max residual of roughly 1500 m can be seen; for the range rate, a max residual magnitude of roughly 10 m/s can be seen. The pass 1 plots can also be characterized by the existence of systematic (non-random) errors due to the smooth lines that connect the residuals for groups of observations. This is in stark contrast to the pass 3 residual plots in which the trends are sharp and seemingly random. For this pass, a max range residual of roughly 0.02 m and a max range rate residual of roughly 0.0018 m/s can be seen. This random variation in the pass 3 plots shows that the residuals have gotten down to the 'white-noise' level of accuracy in which the error has transformed from systematic to random. As a result, any further passes through the algorithm will show no significant changes in the residuals. And, although not shown here, the plot outputted for a 4th pass would verify this claim. To better exemplify this, the range and range rate residual RMS values for 4 passes are tabulated below.

	Pass 1	Pass 2	Pass 3	Pass 4
ρ(RMS) [m]	732.75	0.3196	0.00972	0.00972
$\dot{\rho}$(RMS) [m/s]	2.9002	0.00120	0.000998	0.000998

Table 5: Range and range rate residual RMS values for 3 Batch Algorithm passes

As can be seen in Table 5, there is no change in the RMS values up to at least 3 significant figures between passes 3 and 4. In fact, a change in the range rate RMS value for pass 4 doesn't occur until 10 places after the decimal. As such, it is safe to assume that convergence has occurred and that any further residuals are a product of 'white-noise' error only.

As discussed in section III part D, the BOPPA software will also output the 3D error ellipsoids for the satellite's position and velocity deviations associated with the final iteration's covariance matrix. They are below in Fig. 4.

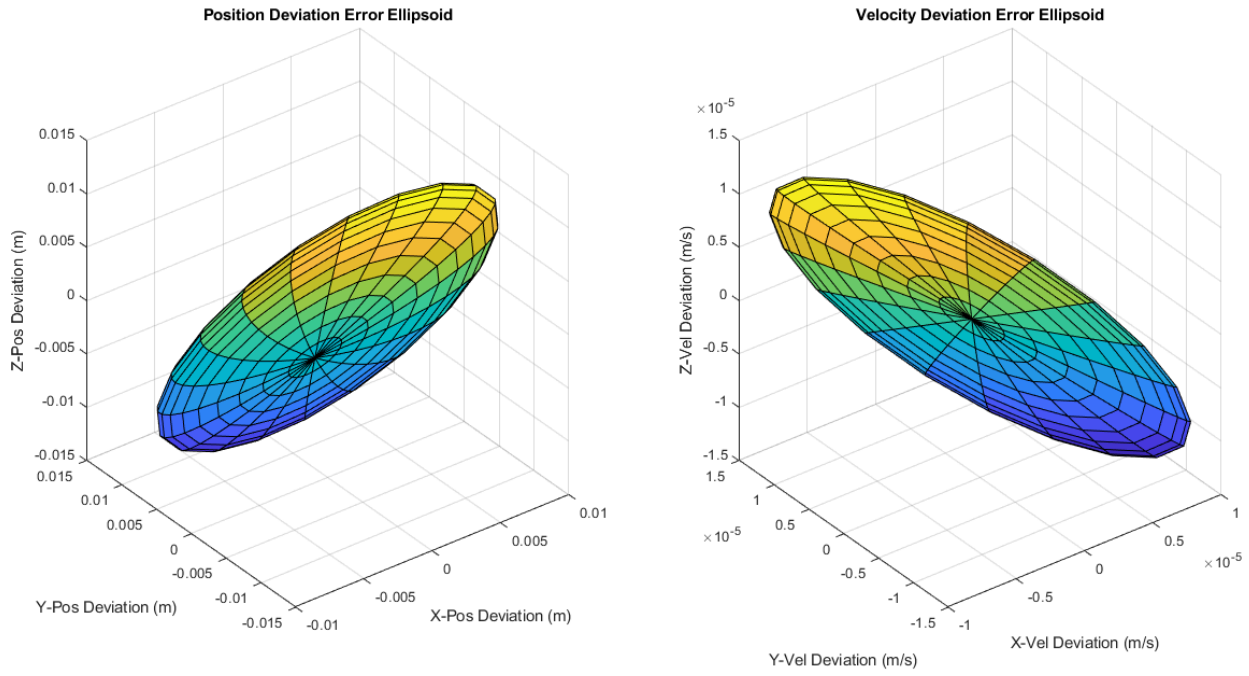

Figure 4: Position (left) and velocity (right) 3D error ellipsoids after 3 Batch Algorithm passes

The above figure allows for a visual representation of where the deviations could lie in 3D space. That is, the satellite's position and velocity could vary according to any point within the corresponding ellipsoid presented in Fig. 4. The level of variation is given by the axes of the plots; for the worst case, the position could vary by around 1.5 cm in any direction and the velocity could vary by roughly 15 μm/s in any direction.

Finally, a plot of the satellite's orbit corresponding to the final Batch Algorithm pass is outputted and shown below in Fig. 5.

