SGP4 Orbit Determination

David A. Vallado[†]
Center for Space Standards and Innovation, Colorado Springs, Colorado, 80920

Paul Crawford[‡]
University of Dundee, Dundee, UK

In 2006, an updated version of SGP4 was presented for general use. The propagation routine relies on Two-Line Element (TLE) set data provided by the U.S. Government for several decades. Many independent organizations have access to observations, or ephemerides developed with other programs. Because the TLE is a compact means to obtain modestly accurate and fast calculations, it is desirable to have an orbit determination package that is compatible with the aforementioned code. This paper presents the code, test cases, and discussion of computer code to produce TLE data from an externally derived ephemeris. Initial code is also developed for processing raw observations.

I.   INTRODUCTION AND HISTORY

The Simplified General Perturbations (SGP) model series began development in the 1960s (Lane 1965), and became operational in the early 1970s (Lane and Cranford, 1969). A variety of publications presented the mathematical theory, and the paper by Vallado et al. (2006) tracks the history of these publications. Unfortunately, there has never been a release of any kind of differential correction code to implement the SGP4 method in a systematic approach to create Two-line Element (TLE) data. With the increased number of observing sites, and the availability of low-cost high quality optical observations, it is desirable to have such codes. The primary uses would be to obtain a more accurate TLE from independent data, and to have the ability to examine covariance data to support mission operations (such as conjunction operations) . [§]

A.Motivation

Spacetrack Report Number 3 (Hoots et al 1980) noted the importance of using the specific equations and data input to ensure proper operation and we repeat it here. "*The most important point to be noted is that not just any prediction model will suffice… The NORAD element sets must be used with one of the models described in this report in order to retain maximum prediction accuracy.*" This compatibility applies for the SGP4 propagation code, as well as the orbit determination code. We noted several minor points in the original SGP4 paper in which the performance of SGP4 could be improved. To maximize the usefulness of these features one should ideally use TLE formed with differential correction using an identical model as well. Thus, this paper provides a way to accomplish this action.

As with the original 2006 SGP4 code release, we have tried to make the code compatible with the existing TLE data released by NORAD. However, feedback from the community over the last few years has suggested improved methods of operation. The 2006 code has been updated and although most of the changes are minor, the full code, updated paper and test cases are available on the web at http://www.celestrak.com/publications/AIAA/2006-6753/. A new factor is that a switch is added to allow one to use the "standard" AFSPC mode of operation, or an improved mode that the user can experiment with.

The expanded conjunction processing work (for example the innovative work of SOCRATES to generate TLE on TLE conjunction notifications for several years now) could benefit substantially from estimates of the accuracy of the TLE data. Kelso (2007) has shown the ability to generate improved TLE data using externally generated

---

[†] Senior Research Astrodynamicist, Analytical Graphics Inc., Center for Space Standards and Innovation, 7150 Campus Dr., Suite 260, Colorado Springs, Colorado, 80920-6522. Email dvallado@centerforspace.com. Phone 719-573-2600, direct 610-981-8614, FAX 719-573-9079, AIAA Associate Fellow.
[‡] Principal Engineer, psc@sat.dundee.ac.uk.
[§] The standard TLE are routinely available from CelesTrak (http://CelesTrak.com) and the Air Force Space Command (AFSPC) (www.space-track.org).

ephemerides. If this could be coupled with estimates of the uncertainty, it would be a great increase in performance for the conjunction calculations, and the subsequent knowledge that decision makers have at their disposal. This paper presents a discussion of the covariance data that is generated through the orbit determination process.

II.   ORBIT DETERMINATION REVIEW

The underlying mathematics behind Differential Correction (DC) (also called Orbit Determination, OD) are covered in detail in Vallado (2007:Ch 10). Figure 1 shows the basics. Although it is not obvious or often described as such, the DC process is basically a multi-dimensional Newton-Raphson (NR) root solving method of $y = f(x)$ with a least-squares statistical treatment of the "known data" (y).
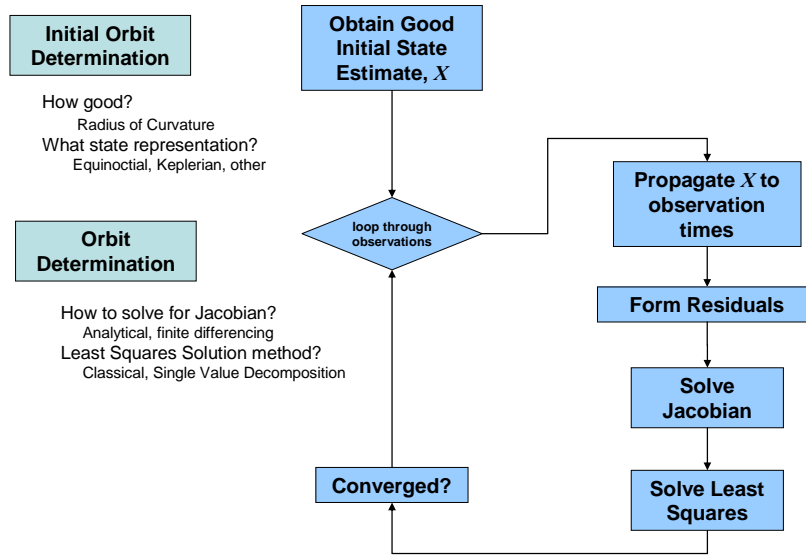

Figure 1. Orbit Determination. The overall process of orbit determination is summarized here. There are a few choices that can be made. Initial orbit Determination can be a very complex topic. We do not dwell on this aspect much in this paper.

We use the following nomenclature. The code uses these names as well where practical.
- X
  - State of orbital model, either XYZ Cartesian position and velocity vectors or Keplerian orbital elements, etc.
- A
  - Partial derivative matrix (Jacobian) relating the calculated "observation" value to the initial model state value. Typically we have the following when an analytical approach is used to compute A. $A = \dfrac{\partial obs}{\partial X_0} = H \cdot \Phi = \dfrac{\partial obs}{\partial X} \cdot \dfrac{\partial X}{\partial X_0}$
- W
  - Weighting matrix of observation accuracy, mathematically this is the diagonal matrix of $1/\sigma^2$ but normally it's stored as the vector of the diagonal terms for efficient computation
- $\underline{b}$
  - Residual vector (obs – calculated, sometimes as $\delta y = Y_{observed} - Y_{calculated}$)
- $(A^T W A)^{-1}$
  - Covariance matrix

If we expand the process, we can produce an algorithm of pseudo-code that details the process (Algorithm 1) which covers the general case of any orbital model. There are several decisions which must be made with respect to actually implementing the algorithm and here, the specific behavior of SGP4 may become a consideration.

We can identify a few aspects where the detail is important to the success and efficiency of the implementation, these are:

- Generating the initial state X_0 as accurately as practical is important, since the differential correction process converges rapidly for 'small' errors but can diverge spectacularly outside of this region (as for all NR techniques).
- The choice of how **X** is represented is also important, since certain formulations lead to less problems with near-singular matrices to be solved.
- The Jacobian matrix *A* can be generated by various techniques, we must consider the mathematical effort to implement a fast 'analytic' formulations versus the computation effort for the 'brute force' approach to estimate it by finite differencing methods.
- Solving the least-squares problem can be done in several ways, some are easier to program, others are more robust if the solution is singular or nearly singular.
- Testing for convergence and method of containing the NR's tendency to jump too far may seem trivial, but can be very important to a robust and reliable system.
- If we are doing this to form TLE from known good ephemeris sets, we must also consider the optimum time scale for fitting the data according to the end user's requirements (e.g. whole day for un-modeled or partially modeled forces such as $J_{2,2}$ or solar gravity, several days for accurate atmospheric drag with higher orbits, etc).

The normal equation for linear systems is $X = (A^T WA)^{-1} A^T Wb$. However, because the orbit problem is highly non-linear, we use the non-linear form. $\delta x = (A^T WA)^{-1} A^T W\underline{b}$. Note that the $A^T WA$ and $A^T W\underline{b}$ matrices are accumulated, therefore avoiding large matrices for storage and inversion. The Single Value Decomposition (SVD) processing is shown as an alternative in red, along with other choices that can be made for the solution.

__Algorithm 1: SGP4 Differential Correction__ (obs, $X_{nom} \Rightarrow X_o$)

- FOR i = 1 to the number of observations (*N*)
    - Propagate the nominal state X_0 to the time of the observation using SGP4 (result is X_i in TEME coordinate system)
    - Transform the TEME prediction X_i in to the correct coordinate system and type of available observation data **obs_i** (usually topoocentric)
    - Find the *b* matrix as observed – nominal observations
    - Form the *A* matrix (select 1 of 2 approaches)
        - Finite (or central) differences
        - Analytical partials $\qquad\qquad A = \dfrac{\partial obs}{\partial \hat{X}_o} = \dfrac{\partial obs}{\partial \hat{X}} \dfrac{\partial \hat{X}}{\partial \hat{X}_o} = H\Phi$
            - *H*, Partials depending on observation type
            - Φ, Partials for state transition matrix.
    - Accumulate $\sum A^T WA$ and $\sum A^T Wb$
- END FOR
- Find $P = (A^T WA)^{-1}$ using Gauss-Jordan elimination (LU decomposition and back-substitution)
- Solve $\delta x = P\, A^T W\underline{b}$
- Check RMS for convergence
- Update state $X_0 = X_0 + \delta x$
- Repeat if not converged using updated state

On completion the term $P = (A^T WA)^{-1}$ is the covariance matrix.

If using SVD, replace the matrix inversion with the following
- Use SVD to decompose the weighted Jacobian matrix $[S \cdot A]$ into *U*, *V*, and *w*, where ($S = A^T W$)
- Inspect w_j, set any 'very small' w_j terms to zero rather than any $1/w_j$ term near-infinite
- Solve $\delta x = V \cdot (\text{diag}[1/\, w_j]) \cdot U^T \cdot S \cdot \underline{b}$
On completion the covariance matrix is simple to compute from $V \cdot (\text{diag}[1/\, w_j]) \cdot U^T$

B. Choice of State Elements

We have a choice of elements to use in the state. A choice of position and velocity vectors is sometimes envisioned, however the parameters all change rapidly (over the course of a revolution), and this is undesirable for a differential correction process. Keplerian orbital elements (a, e, i, Ω, ω, M) can be used, but there can be singularities with some types of orbits. Notice the use of the semimajor axis instead of the mean motion. This can make a change with some implementations with respect to numerical precision. Equinoctial elements afford a little more stability in terms of singular orbits, but may still encounter problems. We have included options for Keplerian, Equinoctial elements, and state vectors in the code. We show alternate variable names, but prefer the k_e, h_e, etc set as they have fewer ambiguities with other notations in astrodynamics.

$$
\begin{aligned}
a_f &= k_e = e\ \mathbf{COS}(\Omega + \omega) \\
a_g &= h_e = e\ \mathbf{SIN}(\Omega + \omega) \\
&\quad a \\
L &= \lambda_e = M + \omega + \Omega \\
\chi &= p_e = \mathbf{TAN}(i/2)\ \mathbf{SIN}(\Omega) \\
\psi &= q_e = \mathbf{TAN}(i/2)\ \mathbf{COS}(\Omega)
\end{aligned}
\tag{1}
$$

The inverse functions are needed for transformations and they are found as follows: (using $\mathrm{TAN}(i/2)$)

$$
\begin{aligned}
e &= \sqrt{h_e^2 + k_e^2} \\
i &= 2\arctan\left(\sqrt{p_e^2 + q_e^2}\right) \\
\Omega &= \arctan\left(\frac{p_e}{q_e}\right) \\
\omega &= \arctan\left(\frac{h_e}{k_e}\right) - \arctan\left(\frac{p_e}{q_e}\right) \\
M &= L - \arctan\left(\frac{h_e}{k_e}\right)
\end{aligned}
\tag{2}
$$

C. Calculation of the Jacobian Matrix (*A*)

There are several techniques to find the *A*-matrix, ranging from simple finite differencing to complex numerical integration. See Vallado (2007:Sec 10.7) for development of the various approaches for the options shown in Fig. 2. Elegant operational systems generally employ the numerical integration approach, however, with a simple analytical propagation theory (SGP4) it may make sense to simply use finite differencing. After all, the time (and money) spent developing the partial derivatives for each observational data type, state form, etc. to reduce the computational processing by perhaps a factor of 2, is often negated by simply allowing an extra iteration or so with a finite differencing approach. In addition, the partial derivatives for each force model are not needed as they are *all* included in each finite differencing calculation.

Operation	Simple	Analytical	Constant System	Complex
$X = X_o$		Kepler, FKepler, etc.		Numerical
H	N/A	$H = \dfrac{\partial \text{ observations}}{\partial X}$ (analytic)		
Φ	N/A	$\vec{r}(\tau) = \vec{r}_o(t) + \dot{\vec{r}}_o(t)\tau + \dfrac{\ddot{\vec{r}}_o(t)\tau^2}{2!} + \dfrac{\dddot{\vec{r}}_o(t)\tau^3}{3!} + \cdots$ $\Phi = \dfrac{\partial \vec{X}}{\partial \vec{X}_o} = \begin{bmatrix} \dfrac{\partial \vec{r}}{\partial \vec{r}_o} & \dfrac{\partial \vec{r}}{\partial \vec{v}_o} \\ \dfrac{\partial \vec{v}}{\partial \vec{r}_o} & \dfrac{\partial \vec{v}}{\partial \vec{v}_o} \end{bmatrix}$	$F = \dfrac{\partial \dot{X}}{\partial X} = \begin{bmatrix} \dfrac{\partial v_I}{\partial r_I} & \dfrac{\partial v_I}{\partial r_J} & \dfrac{\partial v_I}{\partial r_K} & \dfrac{\partial v_I}{\partial v_I} & \dfrac{\partial v_I}{\partial v_J} & \dfrac{\partial v_I}{\partial v_K} \\ \dfrac{\partial v_J}{\partial r_I} & \cdots & & & & \\ \vdots & & & & & \vdots \\ \dfrac{\partial a_K}{\partial r_I} & \dfrac{\partial a_K}{\partial r_J} & \dfrac{\partial a_K}{\partial r_K} & \dfrac{\partial a_K}{\partial v_I} & \dfrac{\partial a_K}{\partial v_J} & \dfrac{\partial a_K}{\partial v_K} \end{bmatrix}_{ITRF}$ $\Phi = \left[I + F\Delta t + \dfrac{1}{2!}F^2\Delta t^2 + \dfrac{1}{3!}F^3\Delta t^3 + \cdots \right]$ $\dot{\Phi}(t, t_o) = F(t)\Phi(t, t_o)$ $\Phi(t_o, t_o) = I$	
A	Finite Differencing	$A = H\Phi$		

Figure 2. A Matrix Calculation Options. This table lists four options for employing least squares techniques. The complex method numerically integrates the equation for **F**. Note that we can also approximate the observation (**H**), partial derivative (**F**), and error state transition (**Φ**) matrices by finite differencing, as well as the **A** matrix.

The general process of finite differencing relies on perturbing the nominal state, element by element, and then comparing to the nominal state results. In the case of SGP4, we simply perturb each element, initialize the newly perturbed TLE, propagate to the observation time, and then perform any coordinate system conversion and difference the perturbed and nominal results. Figure 3 shows the notional concept.
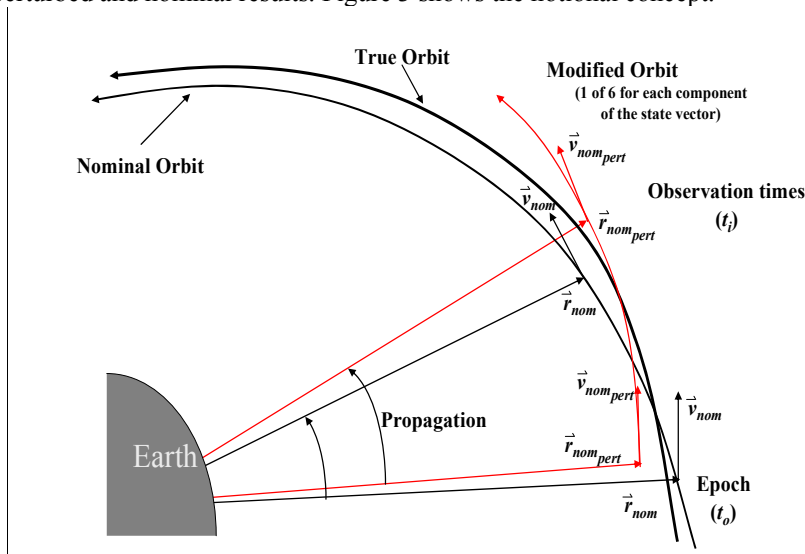

Figure 3. Finite Differencing. Forming the A matrix with finite differencing requires numerous propagations, but no analytical partial derivatives. The nominal state is moved to each time of the observations. Then, each component of the state is perturbed and the resulting state is propagated to the same observation time. This allows the nominal and perturbed values at each observation time to form a portion of the partial derivative matrix.

D. Solving the matrix inversion

There are several techniques to solve the least-squares problem of $y = A.x$ and any good text book on numerical analysis will cover those in reasonable details, for example (Press *et al.* 1992). Here we have only covered the two common methods as the choice:

- LU decomposition and back-substitution
- SVD decomposition and solution.

Traditionally the LU method was favored as it lends itself to efficient use in terms of computer memory and CPU load for the operations required. However, if the matrix (A) is singular or near-singular, the classical LU method (and others) can fail completely or result in very inaccurate answers. The common use of double precision in modern software helps push up the matrix solution accuracy in near-singular cases, but this is not dealing with the underlying problem. Typically what you get in near-singular cases is some solved parameters with impracticably huge values delicately canceling each other. Marshall (1999) investigated this and reached the same conclusions for the Navy operational system.

The use of SVD has the major advantage that (theoretically at least) is *cannot fail* to produce a useful solution (one without 'canceling infinities'). The disadvantage is the much greater CPU load and memory demands, but those are no longer significant because memory is now large, cheap, and the N^3 CPU load is still less than the generation of the A matrix for most reasonable numbers of observations (say, N less than 1K – 10K points).

The fact that SVD offers a stable solution should not be treated as an excuse for ignoring the underlying problem, that of certain solved-for parameters not being well defined by the available observations or by the orbital model itself (e.g. BSTAR drag coefficient with high altitude orbits). This type of difficulty is far more common than it initially appears, but can often be helped by a good choice of state representation X.

III.   COMPUTER CODE DEVELOPMENT

The computer code developed in this paper is provided in c++. Although some structures and classes were used, the code was intentionally written without some of the shortcuts available in c++. Conversion to other languages should be aided by the structuring effort that has been performed on the code. Several items apply.

- The overall structure is designed to follow the development and nomenclature of Vallado (2007, Sec 10.4).
- Available input data types are range, azimuth, and elevation observations, position and velocity vector ephemerides, and TLE data. The TLE data is for test and development only as if you have a TLE, you don't need to create another one! This did prove to be a valuable tool in getting the orbit determination to work properly, and to do the initial setup on the options available in the code.
- The program can use TLE variables (Kepler), or equinoctial elements as the nominal elements. Having this variety of state types made it easier to evaluate the performance of each.
- A matrix inversion technique (LU back substitution) is implemented along with a Single Value Decomposition method (Press et al. 2002). Both have positive features.

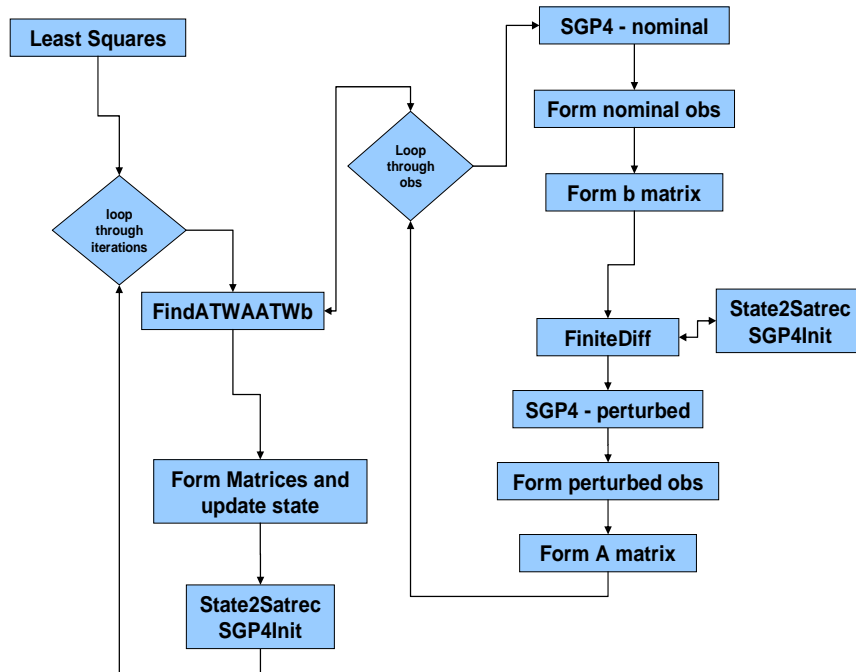The structure of the computer code is shown in Fig. 4 below.

Figure 4. SGP4DC Structural Organization. The computer program structure is shown above. There are two major loops – one for the iterations in the least squares, and one for the observations. The FiniteDiff routine performs the finite differencing by individually perturbing each element of the state.

Several miscellaneous features are implemented in the code to make the operation more robust, and to allow the user additional options in solving particular problems.

 1. There is an option (opsmode, 'a' or 'i') to select the AFSPC or improved mode of operation for SGP4.

 2. You can select the type of observations to process (typeodrun, 'r', 'x', or 't'). These represent range, azimuth, elevation (or right ascension-declination), position and velocity vectors in an ECI, ECEF, or TEME coordinate system, and the TLE values themselves.

- Input a TLE, create the initial state vector and take that as the initial guess for the TLE, develop an ephemeris from the TLE (unperturbed), and try to recover the original TLE. This test is useful because we know the answer. Note that the semimajor axis does not need to be updated (Kozai to Brouwer) because it is already in the Kozai frame. The initial state is simply treated as a single point Kepler translation and is therefore significantly farther off the actual value than a result taken by simply perturbing the individual orbital elements.

- Take an ephemeris (in TEME, ECEF, or ECI), assemble an initial TLE from the first state including the correction for the Kozai-Brouwer semimajor axis, and then try to determine a TLE that best fits that ephemeris.

- Process observations. This requires determining an initial orbit (not addressed in this paper as it's an Initial orbit determination technique – see Vallado (2007, Ch 7). The initial orbit state vector is converted to a TLE (including the Kozai-Brouwer conversion for semimajor axis) , and the TLE is formed via the orbit determination process. Although the code is provided, it is not fully functional at this time and will be updated either with another paper if the changes are substantial, or via the web if it is more minor changes.

 3. The state type determines what type of orbital elements will be used for the whole process. (statetype, 'e', or 't'). Either TLE variables (essentially Kepler variables) or equinoctial variables may be used.

 4. The percentage change for the finite differencing appears to be an important variable in the solution accuracy for the routine (**percentchg = 0.001**). The rationale behind using a percentage was to

limit any uneven effect in perturbing each of the orbital elements. However, when a variable was found to have too small a difference (note this also used the difference amount described next), just that element was changed, and not the whole formation of the *A*-matrix for that time.

5.   The finite differencing of each state element for the *A*-matrix may result in too small a difference on particular variables, in which case you could get a divide by zero. This can be handled by including a loop in the perturbation of the nominal state to test for these small quantities and simply increase the percentage adjustment for the offending element (the last item discussed above). A limit of 5 iterations is used initially, but this is completely arbitrary (**deltaamtchg = 0.0000001**).

6.   The rms tolerance to stop the iterations was another important feature that was worked on (rmsepsilon = 0.002). Of course, you need to have a maximum number of iterations as you don't want the process to get stuck in an infinite loop! You also can evaluate the observation matrix (*b*) and note that it represents the differences of the nominal and observed orbits. With the proper scaling (weighting matrix), you can form a "sigma" value for each iteration. This gives an indication of how the process is going, and can be tested. The test used is

$$\frac{sigmanew - sigmaold}{sigmaold} \qquad\qquad (3)$$

Note that this worked a majority of the time. However, many situations arose where the tolerance (rmsepsilon) was too large, and the sigma values were large or small enough that their relative change did not trigger the exit from the routine. In these cases, the iterations generally reached the maximum number of iterations, and the difference to the initial vector was zero. Although this is a nice computational result, it's not terribly efficient, and so a third check was instituted in which the sigma values were compared directly to the rmsspeilon value. This caused the process to exit when sufficient accuracy had been reached, but the other tests did not trigger any stop.

while ((fabs((sigmanew-sigmaold)/sigmaold) >= epsilon) && (loops < 25) && (sigmanew >= epsilon));

7.   For testing purposes, command line arguments were programmed to permit rapid execution with scripts to examine the sensitivity of various inputs.

8.   The correction to each parameter for each iteration was limited to prevent elements from changing to fast. Two approaches were explored. The first used a default limit set to 40%. Thus, if any element was calculated to change by more than 40%, only 90% of that change was selected for implementation. Note that 90% was also set through various tests. A further refinement was needed for cases in which Bstar was being solved for as it could change quite rapidly. For this case, if the parameter change was greater than 1.0, only 40% of the change was used. The code is as follows where dx is the correction on each iteration. An alternative test was tried, but not found to be globally successful. This code is shown commented out below.

```
     if ( fabs(dx[i][0]) > fabs(0.4 * xnom[i]) )
       {
        if (fabs(dx[i][0]/xnom[i]) > 1.0)
          {
           dx[i][0] = dx[i][0] * 0.4;   // usually needed for bstar solutions, 0.01 for obs?
//            dx[i][0] = 0.10 * xnom[i] * sgn(dx[i][0]);
          }
        else
          {
           dx[i][0] = dx[i][0] * 0.9;
//            dx[i][0] = 0.9 * xnom[i] * sgn(dx[i][0]);
          }
       }
```

The second approach, used as our default, tried a more objective method, just looking at the magnitude of the calculated change. We found that most of the changes were relatively small fractions of the nominal values (as expected). However, in cases where the orbit was diverging, or about to diverge, the ratios quickly become large. The levels below are somewhat arbitrary, and not fully researched to determine the effect on the overall solution. However, as we note later, these two approaches solve all but a handful of the satellites in a full catalog, including the tough geosynchronous low inclination orbits. We suspect a

combination of the two, or a better binning of the ratio differences may produce an optimal single approach for solution. The code for this approach is as follows with dx as the correction for each iteration, and xnom as the nominal state vector.

```
if (fabs(dx[i][0]/xnom[i]) > 10.0)
        if ( (loops > -1) && (fabs(dx[i][0]/xnom[i]) > 1000.0) )   // 100
          dx[i][0] = 0.10 * xnom[i] * sgn(dx[i][0]);  // 0.30 try leaving the same
         else
        if ( (loops > 0) && (fabs(dx[i][0]/xnom[i]) > 200.0) )   // 100
         {
           dx[i][0] = 0.30 * xnom[i] * sgn(dx[i][0]);   // 0.30 try leaving the same
          else
          if ( (loops > 0) && (fabs(dx[i][0]/xnom[i]) > 100.0) )  // 20
            dx[i][0] = 0.70 * xnom[i] * sgn(dx[i][0]);   // 0.70 - 0.80 about same
          else
          if ( (loops > 0) && (fabs(dx[i][0]/xnom[i]) > 10.0) )     // 5
            dx[i][0] = 0.90 * xnom[i] * sgn(dx[i][0]);   // 0.90 try leaving the same
         }
```

9.   The fit span is very important for all batch least squares applications. We initially used a baseline with 72 points at 20 minute intervals for all satellites. This worked on some satellites, but was clearly not as efficient as possible given GEO and LEO orbits have different fit span requirements. Thus, we changed to using 2 orbital periods, again with 72 points per revolution. It performed much better. However, do not confuse this fit span with an "operational" fit span. Typically satellites use a 3-4 day fit span in operations. We can afford a reduced time span because we will have continuous observations throughout the orbit (using the satellite ephemeris).

10. The ability to solve for BStar, or not, is important. Because AFSPC uses BStar even for GEO orbits, it's important to have the capability to solve for this parameter, as it acts to soak up force model deficiencies in the propagation routine. In some cases, solving for BStar is important and will provide better results. However, in other cases it can cause the iterations to diverge. Thus, the option is included in the SGP4DC program.