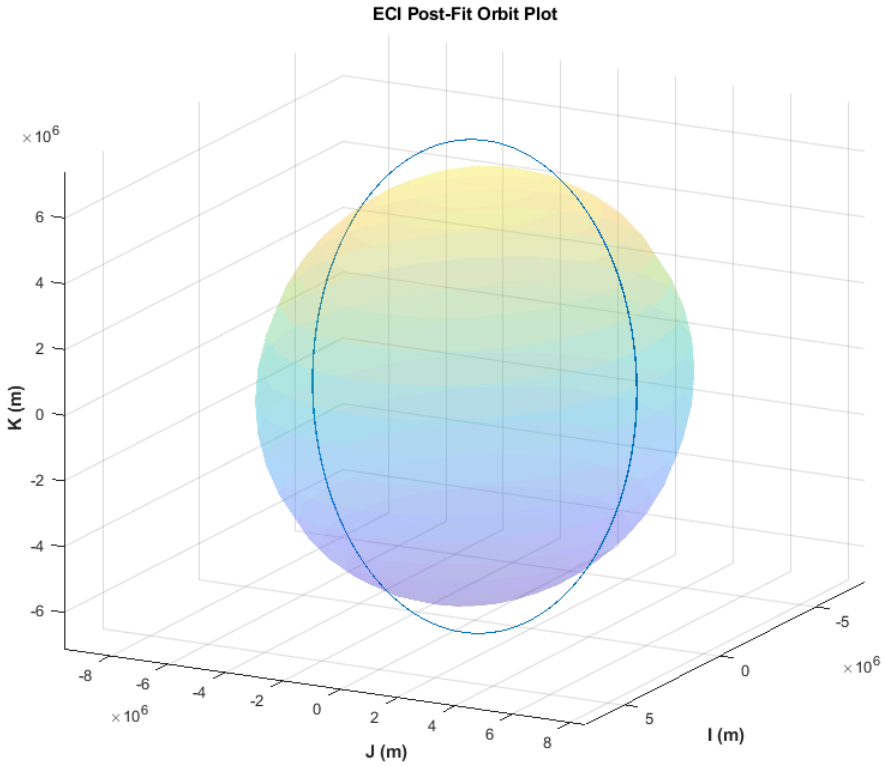

Figure 5: Propagated orbit in the ECI frame after 3 Batch passes

As expected, the orbit is low altitude (LEO), nearly polar, and nearly circular. While not visible in the above figure, there is orbital precession due to the effects of J_2 and atmospheric drag. A zoomed-in view of the orbit trail would show distinct paths that represent the precession as the satellite completes more orbits around the Earth.

VI. Conclusions

It's important to note that the above results, while good, were based on several assumptions. For instance, although not completely random, the values for the *a priori* covariance matrix were set to arbitrary values that were thought to best represent the error in the initial state. These so-called 'best guess' values turned out to be accurate enough to ensure convergence within 3 iterations as shown in Fig. 3. However, had the guess been sufficiently poor, it's possible that convergence could have occurred at larger residual values or, in certain cases, may have not even been possible. Such a result is a product of voiding the linear assumption made within the linearization process. That is, the covariance would represent an error in the initial state that is no longer within the linear regime of the actual trajectory. Conversely, had the *a priori* covariance been set to values similar to those obtained after 3 iterations of the Batch Algorithm, convergence could have happened in as little as 2 iterations. While insignificant in this example, reducing the number of iterations for every orbit determination done could amount to significant savings in computational cost and time.

We also assumed that observation site 101's position in the ECEF frame was exact; this was reflected in the near zero *a priori* covariance matrix values for the site. The reason for doing this

was so that the site could serve as a reference for the rest of the unknowns. Had every site been treated as varying with corresponding covariance values, the possible deviations in the satellite's position and velocity would have been considerably larger. This would be exemplified by greatly increased error ellipsoids that would allow for deviations with orders of magnitudes larger than those seen in Fig. 4. It should be noted that which site is fixed is irrelevant as all will lead to allowable deviations on the same orders of magnitudes. Generally speaking, however, it makes sense to fix the site for which the position data holds the most confidence. Fixing a site also serves to avoid degeneracy within the problem. That is, should every site be unknown, it becomes possible for there to be a plethora of solutions that fit the supplied observation data (hence, the wrong final orbit could be propagated). However, one must also be careful not to fix every observation site as this can lead to greatly increased error within the estimation. Similarly, if the site coordinates are taken out of the state vector (not solved for), the state vector will become incomplete and one risks adding significant error to the estimation due to the loss of this information.

Overall, section V shows results of high precision that were calculated with great computational efficiency. Despite assuming no *a priori* initial state deviation and a 'best guess' estimate for the covariance values, the BOPPA software was able to get position and velocity deviations within roughly 1.5 cm and 15 μm/s respectively in as little as 3 iterations. With thousands of some of our most important satellites occupying this region of space, the BOPPA software's precision in determining the orbits of LEO satellites demonstrates it as an indispensable tool for any agency, company, or researcher within the realm of space mission analysis and design.

VII. References

[1] Rosengren, Aaron. "Lesson 1: Introduction of the GNC System." MAE 182 Spacecraft Guidance and Navigation, 9 Jan. 2024, UC San Diego. Keynote presentation.

[2] Tapley, Byron D., et al. "Chapter 4: Fundamentals of Orbit Determination." *Statistical Orbit Determination*, Elsevier Academic Press, Amsterdam, 2004, pp. 159–200.

* While never explicitly copied or quoted, ChatGPT was used to verify/substantiate certain claims made throughout the report. Similarly, ChatGPT was used to lend help in how to code the error ellipsoids presented in Fig. 4. However, no code was ever taken directly from the AI (none of it actually worked), and all code presented within the scripts is mine and mine alone.