11. The full catalog cases showed that the semimajor axis, eccentricity, and mean motion could all cause problems (even with the iteration limitations discussed earlier), although there is probably a tie between the two. This, we implemented a sim0lke approach that limited each orbital element if they were outside certain bounds. The three checks in the computer code are as follows.

```
if (satrec.a < 1.0)  // can't be less than the radius of the earth
 {
   satrec.a = satrecorig.a * 1.01 ;  // er  1.05, .9
   xnom[2] = satrec.a;  // remember to change both!!!
 }

if (satrec.no < 1.0e-5)
 {
   satrec.no = satrecorig.no * 0.9 ;
   satrec.a = pow( 1.0 / (satrec.no * satrec.no * 13.446839 * 13.446839) , 1.0 / 3.0 );  // er
   xnom[2] = satrec.a;  // remember to change both!!!
 }

if (satrec.ecco > 1.0)
 {
   satrec.ecco = satrecorig.ecco * 0.9;
   // be careful of order here…
   // could iterate, but since we're just changing values, it's not needed
   xnom[0] = ( satrec.ecco * cos(satrec.argpo + satrec.nodeo) ) * scalef[1][0];   // ke, af
   xnom[1] = ( satrec.ecco * sin(satrec.argpo + satrec.nodeo) ) * scalef[1][0];    // he, ag
 }
```

IV.   SOLUTION and SENSITIVITY SETUP

We include several stressing test cases (twoline.txt). The TLE's in this file were taken from the original SGP4 verification tests, from some difficult orbits found in random testing, and from the full catalog testing we describe later in the paper. The idea was to establish a set of test cases that would exercise each of the relevant features of the code. The file also served as a quick way to evaluate what the entire catalog performance would be without having to process over 15000 objects.

An initial set of tests looked at perturbing the elements of the TLE. The elements were changed as follows. Note that the mean anomaly was not changed as the node and argument of perigee contained sufficient differences to the satellite position.

<div align="center">

mean motion * 0.999
eccentricity * 0.999
inclination – 0.1 deg
right ascension of the node * 1.0002
argument of perigee * 1.0001

</div>

These differences resulted in the initial vectors being off from about 5 to 20 km for the initial guess. Note that some of the cases had substantially higher initial differences. However, this gave optimistic results, and actually presented too narrow a radius of curvature for solution. Thus, we adopted an approach where the initial TLE state was used, in a Keplerian sense, as the initial guess. This resulted in initial differences of 15 to 40 km (and more). We felt this was a more realistic situation.

The test matrix in Table 1 was used to develop the initial parameters for the full catalog testing.

Table 1. Initial Test Options. The following parameters were changed to identify shifts in the performance of the orbit determination technique. Each column provides some of the values explored. The finite difference options are indicated by "fd". The number of iterations did not substantially change the results. The initial state perturbation had the largest effect on the overall solution as expected. The highlighted cases were the remaining parameters for which tailoring was indicated. These parameters then became inputs into the program for further testing.

Elem type	LS iter	LS limit correction each iter	fd % chg for each elem	Amount to test small pert in fd calc	RMS to stop	Solve for Bstar	fd iter if loop for small pert
equin	**25**	**0.40 and *0.9 ***	**0.001**	**0.0000001**	**0.02**	**6**	5
tle	35	0.10 and *0.9	0.01	0.000001	0.002	7	10
		0.15 and *0.9	0.0001	0.00001	0.0002		
		Levels					

The highlighted columns were the biggest contributors to the success of the orbit determination. Of course, the rmsepsilon tolerance requires more iterations as it decreases, but the other highlighted parameters also had an effect on the number of iterations.

Remember that all batch least squares techniques suffer from determining an accurate fit span. Thus, we tried arbitrarily setting the fit span to 2 orbital periods, with a total of 72 observation points per revolution. This is about 2 minutes for LEO satellites, and about 20 minutes for GEO satellites. The importance of choosing the proper amount of data became apparent in the ephemeris file testing, but for the initial TLE evaluations, we simply used a constant 2 orbital period fit span, with 72 points per period.

V.   FULL CATALOG TESTING

To test the overall performance of the method, we tested against the full satellite catalog. The sample catalog contained 16235 objects. Table 4 shows the results of various options for the processing of each satellite. Note that the really 'good' fit you normally get to a TLE-generated ephemeris is somewhat false, but nevertheless still an indication of the DC process working. From most orbits we get very good TLE-TLE fit (less than a meter).

Table 2. Test Results. The complete catalog was tested against the parameters for change. The results were binned into several regions. These all use the "a" operation sgp4 mode which emulates the believed implementation of AFSPC. The highlighted columns represent "baseline" parameters chosen for generally good performance over the whole catalog. The remaining columns simply highlight the parameter that was changed from those baseline configurations.

	.0001 .0000001 .0002 / 7 state, 72 points 0.4, 0.9	.001 .0000001 .0002 / 7 state, 72 points svd, new	.01 .0000001 .0002 / 7 state, 72 points 0.4, 0.9	.001 .000001 .0002 / 7 state, 72 points 0.4, 0.9	.001 .00001 .0002 / 7 state, 72 points 0.4, 0.9	.001 .0001 .0002 / 7 state, 72 points 0.4, 0.9	.001 .0000001 .002 / 7 state, 72 points 0.4, 0.9	.001 .0000001 .02 / 7 state, 72 points 0.4, 0.9
< 1m	16183	16198	16067	16163	16161	16145	16040	15053
1m << 10 m	15	4	9	13	17	22	130	1013
10m << 100 m	8	2	10	7	6	9	4	94
100 m < 1 km	6	6	9	7	7	7	5	9
1 km < 10 km	9	20	9	5	7	6	7	16
> 10 km	14	5	131	40	37	46	49	50
Avg iterations	9.49	4.01	4.46	9.39	18.98	23.63	4.51	4.05

	.001 .0000001 .0002 / 7 state, 72 points 0.4, 0.9	.001 .0000001 .0002 / 7 state, 50 points 0.4, 0.9	.001 .0000001 .0002 / 7 state, 100 points 0.4, 0.9	.001 .0000001 .0002 / 6 state, 72 points svd, new		.001 .0000001 .0002 / 6 state, 72 pts, tle state 0.8, 0.1		.001 .0000001 .0002 / 6 state, 72 points, 0.8 and 0.1
< 1m		16166		16181		16166		16061
1m << 10 m		7		21		7		95
10m << 100 m		5		2		5		28
100 m < 1 km		5		7		5		36
1 km < 10 km		3		14		3		6
> 10 km		49		10		49		9
Avg iterations		4.98		4.31		4.98		4.99

The essential problem is to determine the radius of convergence for the various cases, although this can be quite difficult. These test permitted several tests to be included in the code to check for conditions such as zero mean motion, and negative eccentricity. We also experimented with changing the statesize to 6 (if it was 7) in cases where the iterations diverged on 3 successive iterations. This helped a few problem cases.

The performance of the 6-state case is shown in Fig. 5 below. Notice that the iterations and the difference between the original and converged TLE are shown vs the apogee and perigee values. While there were many cases in which the circular orbits required additional iterations, most resulted in a very small final difference. We tried to be computationally efficient, but it wasn't our overall goal – we were primarily interested in the accuracy of the solution, and thus, a few additional iterations was not considered as detrimental.
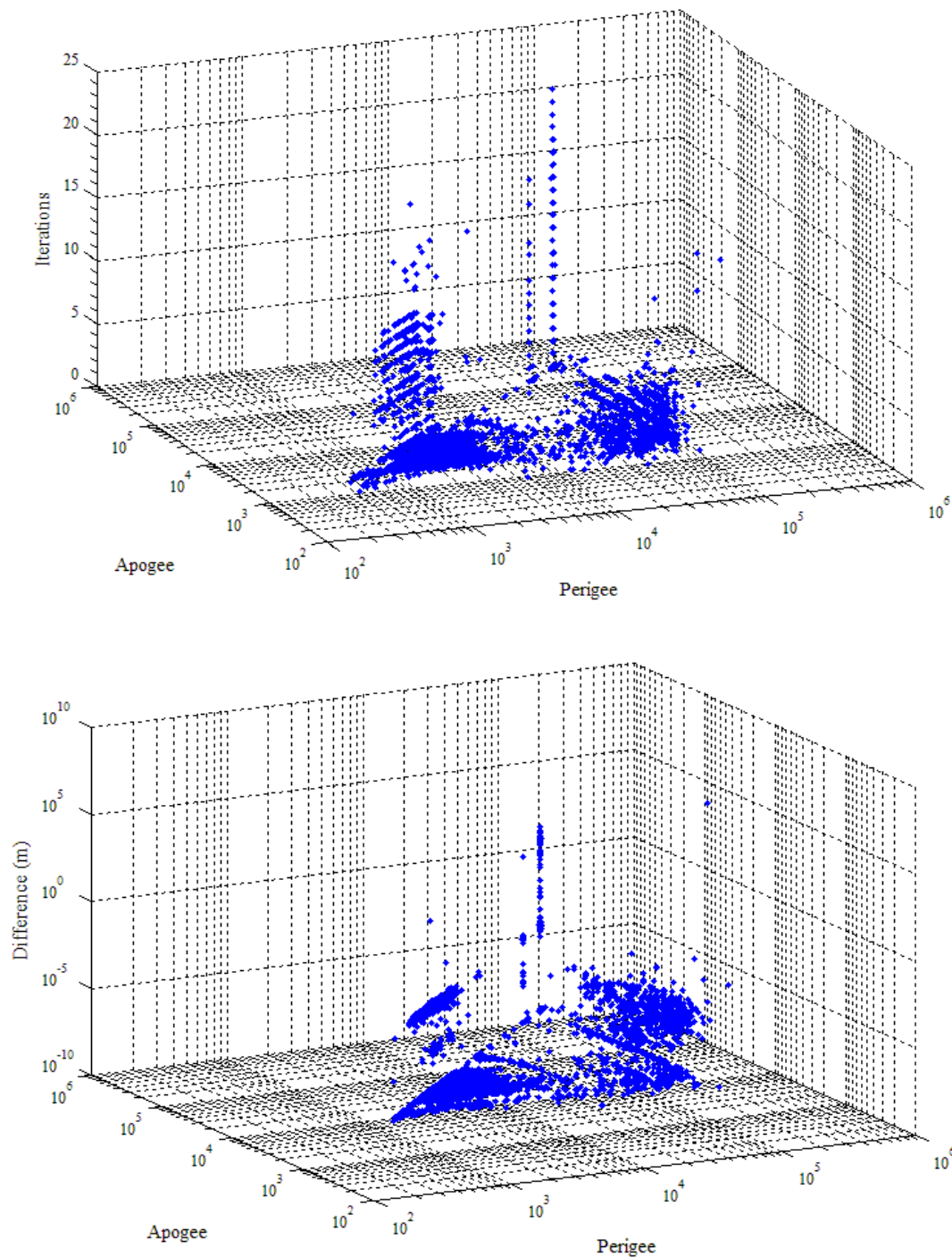
Figure 5. SGP4DC Full Catalog Test. The iterations and the final difference from the original TLE is plotted against the apogee and perigee altitudes in km. Note that the difference plot the bottom line at 0.0000001 is artificial as those values were set (from 0.0) to permit plotting on the log scale. Below about 10 m, there are many cases between the extremes in iteration, while at larger differences, the discreet cases are easier to identify.

A histogram reveal the nature of the solutions, as shown in Fig. 6. You can see that the majority of the cases were handled with the default configuration.
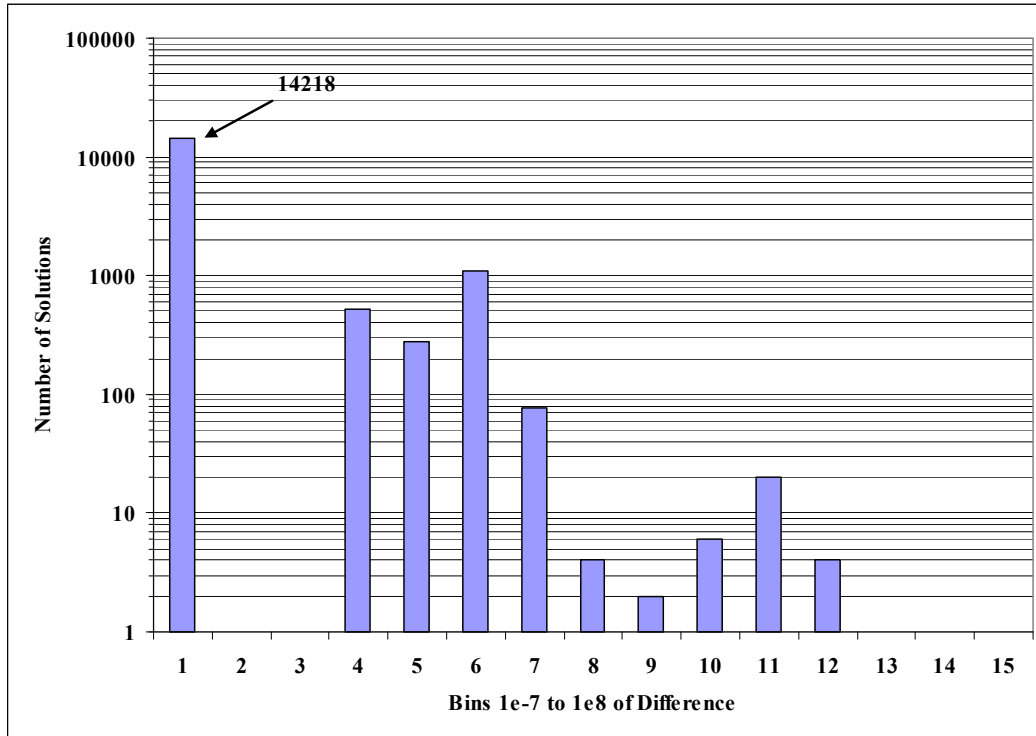
Figure 6. SGP4DC Full Catalog Test distribution. A simple bar chart is shown for the full catalog test and bins of the final difference. The 14218 number represents the number of cases that were solved to under 0.0001 mm.

There was a group of 24 problem cases examined for the full catalog – those with differences above 10 m. Most of these are Geosynchronous satellites with low inclinations, shown in Table 3.

Table 3. Difficult Cases Test Results. These satellites exhibited additional uncertainty in solution when processing the full catalog test. Note that most are geosynchronous, low inclination cases, and most required all 25 (maximum) iterations.

iter	norad #	init diff (m)	final diff (km)	a (km)	e	i (deg)	ap alt	pr alt
12	28707	19322	151.786626	42164.790000	0.0002420	0.0805	35798	35777
25	26451	16289	210.260239	42164.290000	0.0002690	0.0137	35798	35776
20	23864	15422	455.774603	42163.640000	0.0001700	0.0178	35794	35779
25	27298	15261	750.837509	42163.610000	0.0003960	0.0243	35803	35770
6	22028	18674	779.249604	42501.060000	0.0004730	6.4477	36144	36104
14	23598	16468	976.903621	42164.140000	0.0003060	0.0410	35800	35774
14	31102	22250	1134.548046	42163.880000	0.0002150	0.0105	35796	35778
25	26608	21962	1236.703197	42163.900000	0.0004450	0.0205	35805	35768
25	28911	18333	1515.689957	42163.980000	0.0003500	0.0172	35802	35772
25	28911	18333	1515.689957	42163.980000	0.0003500	0.0172	35802	35772
25	23864	16956	2871.535703	42163.600000	0.0001670	0.0168	35793	35779
2	28915	7428	3223.733382	25507.480000	0.0008170	64.8649	19151	19109
8	27378	15630	3570.763967	42163.420000	0.0001910	0.0368	35794	35778
25	28937	19982	3956.492409	42163.830000	0.0003010	0.0288	35799	35774
25	24732	22838	5131.566915	42164.460000	0.0002060	0.0019	35796	35779
25	27811	20743	5397.006447	42163.060000	0.0003140	0.0226	35799	35773
25	26451	17747	5944.783917	42164.200000	0.0002700	0.0142	35798	35776
25	28154	18420	6124.595781	42163.580000	0.0002730	0.0293	35798	35775
25	25237	18731	6167.139594	42163.570000	0.0006310	0.0192	35813	35760
5	28893	15108	6309.126644	7071.060000	0.0016310	98.1064	705	682
25	27298	20138	6473.030673	42163.620000	0.0003960	0.0239	35803	35770
25	25913	16139	7348.898831	42163.240000	0.0001380	0.0397	35792	35780
8	32479	14184	7604.600156	41149.800000	0.0190430	0.0000	35556	33989
25	28393	15255	8091.215056	42163.540000	0.0002430	0.0180	35797	35776
25	29398	18562	8582.413235	42164.190000	0.0004390	0.0340	35805	35768
5	25049	17590	9444.154045	42163.170000	0.0001610	0.0058	35793	35779
25	27378	16362	14672.238391	42163.350000	0.0001950	0.0349	35794	35778
25	27820	22150	17041.820184	42164.070000	0.0003080	0.0074	35800	35774

25	26590	22855	20121.270679	42163.530000	0.0003070	0.0190	35799	35773
25	29045	16362	31956.465732	42163.790000	0.0002060	0.0019	35795	35778
7	25068	782815	845118631.410830	40200.160000	0.8286580	7.5609	67136	510

The interesting thing to note with this approach is that using a previous approach at limiting the corrections, we obtained a different set of problem cases. This indicates that careful selection of the limiting features of the corrections, and of the semimajor axis, eccentricity, and mean motion is the key to solving most problems. Note between Tables 3 and 4 that there are very few duplicate satellites. In fact, the only ones that appear in both are low inclination geosynchronous satellites (23598, 23864, 24732, 25068, 25237, 26451, 26590, 26608, 27298, 27378, 27811, 28393, 28911, 29398) and one LEO satellite (28893).

Table 4. Difficult Cases Test Results. These satellites exhibited additional uncertainty in solution when processing the full catalog test and when using the second approach for limiting the least squares corrections. Note that most are geosynchronous, low inclination cases, and most required all 25 (maximum) iterations.

25	29398	18562	14.096002	42164.19	0.000439	0.0340	35805	35768
25	23598	16549	15.806848	42164.19	0.000303	0.0387	35800	35774
25	28094	17422	32.263324	42164.35	0.000293	0.0347	35799	35775
25	23864	15422	34.781485	42163.64	0.000170	0.0178	35794	35779
25	32018	20168	104.571616	42163.73	0.000016	0.0331	35787	35786
25	26608	21962	227.816075	42163.90	0.000445	0.0205	35805	35768
25	26451	17747	259.881170	42164.20	0.000270	0.0142	35798	35776
25	24732	22838	834.589047	42164.46	0.000206	0.0019	35796	35779
25	27811	20743	1310.247496	42163.06	0.000314	0.0226	35799	35773
25	23175	17677	2316.348449	42163.45	0.000176	0.0026	35794	35779
25	26590	22855	4581.524603	42163.53	0.000307	0.0190	35799	35773
25	26554	22731	4718.374626	42164.25	0.000069	0.0364	35790	35784
25	28393	15255	6006.229271	42163.54	0.000243	0.0180	35797	35776
25	27298	15261	10246.052988	42163.61	0.000396	0.0243	35803	35770
25	28911	18333	13850.143187	42163.98	0.000350	0.0172	35802	35772
25	32487	20534	19310.755321	41967.19	0.000171	0.0209	35597	35583
25	24880	19388	19735.845732	42163.93	0.000211	0.0208	35796	35778
25	25237	18731	47392.755888	42163.57	0.000631	0.0192	35813	35760
25	23949	21601	50307.705194	42163.36	0.000043	0.0606	35788	35784
5	25068	782815	1523081.031231	40200.16	0.828658	7.5609	67136	510
3	23866	101380	25408756.47274	22827.78	0.707519	4.5589	32602	299
3	28893	15108	3935246.472610	7071.06	0.001631	98.1064	705	682
3	30582	2336029	54747675.90313	66714.35	0.869927	4.7533	118375	2300

It is interesting that the value of the deltaamtchg would make an effect as it should be essentially controlled by the machine accuracy. However, it does have an effect on the solution. Values of 0.000 01 and 0.000 001 seemed to work for some of the problem cases.

VI.   VERIFICATION TEST CASES

The full catalog testing provided a starting point for assembling a set of tests cases that would be challenging. We also used the original SGP4 test cases as a point of continuity. These cases allowed the main features of SGP4DC to be tested. The file (twoline.txt) is on the Internet at the web site listed at the end of the paper, and in the Appendix.

The particularly tough cases helped us decide which parameters to expose in the program for user selection. Some experimentation was also done with the Geosynchronous/low inclination cases and modest success resulted from zeroing the lunar-solar terms in the initialization. This is a potential feature which could be incorporated under the "improved" SGP4 mode of operation. It also appears that the initial uncertainty (essentially the weighting matrix) had a substantial input to the convergence.

From these results, we chose an initial configuration for the full catalog testing. The items for selection are: (baseline values are in bold)

Orbital element type – **equinoctial** or TLE (Keplerian)
Finite difference % chg for each element, (**0.001**)
Amount to test small percentage in finite differencing calculations, (**0.000 000 1**)
RMS to stop, (**0.0002**)
Solve for BStar, (**6 or 7**)

VII.   EPHEMERIS FILE TESTING

We used several ephemerides for testing here as we envisioned this will perhaps be the most popular use for the program. The ephemerides were generally accurate to a few cm and thus could be considered truth for our purposes.

the Geosynchronous ephemerides were accurate to about a hundred meters or so. The resulting TLE's often performed significantly better than the existing TLE data – something also documented by Kelso (2007). Here we examine three ephemerides for different orbital regimes; an ICESat ephemeris, a GPS ephemeris, and several GEO ephemerides. Each configuration used the standard setup (0.001, 0.000 000 1, 0.0002, and 6 (or 7) statesize). We tried some shorter fit spans, but the prediction results were better if we extended the fit span slightly. These fits are not optimal, but seemed to give modest results. Options are included in the code to thin the data if it's very dense, and to fit a certain number of points, while reading many more points. Finally, take note of the vertical (difference) axis as we sometimes use log sales to better depict the results.

For ICESat, the original ephemeris was in Earth Fixed coordinates, and thus, all comparisons were done in ITRF. Notice that the original fit (the initial vector is epoched to the beginning of the interval) was about 500-600 m yet after about 2 days, the error had only grown to about 4-5 km. This is fairly good for a LEO satellite.
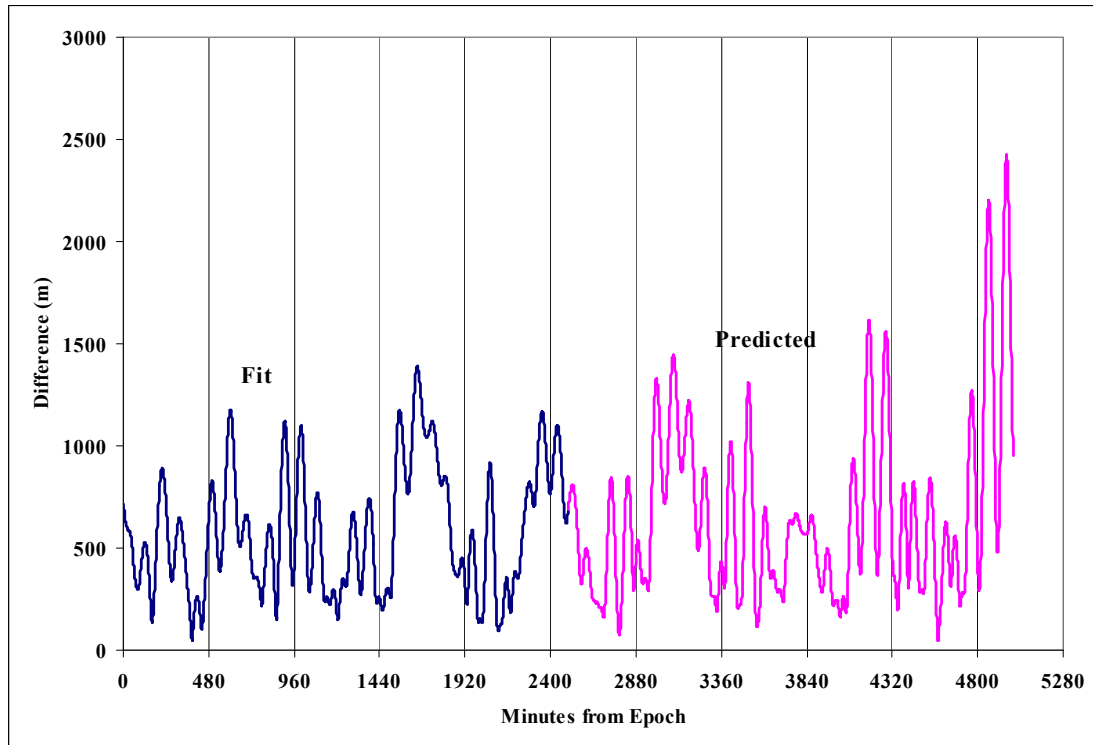

Figure 7. ICESat Solution Performance. The differences of the converged TLE to the original ephemeris are shown. The difference is the RSS of the three positional components. The fit span was about 2500 minutes (every other observation), and a 7-state solution.

Next, we examined a GPS satellite and we extended the prediction interval to show the performance. The original ephemeris was in Earth Fixed coordinates. With about a three-day fit of data every 15 minutes, the initial fit was good to about 150 m. For the prediction, there is reasonable performance up to about 30 days past the original fit span.
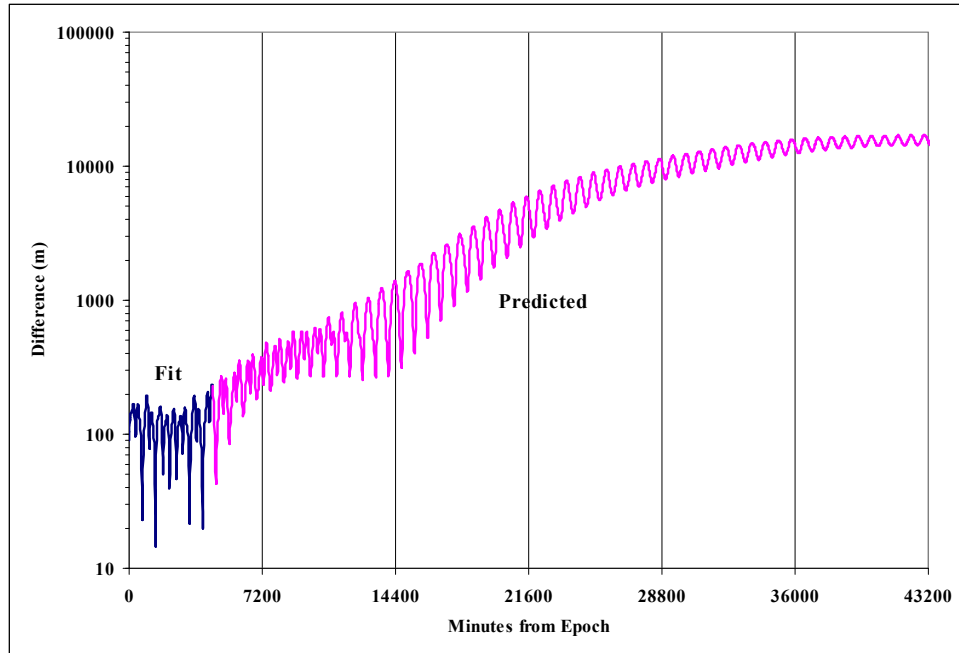
Figure 8. GPS Solution Performance. The differences of the converged TLE to the original ephemeris are shown. The difference is the RSS of the three positional components. The fit span was about 4500 minutes. Notice that the final error is only about 10 km after nearly 30 days.

We had several ephemerides for Geosynchronous satellites from the Intelsat constellation. These satellites have maneuvers which are shown in each of the figures. The time step for each ephemeris is about 15 minutes. Each ephemeris was in Earth Centered Inertial (IAU-76/FK5 J2000) coordinates. There were numerous known maneuvers in the data, and cases were taken to process before, during, and after these maneuvers. In Fig. 9, we look at a solution with maneuvers during the fit span. The overall accuracy doesn't appear to be impacted, nor was the prediction. The magnitude of the maneuver is clearly a factor here though.
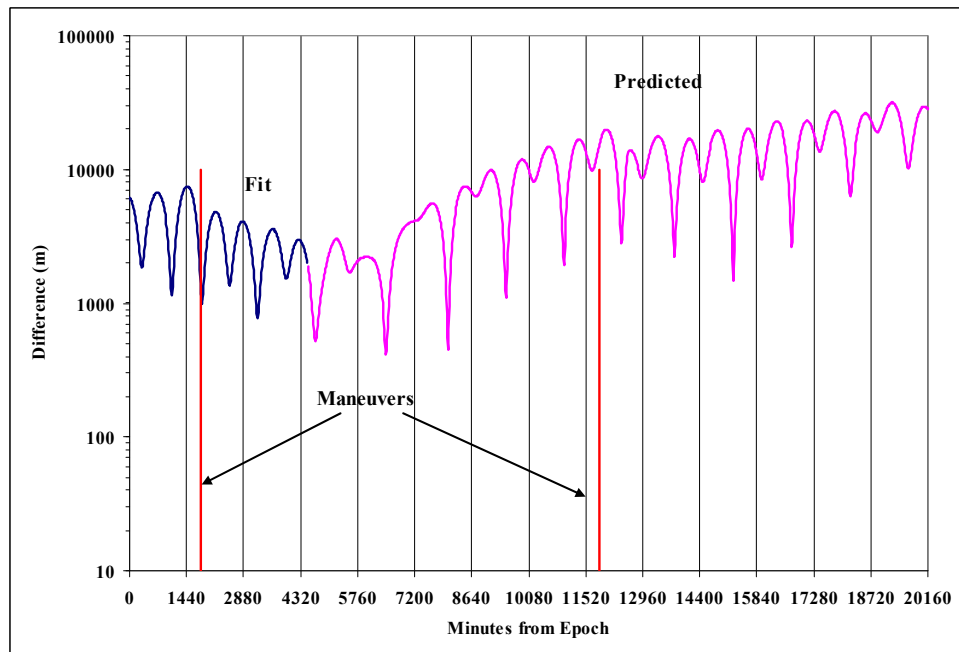

American Institute of Aeronautics and Astronautics

Figure 10 shows a case where we processed before any maneuvers, and as expected, the initial fit was much better. The appearance of maneuvers didn't affect the prediction accuracy tremendously, again depending on the magnitude of the maneuvers.
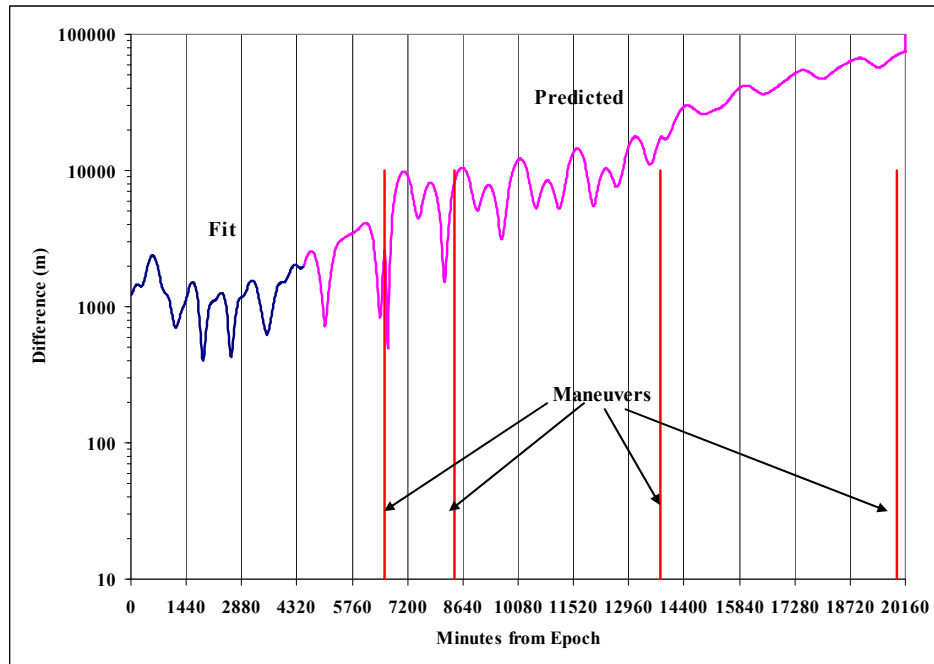

Figure 10. Intelsat 805 Solution Performance. The differences of the converged TLE to the original ephemeris are shown. The difference is the RSS of the three positional components. The fit span was about 4500 minutes.

We tried a case in Fig. 11 where the fit ended just as a maneuver occurred. The step function disconnect is obvious. and this mode of operation is not recommended.
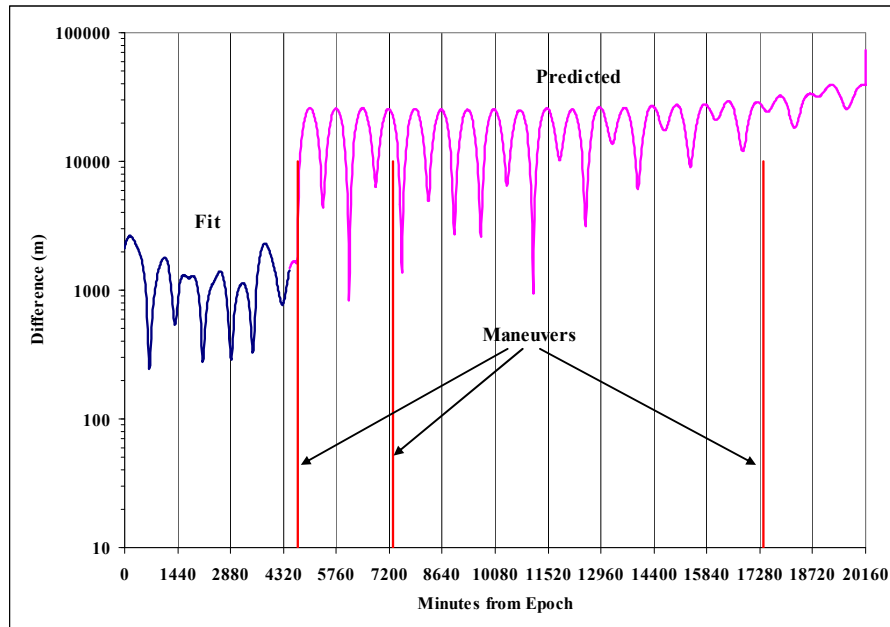
Figure 11. Intelsat 906 Solution Performance. The differences of the converged TLE to the original ephemeris are shown. The difference is the RSS of the three positional components. The fit span was about 4500 minutes.

Figure 12 shows a case where the satellite had a few small maneuvers during the fit, and then a period of no maneuvers, followed by a series of several small maneuvers. Notice that the overall accuracy is reasonably good throughout the period, because of the small magnitude of the maneuvers.
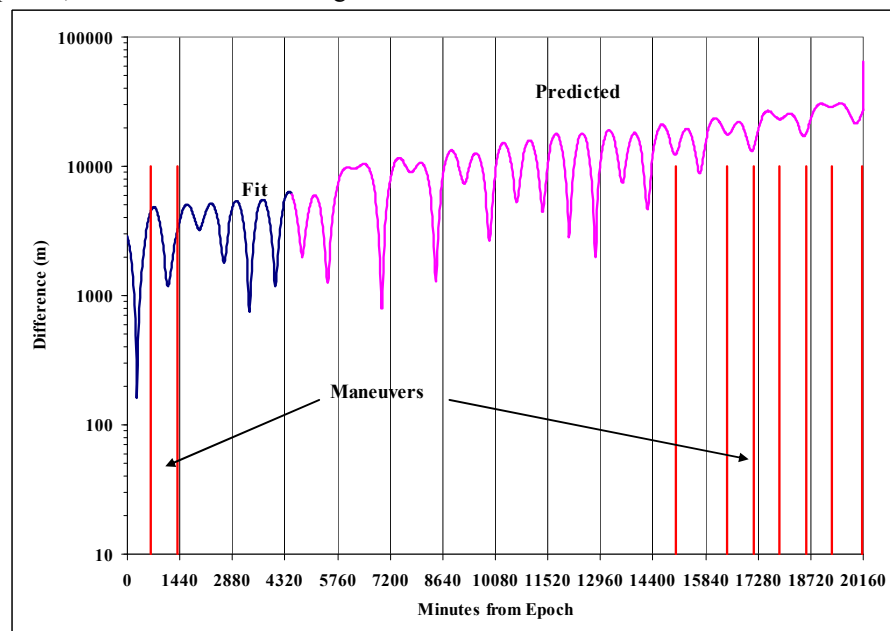

Figure 12. Intelsat 1002 Solution Performance. The differences of the converged TLE to the original ephemeris are shown. The difference is the RSS of the three positional components. The fit span was about 4500 minutes.

Finally, we examined 2 geosynchronous satellites that proved challenging to operate with various fit spans. These satellite ephemerides were simply generated with a numerical propagator. Figure 13 shows various fit spans for the first satellite. These ephemerides were in ECI (IAU-76/FK5 J2000) coordinates.
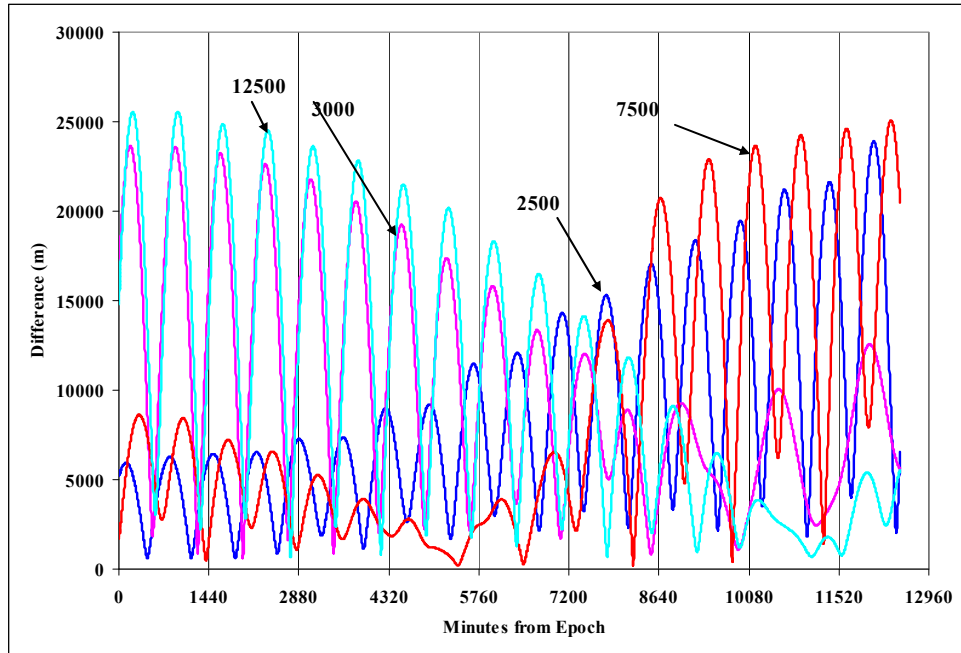
American Institute of Aeronautics and Astronautics

Figure 13. Sat1 Solution Performance. The differences of the converged TLE to the original ephemeris are shown. The difference is the RSS of the three positional components. Notice how the complete fit span yielded (not surprisingly) the best results.

Note that at short fit spans (2500 and 3000 min), the performance varied significantly in both the short and long term. Likewise for the longer fit spans, the results were not consistent. The second satellite produced similar, but slightly different results. In Fig 14, the fit span is about 3000 minutes.
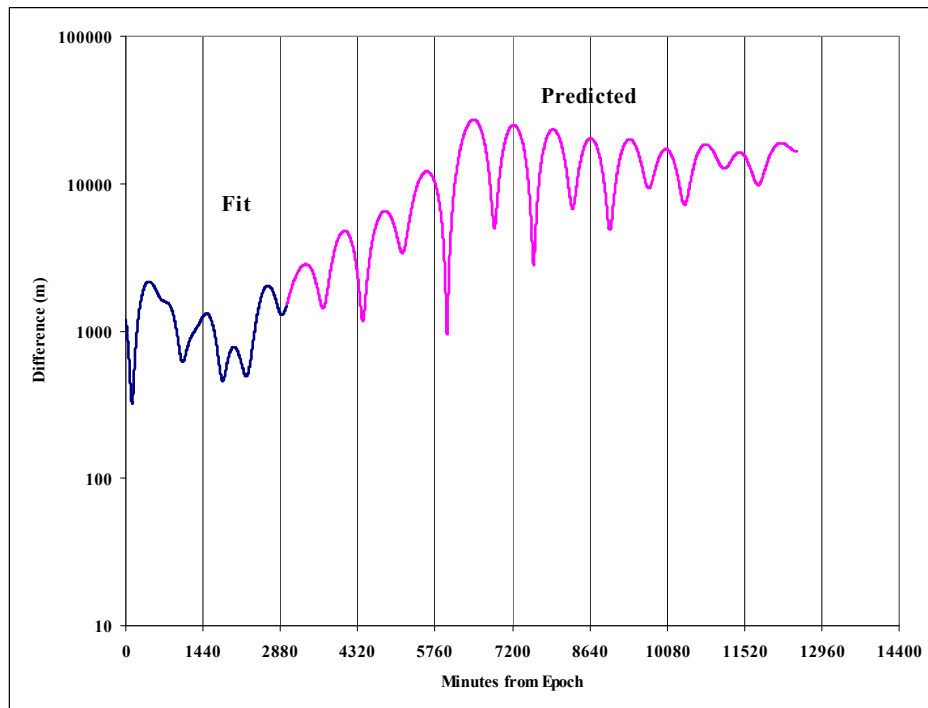

Figure 14. Sat2 Solution Performance. The differences of the converged TLE to the original ephemeris are shown. The difference is the RSS of the three positional components.

American Institute of Aeronautics and Astronautics

While these series of tests are not exhaustive, we feel they represent a good cross-section of potentially useful orbits for individuals to process and create TLE data from.

VIII.   OBSERVATIONAL DATA TESTING

We had originally planned to examine the of observational data from Vallado and Carter (1997), and Phillips Laboratory (1996), but time ran short and the results were not adequate for proper analysis. The code contains most of the necessary routines, but the convergence from the initial state vectors was not sufficient for adequate convergence. Thus, the most likely errors are in the initial orbit determination routine, and not the orbit determination section. If the fix is found quickly, we will simply put an updated version on the web, and if there were deeper problems, we will present the results in a follow-on paper. The ability to form a TLE from an ephemeris was deemed as an important goal for this version of the paper.

IX.   COVARIANCE CONSIDERATIONS

At the outset of this project, a goal was to examine the covariance matrix that resulted from solution in the SGP4DC process. Operationally, AFSPC does not produce and transmit a covariance with the TLE data. However, this could be of great benefit to operational programs like the SOCRATES and SOCRATES-GEO efforts (Kelso and Alfano 2005), currently in operation at the Center for Space Standards and Innovation.

There are several details to consider when looking at the covariance. First, if the orbital elements are equinoctial, the resulting covariance matrix will also be in equinoctial elements. From Vallado (2003), you could convert to Cartesian elements and then look at the three components of position uncertainty. See Vallado (20030 for the equations.

A simpler method is to convert the equinoctial orbital elements to classical orbital elements using the following matrix, multiplied by the covariance matrix in equinoctial elements.

$$
\begin{bmatrix}
 & a & e & i & \Omega & \omega & M \\
\frac{\partial}{\partial a_f} & & \frac{a_f}{\sqrt{a_f^2+a_g^2}} & 0 & 0 & \frac{-a_g}{a_f^2+a_g^2} & \frac{a_g}{a_f^2+a_g^2} \\
\frac{\partial}{\partial a_g} & 0 & \frac{a_g}{\sqrt{a_f^2+a_g^2}} & 0 & 0 & \frac{a_f}{a_f^2+a_g^2} & \frac{-a_f}{a_f^2+a_g^2} \\
\frac{\partial}{\partial L} & 0 & 0 & 0 & 0 & 0 & 1 \\
\frac{\partial}{\partial n} & -\frac{2}{3n}\left(\frac{\mu}{n^2}\right)^{1/3} & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial}{\partial \chi} & 0 & 0 & \frac{2\chi}{(1+\chi^2+\psi^2)\sqrt{\chi^2+\psi^2}} & \frac{\psi}{\chi^2+\psi^2} & \frac{-\psi}{\chi^2+\psi^2} & 0 \\
\frac{\partial}{\partial \psi} & 0 & 0 & \frac{2\psi}{(1+\chi^2+\psi^2)\sqrt{\chi^2+\psi^2}} & \frac{-\chi}{\chi^2+\psi^2} & \frac{\chi}{\chi^2+\psi^2} & 0
\end{bmatrix} (A^T W A)^{-1}
$$

(4)

Once in Classical elements, the 3 semimajor axis could be examined to determine the uncertainty.

a = part e * atwai(1,2) = part w * atwai(1,5) + part M * atwai(1,6)

However, best approach is simply to examine the semimajor axis directly from the equinoctial elements. We can examine just the semimajor axis component. In our code, this variable was the middle component, so in meters,

sqrt(atwai[2][2]) * 6378135.0

Results:

iter	norad	inti diff (m)	final diff	a	e	i	ap alt	pr alt	cov (m)
3	5	7521	0.000130	8632.35	0.185967	34.2682	3860	649	494.08
4	4632	36424	0.000001	37357.64	0.145051	11.4628	36399	25561	2385.90
7	6251	12793	0.000001	6776.12	0.003004	58.0579	418	378	313.73
4	8195	96231	0.001236	26566.17	0.687715	64.1586	38459	1918	2689.37
5	9880	21346	0.001015	26537.74	0.706905	64.5968	38920	1400	2960.19
3	9998	2674	0.000001	38219.79	0.027097	9.4958	32878	30807	2863.52
4	14128	10838	0.000001	42561.42	0.001156	11.4384	36233	36135	3287.91
3	16925	25485	0.000379	14671.55	0.559633	62.0906	16505	83	790.18
7	20413	4486546	0.000978	107327.52	0.786445	12.3514	185361	16543	32837.68
4	21897	27002	0.000105	26497.61	0.742169	62.1749	39786	454	2642.32
3	22312	13377	0.000001	6664.24	0.030872	62.1486	492	80	311.29
5	22674	75173	0.000540	26908.30	0.754171	63.5035	40825	237	2793.16
8	23177	160476	0.000468	24534.29	0.725849	7.0496	35965	348	3622.16
24	23333	29287790	47599701.12617	241621.01	0.972830	28.7490	470309	187	2642.59

American Institute of Aeronautics and Astronautics

4	23599	19706	0.000886	15547.92	0.578202	6.9327	18160	180	1208.44
3	24208	21039	0.000001	42022.52	0.002664	3.8536	35757	35533	3228.38
25	25954	16671	88.511236	42163.99	0.000177	0.0004	35794	35779	3239.51
20	26900	17207	0.113500	42163.27	0.000332	0.0164	35800	35772	3237.75
4	26975	38749	0.000001	26119.39	0.560288	68.4714	34376	5107	1600.94
19	28057	15097	0.091523	7151.47	0.000088	98.4283	774	773	326.46
3	28129	7776	0.000000	26559.87	0.004851	54.7298	20311	20053	1631.68
6	28350	13698	0.000001	6522.99	0.002487	64.9977	161	129	307.05
3	28623	22510	0.000001	17362.10	0.624905	28.5200	21834	134	446.29
25	28626	8916	349880.974300	42164.30	0.000034	0.0019	35788	35786	3239.64
3	28872	15442	0.000001	6527.85	0.030395	96.4736	348	-49	301.67
5	29141	15108	0.000001	6670.93	0.001585	82.4288	304	282	311.53
3	29238	11738	0.000001	6725.98	0.020258	51.5595	484	212	312.89
3	29238	11738	0.000001	251909.30	0.020258	0.8999	250640	240433	312.89
3	58	7357	0.000001	7464.91	0.016499	28.3286	1210	964	330.05
15	1317	19632	0.081589	42137.89	0.000110	14.3842	35765	35756	3236.25
4	836	581041	0.000000	109606.10	0.014510	37.9564	104821	101640	13525.23
4	6893	4572471	0.000001	223726.87	0.162482	50.5548	253706	181001	33348.86
5	9931	124135	0.000749	26954.03	0.660965	26.4846	38393	2760	2948.22
5	11416	15084	0.000001	7195.42	0.001136	98.7380	826	809	327.59
6	11801	30244	0.003589	24345.61	0.731804	46.7916	35785	151	3326.17
3	11871	8911	0.000000	26521.90	0.634478	67.5731	36972	3316	769.67
3	12032	53843	0.000001	26531.94	0.708622	65.2329	38956	1353	722.71
4	13446	32464	0.000208	26600.29	0.749888	62.1717	39225	140	1758.96
18	19483	3812	0.089475	42163.59	0.000223	7.9373	35796	35777	3239.52
20	19484	22905	0.079662	42162.97	0.000118	0.0718	35791	35781	3238.49
23	19508	11620	0.080769	42163.60	0.000242	0.0090	35797	35776	3238.46
5	19541	75735	0.000644	26532.61	0.730406	64.5823	39535	775	2544.05
12	19548	19300	0.065515	42165.58	0.000183	3.1655	35796	35781	3238.97
22	19596	17288	0.099627	42166.94	0.000065	5.5272	35792	35787	3239.25
3	19621	19577	0.000000	42467.44	0.000494	0.7802	36111	36069	3275.15
3	19683	10810	0.000001	42178.18	0.000559	5.4585	35825	35777	3238.80
3	19765	13292	0.000001	42504.85	0.001802	5.2955	36204	36051	3275.80
4	20338	33224	0.000001	26557.01	0.705085	64.7304	38905	1454	2578.53
3	20547	15329	0.000001	6939.45	0.012149	94.1277	646	477	314.78
3	22490	6631	0.000001	7134.31	0.004515	24.9657	789	724	324.36
3	23246	24427	0.000164	15628.61	0.566127	7.1648	18099	403	830.07
3	23581	25637	0.000001	42164.28	0.000574	1.1236	35811	35763	3237.27
2	23632	11278535	13792190.92840	103161.02	0.933270	63.8240	193064	506	5768.91
10	23866	98312	0.000973	22827.92	0.707615	4.5577	32604	297	3570.28
25	23949	21601	0.283295	42163.36	0.000043	0.0606	35788	35784	3238.92
4	24640	7929	0.000001	26550.63	0.733828	63.0016	39657	689	3718.24
18	24729	15092	0.085834	7789.69	0.000180	82.6021	1413	1410	348.40
3	24792	15241	0.000001	7149.69	0.000913	86.3890	778	765	326.39
5	24808	190253	0.001454	25312.34	0.697452	19.5851	36589	1280	4532.14
5	25068	782815	42092341.65759	40200.16	0.828658	7.5609	67136	510	351293.40
25	25237	18731	4175.166056	42163.57	0.000631	0.0192	35813	35760	3236.80
5	25989	349890	0.001211	66934.43	0.599671	54.6654	100697	20418	12753.65
3	26536	15066	0.000001	7230.28	0.001126	98.9441	860	844	329.20
11	27453	15074	0.051861	7192.00	0.001257	98.7410	823	805	328.25
24	28358	10855	0.093541	42163.27	0.000111	0.0130	35791	35781	3239.06
6	28893	15108	0.000000	7071.06	0.001627	98.1063	705	682	324.62
15	30582	2336029	0.034674	66714.35	0.869927	4.7533	118375	2300	16820.99
4	88888	13423	0.000000	6636.32	0.008673	72.8435	316	201	308.97
4	88888	13423	0.000001	248440.28	0.008673	1.2714	244222	239913	308.97
8	88888	724561	0.000001	248440.28	0.008673	1.2714	244222	239913	46622.60

There are several things to consider. Most importantly, we do not know the accuracy of the original TLE. Some TLE data propagates quite well into the future, while the next TLE for the same satellite can depart dramatically after only a day or less. Because there is no measure of accuracy with each TLE, it's impossible to determine if the fit is contributing to the error, or if the original TLE was faulty. This could be the case where some satellites converged, but had a high covariance.

In the cases where the solution diverged, examining the covariance is of no use and can be ignored. The remaining cases look "reasonable" although significant additional testing is needed to determine if these numbers are indeed representative of the actual accuracies, and could actually be useful for other applications.

For the cases in which we used ephemerides, we know the precision of the in put observations, so the covariance numbers could become more meaningful. The results were:

IceSat	0.615 m
GPS	3.35 m
Intelsat 704	6.85 m
Intelsat 805	6.85 m
Intelsat 906	6.85 m
Intelsat 1002	6.85 m

Unfortunately, these seem to be more a measure of the frequency of the data than the accuracy. This area probably needs additional analysis and consideration, including results from processing observational data.

X. AVAILABILITY

The current code is available in C++ as this appears to be the most common language for operations today. There are several debugging statements and paths that were left to facilitate experimentation with various solutions. All the necessary files are located on the Internet for convenience. They are available from the Center for Space website:[**]
http://www.centerforspace.com/downloads/

XI. CONCLUSIONS

This paper has presented an orbit determination approach using the *Revisiting Spacetrack Report Number 3* formulation of analytical propagation (Vallado et al. 2006). The current code supports two types of observations – ephemeris observations, and a test mode using TLE data, and also with most of the code inserted to process range, azimuth and elevation angles. We hope this form of documentation will motivate analyses to solve some of the challenging test cases identified in this document so a more complete and robust technique may be provided at a future time. Any questions, comments, additions, etc. may be addressed to David Vallado at dvallado@centerforspace.com.

References

Note that some of these references may be difficult to find. AFSPC should be able to provide all the necessary information.

Brouwer, D. 1959. Solution of the Problem of Artificial Satellite Theory without Drag. *Astronomical Journal*, Vol. 64, No. 1274, pp. 378–397.

David A. Cappellucci. 2005. "Special Perturbations to General Perturbations Extrapolation Differential Correction in Satellite Catalog Maintenance." Paper AAS 05-402 presented at the AIAA/AAS Astrodynamics Specialist Conference. Lake Tahoe, California.

Cefola, Paul J., and D. J. Fonte. 1996. Extension of the Naval Space Command Satellite Theory to include a General Tesseral m-daily Model. Paper AIAA-96-3606 presented at the AIAA/ AAS Astrodynamics Conference. San Diego, CA.

Hoots, Felix R., and R. L. Roehrich. 1980. "Spacetrack Report #3: Models for Propagation of the NORAD Element Sets." U.S. Air Force Aerospace Defense Command, Colorado Springs, CO.

Kelso, T. S., and S. Alfano. 2005. "Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES)." Paper AAS 05-124 presented at the AAS/AIAA Space Flight Mechanics Conference. Copper Mountain, CO.

Kelso, T. S. 2007. "Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides." Paper AAS 07-127 presented at the AAS/AIAA Spaceflight Mechanics Conference. Sedona, Arizona.

Kozai, Yoshihide. 1959. The Motion of a Close Earth Satellite. Astronomical Journal. 64(1274):367-377.

Marshall, P. M. 1999. Least Squares Solutions in Statistical Orbit Determination Using Singular Value Decomposition. MS Thesis. Naval Postgraduate School. Monterey, CA.

Press, William H. et al. 1992. Numerical Recipes in Fortran. Cambridge, England: Cambridge University Press.

Vallado, David A. 2007 *Fundamentals of Astrodynamics and Applications*. Third edition. Springer/Microcosm, Hawthorne, CA.

Vallado, David A. 2005. "An Analysis of State Vector Propagation using Differing Flight Dynamics Programs." Paper AAS 05-199 presented at the AAS/AIAA Space Flight Mechanics Conference. Copper Mountain, CO.

Vallado, David A. 2003. Covariance Transformations for Satellite Flight Dynamics Operations. Paper AAS-03-526 presented at the AAS/AIAA Astrodynamics Specialist Conference. Big Sky, MT.

[**] Users of Analytical Graphics Inc. Satellite Toolkit (STK) will find the SGP4 source code integrated within the latest release of the program. The orbit determination code is not yet implemented in STK as there is another version of the conversion operation currently in use.

Appendix A – Observational Data Format

Data formats for observations tend to be quite complicated. Most were developed many years ago when computer transmission was difficult – both in terms of speed and volume. Today, those limitations are gone, but the formats remain. I have designed a simple system to transmit observations of different types. There are two key elements to realize. First, the format is space delimited to permit easy parsing in any computer language. Too often, the data is packed into a no-space format that must be painstakingly extracted before use. Second, the data type controls how many variables follow. Thus, radar, optical, and other observations may all coexist in a single file. Because of the spacing feature, the number of decimal places and the options for various observations formats are essentially unlimited.

```
# Tracking obs format
# Ver 1.0 dav 30 Jul 08
#
# this is intended to be a simple obs format
# the data is all space delimited so variable sizing is possible
# comment lines are preceded by the # symbol in the first col
# units are km and km/s and deg
# obs are assumed to be earth fixed topocentric
# coordinate system is ITRF, ECI (Specify GCRF or J2000), or TEME
#
# data types control how much information follows:
# 0 range only
# 1 azimuth and elevation
# 2 range, azimuth and elevation
# 3 topocentric right ascension and declination
# 4 ephemeris values (xyz, xyz dot)
# 5
# 6
# 7
# 8
# 9
# 10
#
#Typ   Sat #   Sen # Yr   M D  hr min sec         value      value      value
  2    7734     932 1995  1 29 02 38 37.000    2047.50200   60.4991    16.1932
  2    7734     932 1995  1 29 02 38 49.000    1984.67700   62.1435    17.2761
  2    7734     932 1995  1 29 02 39 02.000    1918.48900   64.0566    18.5515
  2    7734     932 1995  1 29 02 39 14.000    1859.32000   65.8882    19.7261
  2    7734     932 1995  1 29 02 39 26.000    1802.18600   67.9320    20.9351
  2    7734     932 1995  1 29 02 39 38.000    1747.29000   70.1187    22.1319
  2    7734     932 1995  1 29 02 39 50.000    1694.89100   72.5159    23.3891
  2    7734     932 1995  1 29 02 40 03.000    1641.20100   75.3066    24.7484
  2    7734     932 1995  1 29 02 40 15.000    1594.77000   78.1000    25.9799
  2    7734     932 1995  1 29 02 40 27.000    1551.64000   81.1197    27.1896
  2    7734     932 1995  1 29 02 40 39.000    1512.08500   84.3708    28.3560
  2    7734     932 1995  1 29 02 40 51.000    1476.41500   87.8618    29.4884
  2    7734     932 1995  1 29 02 41 03.000    1444.91500   91.5955    30.5167
  2    7734     932 1995  1 29 02 41 15.000    1417.88000   95.5524    31.4474
  2    7734     932 1995  1 29 02 41 27.000    1395.56300   99.7329    32.2425
  2    7734     932 1995  1 29 02 41 39.000    1378.20200  104.0882    32.8791
  2    7734     932 1995  1 29 02 41 51.000    1366.01000  108.6635    33.3788
  2    7734     932 1995  1 29 02 42 03.000    1359.10000  113.2254    33.5998
  2    7734     932 1995  1 29 02 42 15.000    1357.57000  117.8856    33.6487
  2    7734     932 1995  1 29 02 42 27.000    1361.44200  122.5305    33.4924
  2    7734     932 1995  1 29 02 42 39.000    1370.66200  127.0982    33.1160
  2    7734     932 1995  1 29 02 42 51.000    1385.13000  131.5544    32.5864
  2    7734     932 1995  1 29 02 43 03.000    1404.68400  135.8363    31.8581
  2    7734     932 1995  1 29 02 43 15.000    1429.10700  139.9144    30.9931
  2    7734     932 1995  1 29 02 43 27.000    1458.16000  143.7770    30.0041
  2    7734     932 1995  1 29 02 43 39.000    1491.56100  147.3908    28.9206
```

Appendix B – Program interface Issues

In the 2006 paper, we discussed several program interface issues. Most of these are discussed in the paper, but a few require additional discussion for the orbit determination process.

A. Coordinate System

The True Equator Mean Equinox (TEME) coordinate system is assumed for all SGP4 operations[††]. If observations are used, they are implicitly in an Earth-Fixed (International Terrestrial Reference Frame, ITRF) frame, and thus require standard reduction techniques to move to TEME for used in the formation of the TLE data. Likewise, external ephemerides are generally in ITRF or IAU-76/FK5 and require similar reduction. I transform all vectors to the ITRF, and then to TEME through polar motion and Greenwich Mean Sidereal Time. TEME is found by observing that θ_{GAST82} may be separated into its components. Thus,

$$\vec{r}_{TOD} = ROT3(-\theta_{GAST82})\vec{r}_{PEF} \quad and \quad \theta_{GAST82} = \theta_{GMST82} + Eqe_{82}$$
$$\vec{r}_{TEME} = ROT3(-\theta_{GMST82})\vec{r}_{PEF}$$
$$\vec{r}_{PEF} = ROT3(\theta_{GMST82})\vec{r}_{TEME} \tag{5}$$

We recommend converting TEME to a truly standard coordinate frame before interfacing with other external programs. The preferred approach is to rotate to PEF using Greenwich Mean Sidereal Time (GMST), and then rotate to other standard coordinate frames. Conversions are well documented from this point. To implement, you simply apply a sidereal rotation about the Z-axis by GMST (using UT1 as we discuss later). Because polar motion has been historically neglected for General Perturbation (GP) applications, we assume that the pseudo Earth-fixed frame is the closest conventional frame.[‡‡] The coordinate system is important when processing observations, and external ephemerides as the data must first be converted to the TEME coordinate system.
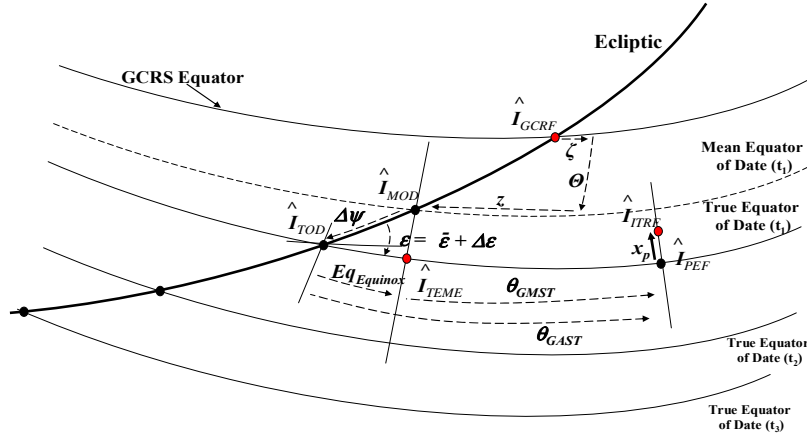

Figure 1. TEME Coordinate System. This coordinate system is between TOD and PEF and differs from PEF by the mean Greenwich Sidereal Time.

B. Mean Motion and Mean Motion Rate

We do not include estimates for mean motion rate and acceleration as they are not needed by the SGP4 routine. These values were included in the original SGP4 program to support users still using the SGP program. The terms essentially modeled the drag effect on the satellite. The BStar term accomplishes this in SGP4, so the additional

---

[††] We note that AFSPC has never published a formal definition of TEME, although we assume the definition here is consistent with their approach. Further, the AFSPC has likely changed its definition over the years, but again, this is uncertain due to a lack of official publication.

[‡‡] We assume that CMOC orbit determination approximates the reference frames of radar and optical differently, and that numerical and analytical orbit determination methods use different techniques due to the differences in TEME, ECI, and the uncertain use of polar motion in coordinate systems.

burden of calculating parameters that would not be used is omitted. The TLE format is shown below. The program outputs this format directly for ease of use with existing applications. Some of the fields are not calculated, and are therefore left blank. the classification field is set to "I" to indicate the improved formulation.

Card #	Satellite Number	Class	International Designator			Yr	Epoch Day of Year (plus fraction)		Mean motion derivative (rev/day /2)		Mean motion second derivative (rev/day2 /6)		Bstar (/ER)			Eph	Elem num	Chk Sum
			Year	Lch#	Piece				S		S.		S.		S E			
1	1 6 6 0 9	U	8 6 0 1 7		A	9 3	3 5 2 . 5 3 5 0 2 9 3 4		. 0 0 0 0 7 8 8 9		0 0 0 0 0 - 0		1 0 5 2 9 - 3			0	3 4	2

Card #	Satellite Number		Inclination (deg)		Right Ascension of the Node (deg)		Eccentricity		Arg of Perigee (deg)		Mean Anomaly (deg)		Mean Motion (rev/day)		Epoch Rev	Chk
2	1 6 6 0 9		5 1 . 6 1 9 0		1 3 . 3 3 4 0		0 0 0 5 7 7 0		1 0 2 . 5 6 8 0		2 5 7 . 5 9 5 0		1 5 . 5 9 1 1 4 0 7 0 4 4 7 8 6			9

Figure 2. Two-line Element Set Format. An example TLE is shown, with descriptions and units of each field. Note that the eccentricity, mean motion second derivative, and Bstar have implied decimal points before the first numerical value. The mean motion derivative is already divided by 2, and the second derivative is already divided by 6. Shaded cells do not contain data. The signs may be blank, "+" or "–". The classification field is sometimes included after the satellite number, and for the case of generated orbits via this program, it is set to "I" to signify the improved nature of the elements.

The TLE use 'mean motion' in terms of Kozai's theory (1959), but internally the SGP4 model is based on Brouwer's work (1959). Normally the TLE data are converted at the initialization of SGP4 to match, but it leads to a total of 3 possible interpretations of mean motion and semimajor axis that can be confusing.

- Simple single body orbit and the Earth's GM constant.

- Brower mean elements in SGP4.

- Kozai mean formulation for TLE.

For the simplest case we have the conversion from semi-major axis a to mean motion n from:

$$n = \sqrt{\frac{\mu}{a^3}} \qquad a = \sqrt[3]{\frac{\mu}{n^2}}$$

(6)

where μ GM is the Earth's gravitational constant. But note that SGP4 was developed using a WGS72 value (398600.8 km^{3}s^{-2}), while it is more common today to use an EGM-96 value of 398600.4415 km^{3}s^{-2}. Whatever constant is used, you must always be *consistent* with all definitions so that conversion forward and backwards leads to the same value for maximum accuracy!

If you are converting from a state vector and/or from an ephemeris generated form another model, you first need to convert the semimajor axis to a suitable mean motion for SGP4. The iterated single point conversion of osculating to mean elements will do this, but it also helps if you can convert using the inverse of the TLE to Brower mean equations.

The TLE conversion for SGP4 is given in (Hoots and Roehrich, 1980:10) with the following steps:

$$a_1 = \left(\frac{XKE}{n_0}\right)^{\frac{2}{3}}$$

$$k_2 = \frac{1}{2} J_2 a_E^2 \qquad K = \left(\frac{3}{2} k_2 \frac{3\cos^2(i_0) - 1}{(1 - e_0^2)^{\frac{3}{2}}}\right)$$

$$\delta_1 = \frac{K}{a_1^2}$$

(7)

American Institute of Aeronautics and Astronautics

$$a_0 = a_1\left(1 - \frac{1}{3}\delta_1 - \delta_1^2 - \frac{134}{81}\delta_1^3\right)$$

$$\delta_0 = \frac{K}{a_0^2}$$

$$a_0'' = \frac{a_0}{1 - \delta_0} \qquad n_0'' = \frac{n_0}{1 + \delta_0}$$

where XKE is the gravity constant of 7.43669161331734132e-2 Earth radii^3 min^{-2} used by SGP4 and the semi-major axis is in Earth radii, based on the conversion factor from WGS72 of XKMPER = 6378.135 km. Again, what matters for accuracy is to use the *same assumptions* as the SGP4 code uses. The two corrections are made and can be iterated, but usually are not. The constant K is based on the initial (TLE) values of inclination i_0, eccentricity e_0, and the J_2 gravity and Earth radius term k_2 and need not be updated each time in an iterated solution. Basically we have a function defined in Eq-3 representing:

$$a_0'' = f(n_0) \tag{8}$$

However, sometimes we want to invert this to allow us to compute a TLE value of the mean motion from a given Brower mean semi-major axis a_0''. This is best implemented by Newton-Raphson root solving, first by defining the function to zero as:

$$g(n_0) = a_0'' - f(n_0) = 0 \tag{9}$$

We can start with a good initial guess for the mean motion from our value of semi-major axis (converted to units of Earth radii using XKMPER):

$$n_0 = \frac{XKE}{a^{\frac{2}{3}}}$$

Iterate

$$n_{N+1} = n_N - \frac{g(n_N)}{g'(n_N)} \tag{10}$$

$$g'(n_0) \approx \frac{2}{3} \cdot \frac{a_0}{n_0}$$

Normally it only requires 2-4 iterations of the NR process to converge to double precision accuracy. If the last correction to n_0 is less than around 10^{-12} relative value, the loop can be terminated as the remaining error is typically at the machine precision.

Appendix C – Test Case Listing

We combined several sources of test data for this file (twoline.txt) which is intended to model many of the routine and difficult cases for orbit determination.

```
#    ----------------- Verification test cases ---------------------
#                      # TEME example
1 00005U 58002B   00179.78495062  .00000023  00000-0  28098-4 0  4753
2 00005  34.2682 348.7242 1859667 331.7664  19.3264 10.82419157413667
#                      ## fig show lyddane fix error with gsfc ver
1 04632U 70093B   04031.91070959 -.00000084  00000-0  10000-3 0  9955
2 04632  11.4628 273.1101 1450506 207.6000 143.9350  1.20231981 44145
#   DELTA 1 DEB        # near earth normal drag equation
#                      # perigee = 377.26km, so moderate drag case
1 06251U 62025E   06176.82412014  .00008885  00000-0  12808-3 0  3985
2 06251  58.0579  54.0425 0030035 139.1568 221.1854 15.56387291  6774
#   MOLNIYA 2-14       # 12h resonant ecc in 0.65 to 0.7 range
1 08195U 75081A   06176.33215444  .00000099  00000-0  11873-3 0   813
2 08195  64.1586 279.0717 6877146 264.7651  20.2257  2.00491383225656
#   MOLNIYA 1-36       ## fig 12h resonant ecc in 0.7 to 0.715 range
1 09880U 77021A   06176.56157475  .00000421  00000-0  10000-3 0  9814
2 09880  64.5968 349.3786 7069051 270.0024  16.3320  2.00813614112380
#   SMS 1 AKM          # show the integrator problem with gsfc ver
1 09998U 74033F   05148.79417928 -.00000112  00000-0  00000+0 0  4480
2 09998   9.4958 313.1750 0270971 327.5225  30.8097  1.16186785 45878
#   EUTELSAT 1-F1 (ECS1)## fig lyddane choice in GSFC at 2080 min
1 14128U 83058A   06176.02844893 -.00000158  00000-0  10000-3 0  9627
2 14128  11.4384  35.2134 0011562  26.4582 333.5652  0.98870114 46093
#   SL-6 R/B(2)        # Deep space, perigee = 82.48 (<98) for
#                      # s4 > 20 mod
1 16925U 86065D   06151.67415771  .02550794 -30915-6  18784-3 0  4486
2 16925  62.0906 295.0239 5596327 245.1593  47.9690  4.88511875148616
#   SL-12 R/B          # Shows Lyddane choice at 1860 and 4700 min
1 20413U 83020D   05363.79166667  .00000000  00000-0  00000+0 0  7041
2 20413  12.3514 187.4253 7864447 196.3027 356.5478  0.24690082  7978
#   MOLNIYA 1-83       # 12h resonant, ecc > 0.715 (negative BSTAR)
1 21897U 92011A   06176.02341244 -.00001273  00000-0 -13525-3 0  3044
2 21897  62.1749 198.0096 7421690 253.0462  20.1561  2.01269994104880
#   SL-6 R/B(2)        # last tle given, decayed 2006-04-04, day 94
1 22312U 93002D   06094.46235912  .99999999  81888-5  49949-3 0  3953
2 22312  62.1486  77.4698 0308723 267.9229  88.7392 15.95744531 98783
#   SL-6 R/B(2)        # 12h resonant ecc in the > 0.715 range
1 22674U 93035D   06176.55909107  .00002121  00000-0  29868-3 0  6569
2 22674  63.5035 354.4452 7541712 253.3264  18.7754  1.96679808 93877
#   ARIANE 44L+ R/B    # Lyddane bug at <= 70 min for atan2(),
#                      # no quadrant fix
1 23177U 94040C   06175.45752052  .00000386  00000-0  76590-3 0    95
2 23177   7.0496 179.8238 7258491 296.0482   8.3061  2.25906668 97438
#   WIND               # STR#3 Kepler failes past about 200 min
1 23333U 94071A   94305.49999999 -.00172956  26967-3  10000-3 0    15
2 23333  28.7490   2.3720 9728298  30.4360   1.3500  0.07309491    70
#   ARIANE 42P+3 R/B   ## fig Lyddane bug at > 280.5 min for AcTan()
1 23599U 95029B   06171.76535463  .00085586  12891-6  12956-2 0  2905
2 23599   6.9327   0.2849 5782022 274.4436  25.2425  4.47796565123555
#   ITALSAT 2          # 24h resonant GEO, inclination > 3 deg
1 24208U 96044A   06177.04061740 -.00000094  00000-0  10000-3 0  1600
2 24208   3.8536  80.0121 0026640 311.0977  48.3000  1.00778054 36119
#   AMC-4              ## fig low incl, show incl shift with
#                      ## gsfc version from 240 to 1440 min
1 25954U 99060A   04039.68057285 -.00000108  00000-0  00000-0 0  6847
2 25954   0.0004 243.8136 0001765  15.5294  22.7134  1.00271289 15615
#   INTELSAT 902       # negative incl at 9313 min then
#                      # 270 deg Lyddane bug at 37606 min
1 26900U 01039A   06106.74503247  .00000045  00000-0  10000-3 0  8290
2 26900   0.0164 266.5378 0003319  86.1794 182.2590  1.00273847 16981
#   COSMOS 1024 DEB    # 12h resonant ecc in 0.5 to 0.65 range
1 26975U 78066F   06174.85818871  .00000620  00000-0  10000-3 0  6809
2 26975  68.4714 236.1303 5602877 123.7484 302.5767  2.05657553 67521
#   CBERS 2                # Near Earth, ecc = 8.84E-5 (< 1.0e-4)
#                      # drop certain normal drag terms
1 28057U 03049A   06177.78615833  .00000060  00000-0  35940-4 0  1836
2 28057  98.4283 247.6961 0000884  88.1964 271.9322 14.35478080140550
#   NAVSTAR 53 (USA 175)# 12h non-resonant GPS (ecc < 0.5 ecc)
```

```
1 28129U 03058A   06175.57071136 -.00000104  00000-0  10000-3 0   459
2 28129  54.7298 324.8098 0048506 266.2640  93.1663  2.00562768 18443
#   COSMOS 2405        # Near Earth, perigee = 127.20 (< 156) s4 mod
1 28350U 04020A   06167.21788666  .16154492  76267-5  18678-3 0  8894
2 28350  64.9977 345.6130 0024870 260.7578  99.9590 16.47856722116490
#   H-2 R/B            # Deep space, perigee = 135.75 (<156) s4 mod
1 28623U 05006B   06177.81079184  .00637644  69054-6  96390-3 0  6000
2 28623  28.5200 114.9834 6249053 170.2550 212.8965  3.79477162 12753
#   XM-3               # 24h resonant geo, incl < 3 deg goes
#                      # negative around 1130 min
1 28626U 05008A   06176.46683397 -.00000205  00000-0  10000-3 0  2190
2 28626   0.0019 286.9433 0000335  13.7918  55.6504  1.00270176  4891
#   MINOTAUR R/B       # Sub-orbital case - Decayed 2005-11-29
#                      #(perigee = -51km), lost in 50 minutes
1 28872U 05037B   05333.02012661  .25992681  00000-0  24476-3 0  1534
2 28872  96.4736 157.9986 0303955 244.0492 110.6523 16.46015938 10708
#   SL-14 DEB          # Last stage of decay - lost in under 420 min
1 29141U 85108AA  06170.26783845  .99999999  00000-0  13519-0 0   718
2 29141  82.4288 273.4882 0015848 277.2124  83.9133 15.93343074  6828
#   SL-12 DEB          # Near Earth, perigee = 212.24 < 220
#                      # simplified drag eq
1 29238U 06022G   06177.28732010  .00766286  10823-4  13334-2 0   101
2 29238  51.5595 213.7903 0202579  95.2503 267.9010 15.73823839  1061
#COURIER 1B
1 00058U 60013A   97142.85906518  .00000093  00000-0 +10762-4 0  274
2 00058 028.3286 356.4726 0164991 158.6392 202.1128 13.4602145880282
#INTELSAT 1-F1
1 01317U 65028A   96319.00000000 -.00000073  00000-0 +10000-3 0 0497
2 01317  14.3842 342.1674 0001104 344.6118 333.1881  1.0036445301882
#OPS 3662 (VELA 3)
1 00836U 64040  A 68333.71805548  .00000000  00000-0 +00000-0 0 00218
2 00836  37.9564 182.1088 0145101 215.8663 266.6045  0.23924182003816
#EXPLORER 50 (IMP-8)
1 06893U 73078A   04097.00000000 -.00000307  00000-0 +00000-0 0 0009
2 06893  50.5548  69.6045 1624818 111.1032 181.3477  0.0820374100863
#ESA-GEOS 1
1 09931U 77029A   01309.17453186 -.00000329  00000-0 +10000-3 0 05967
2 09931  26.4846 264.1300 6609654  82.2734 342.9061  1.96179522175451
#NOAA-06 (originally had convergence problem on single state vector)
1 11416U         79318.07850216  .00002502           11328-2 0   681
2 11416  98.7380 345.8233 0011358 235.3731 124.6417 14.22344451 19915
#Original report SGP4 test
1 11801U         80230.29629788  .01431103  00000-0  14311-1     13
2 11801  46.7916 230.4354 7318036  47.4722  10.4117  2.28537848     13
#Cosmos 1191
1 11871U 80057A   01309.36911127 -.00000499  00000-0 +10000-3 0 08380
2 11871  67.5731   1.8936 6344778 181.9632 173.2224  2.00993562062886
#Cosmos 1217
1 12032U 80085A   01309.42683181  .00000182  00000-0  10000-3 0  3499
2 12032  65.2329  86.7607 7086222 172.0967 212.4632  2.00879501101699
#Molniya 3-19Rk
1 13446U 82083E   01283.10818257  .00098407  45745-7  54864-3 0  6240
2 13446  62.1717  83.8458 7498877 273.9677 320.2568  2.06357523137203
#GSTAR 3
1 19483U 88081A   97137.45030508 -.00000097  00000-0 +10000-3 0 0521
2 19483   7.9373  57.7037 0002226 205.5811  28.9554  1.0027272402809
#SBS 5
1 19484U 88081B   97142.34600701  .00000026  00000-0 +10000-3 0 0582
2 19484   0.0718  90.9764 0001184 309.3037 201.3786  1.0027491902338
#CS 3B
1 19508U 88086A   97136.58827615 -.00000172  00000-0 +10000-3 0 0150
2 19508   0.0090 342.0093 0002420 132.0143 125.8497  1.0027267703182
#MOLNIYA 3-33
1 19541U 88090A   97140.93749050  .00001389  00000-0 +10000-3 0 0348
2 19541  64.5823  50.3507 7304063 252.7222  21.5082  2.0087196206334
#TDRS 3
1 19548U 88091B   97141.69944100 -.00000180  00000-0 +10000-3 0 0288
2 19548   3.1655  70.0997 0001830  39.5613 106.2873  1.0026561301892
#RADUGA 22
1 19596U 88095A   97141.90851079  .00000073  00000-0 +00000-0 0 0006
2 19596   5.5272  57.1219 0000648 263.9197 257.3937  1.0026075801522
#TDF 1
1 19621U 88098A   97140.87069484 -.00000215  00000-0 +10000-3 0 0047
2 19621   0.7802  81.7901 0004939 302.6242 335.6246  0.9919847602843
#EKRAN 19
1 19683U 88108A   97125.35657144 -.00000028  00000-0 +00000-0 0 0848
```

```
2 19683   5.4585  59.1499 0005594 156.1624 203.7351  1.0022068203075
#GORIZONT 17
1 19765U 89004A   97131.62338924 -.00000257  00000-0 +00000-0 0 0013
2 19765   5.2955  57.7160 0018016 248.1114 111.5186  0.9906756003028
#MOLNIYA 3-36
1 20338U 89094A   97142.09744901  .00000391  00000-0 +00000-0 0 0025
2 20338  64.7304 159.0283 7050852 257.7144 021.7192  2.0059514805480
#USA 55
1 20547U 90028B   97143.01993828 -.00000107  00000-0 -57012-5 0 0595
2 20547  94.1277 126.6788 0121488 214.5866 144.7378 15.0176434139040
#SCD 1
1 22490U 93009B   97142.96910273  .00000385  00000-0 +62296-4 0 0573
2 22490  24.9657 339.7066 0045151 174.3619 185.7429 14.4065988222567
#Ariane Debris
1 23246U 91015G   01311.70347086  .00004957  00000-0  43218-2 0  8190
2 23246   7.1648 263.6949 5661268 241.8299  50.5793  4.44333001129208
#GOES 9
1 23581U 95025A   01311.43599209 -.00000094  00000-0  00000+0 0  8214
2 23581   1.1236  93.7945 0005741 214.4722 151.5103  1.00270260 23672
#PROGNOZ M2
1 23632U 95039A   95215.05103721  .00000252  00000-0  10000-4 0    11
2 23632  63.8240 259.6684 9332704 314.5662   1.9243  0.26200872    02
#ARIANE 44L+3 R/B
1 23866U 96030C   08064.47394878 +.00001642  00000-0 +10591-2 0 0481
2 23866   4.5577 359.7906 7076151 309.1284   7.0534  2.5170366410325
#TURKSAT 1C
1 23949U 96040B   08064.69560088  .00000144  00000-0  10000-3 0  785
2 23949   0.0606 303.8447 0000431 150.1243   1.7326  1.00273524 4269
#MOLNIYA 3-48
1 24640U 96060A   97142.99892724  .00000137  00000-0 -34143-2 0 0172
2 24640  63.0016 126.1251 7338280 288.3170   9.5717  2.0066749200422
#GONETS D1 5
1 24729U 97006E   97143.07478940  .00000011  00000-0 +00000-0 0 0046
2 24729  82.6021 165.6659 0001797 176.5724 183.5388 12.6272439301232
#IRIDIUM 8
1 24792U 97020A   97143.07862317 -.02618709  00000-0 -98679-0 0 0034
2 24792  86.3890 141.4411 0009126  89.7444 270.6053 14.3601161200256
#THOR 2A
1 24808U 97025A   97142.38550799 -.00000113  00000-0 +00000-0 0 0007
2 24808  19.5851  31.8603 6974523 359.4112   0.2569  2.1557125200003
#EQUATOR S
1 25068U 97075B   08063.44758622  .00000013  00000-0  17097-2 0  831
2 25068   7.5609 312.0304 8286575 329.2092   2.0830  1.07707867 4044
#TL BIRD 4 (HOT BIRD 4
1 25237U 98013A   08065.07893157 -.00000060  00000-0  10000-3 0  778
2 25237   0.0192 266.3179 0006306  97.2228 180.8628  1.00272785 3700
#XMM
1 25989U 99066A   07119.20833333  .00000327  00000-0 +00000-0 0 03385
2 25989  54.6654 111.3750 5996711 125.4529 353.4507  0.50131974002323
#NOAA-16
1 26536U 00055A   04021.31559085  .00000045  00000-0  48966-4 0  7146
2 26536  98.9441 332.9210 0011258  75.8813 284.3609 14.12070705171679
#NOAA-17
1 27453U 02032A   03159.44074193  .00000223  00000-0  11792-3 0  4397
2 27453  98.7410 229.9790 0012568  43.3290 316.8875 14.23359968 49598
#INTELSAT 1002
1 28358U 04022A   08151.13080160  .00000004  00000-0  10000-3 0  5824
2 28358   0.0130 357.0652 0001105   8.7095 288.3925  1.00273874 14527
#SINAH 1
1 28893U 05043D   08064.84183463  .00000001  00000-0  97820-5 0  640
2 28893  98.1063 321.4268 0016274  19.4105 340.7706 14.60031533 12542
#THEMIS C
1 30582U 07004C   08065.75000000 -.00000721  00000-0  00000+0 0  212
2 30582   4.7533 308.4897 8699271   6.9855 355.4695  0.50380244  256
#Original report SGP4 test
1 88888U         80275.98708465  .00073094  13844-3  66816-4 0    87
2 88888  72.8435 115.9689 0086731  52.6988 110.5714 16.05824518  1058
```