Verification of the Orekit Java Implementation of the Draper Semi-Analytical Satellite Theory

Paul J. Cefola¹, Barry Bentley², Luc Maisonobe³, Pascal Parraud³, Romain Di-Costanzo³ and Zachary Folcik⁴

¹Department of Mechanical and Aerospace Engineering
State University of New York at Buffalo
Amherst, NY, USA

²University of Cambridge
Cambridge, UK

³CS Communications & Systemes
31506 Toulouse Cedex 5, France

⁴Arlington, MA, USA

AAS/AIAA Space Flight Mechanics Meeting
Lihue, Kauai, Hawaii
10-14 February 2013

AAS Publications Office, P.O. Box 28130, San Diego, CA 92198
VERIFICATION OF THE OREKIT JAVA IMPLEMENTATION OF THE
DRAPER SEMI-ANALYTICAL SATELLITE THEORY

Paul J. Cefola, * Barry Bentley, † Luc Maisonobe, ‡ Pascal Parraud, § Romain Di-Costanzo**, and Zachary Folcik††

Verification of the java Orekit implementation of the Draper Semi-analytical Satellite Theory (DSST) is discussed. The Orekit library for space flight dynamics has been published under the open-source Apache license V2. The DSST is unique among analytical and semi-analytical satellite theories due to the scope of the included force models. However, the DSST has not been readily accessible to the wider Astrodynamics research community. Implementation of the DSST in the Orekit library is a comprehensive task because it involves the migration of the DSST to the object-oriented java language and to a different functional decomposition strategy. The resolution of the code and documentation anomalies discovered during the verification process is the important product of this project.

INTRODUCTION

Orbit propagator packages employ satellite theory models to generate a trajectory over a finite interval of time given the initial conditions (usually Cartesian coordinates or orbital element sets) at a specific epoch and with knowledge of the space environment. The finite time interval may range from a few minutes to 100 or 200 years. Three general paradigms have evolved for solving artificial Earth satellite orbit dynamics problems:

---

* Adjunct Professor, Department of Mechanical and Aerospace Engineering, State University of New York at Buffalo, Amherst, NY, USA, paulcefo@buffalo.edu; also Consultant in Aerospace Systems, Spaceflight Mechanics, and Astrodynamics, Vineyard Haven, MA, USA.
† Graduate Student, University of Cambridge, Cambridge, UK, bntlyb@gmail.com
‡‡ CS Communications & Systemes, 5 rue Brindejonc des Moulinais, 31506 Toulouse Cedex 5 (France), Email: luc.maisonobe@c-s.fr
§ CS Communications & Systemes, 5 rue Brindejonc des Moulinais, 31506 Toulouse Cedex 5 (France), Email: pascal.parraud@c-s.fr
** CS Communications & Systemes, 5 rue Brindejonc des Moulinais, 31506 Toulouse Cedex 5 (France), Email: romain.di-costanzo@c-s.fr
†† 53 Maynard Street, Arlington, MA, USA, zjfolcik@mit.edu; Zachary Folcik is currently Technical Staff at the MIT Lincoln Laboratory.
1. Analytical satellite theories

2. Semianalytical Methods

3. Numerical Integration

Semi-analytical satellite theories are based on a synergistic combination of techniques from analytical satellite theory and from traditional numerical integration. These Semianalytical theories have several advantages:

1. Inclusion of comprehensive force models as in the DSST.

2. The capability to tailor the semi-analytical theory force models at execution time and to update the tailoring during the execution. Tailoring includes features such as enabling or disabling specific force models as well as setting the number and type of terms included in the model.

3. The spherical harmonic expansions associated with the geopotential have been passed through the perturbation transformation so the equations of motion and the short periodic expressions are given in terms of mean elements.

4. The theory includes the non-conservative perturbations of atmosphere drag and solar radiation pressure. The averaging operations for the non-conservative perturbations employ numerical quadrature. Thrust models are also included in DSST.

5. Several orbit estimation algorithms have been built to estimate the mean elements directly from the tracking data.

This paper discusses the validation and verification of the java Orekit implementation of the Draper Semi-analytical Satellite Theory (DSST). Orekit is a library for space flight dynamics. Orekit started in 2002 as a small in-house product developed by CS Systèmes d’Information, Toulouse, France. The ORbits Extrapolation KIT (Orekit) was intended to be a fundamental asset for CS in addition to serving as a basis for custom systems developed for customers [1]. The design goals were to write a tool that is easy to adapt, up to date with respect to recent space flight dynamics models, and still compatible with older models. Over several years the library matured from a small set of core components to a full-fledged collection of core classes and associated algorithms: orbits, time, reference frames, bodies, propagation, attitude, etc. Orekit was first published under the terms of the Apache license V2 in July 2008. The Apache license is a ‘permissive’ license – the Orekit source code is published as open-source but distribution of source code for derived works is not mandated [2]. In early 2011, a public source-forge site was established for Orekit. The Orekit forge provides public access to activity, bug reports, source code repository, documentation, and downloads [3]. Open governance was established for Orekit in 2012 using a meritocratic model [3]. The most prominent application of the meritocratic model is the Apache Software Foundation [4] (Linux also has a very prominent meritocratic model). An initial meeting of the Orekit Project Management Committee was held in July 2012 [5]. This meeting led to an updated Orekit governance charter which is available at [6].

Development of the DSST started at the Computer Sciences Corporation with support from NASA GSFC in the early 1970s. Development continued at the Draper Laboratory in the 1980s.
and 1990s. Since 2001, enhancements of the DSST have been achieved by the technical staff at
the MIT Lincoln Laboratory. References [7] through [14] document DSST modeling and soft-
ware efforts that date from the 1970s and 1980s. References [15] and [16] describe the extension
of the DSST to include the 50 x 50 class of geopotential models and the J2000-based coordinate
systems, respectively.

The DSST theory for the motion of a space object replaces the conventional equations of mo-
tion (and associated variational equations) with the following formulas:

1. Equations of motion for the mean elements
2. Expressions for the short periodic motion in terms of the mean elements
3. Equations of motion for the mean element state transition matrix
4. Expressions for the short periodic partial derivatives
5. A comprehensive interpolation strategy for evaluating the components (mean el-
   ements and short-periodic coefficients) of the DSST

The intent of the semi-analytical theory is that the very small integration grid of the Cowell
numerical integration (on the order of hundreds of steps per orbital revolution) is replaced with a
much larger step (on the order of one or two steps per day) for the mean element equations of mo-
tion and for the mean element state transition matrix. Such large steps are very computationally
efficient. The short-periodic expressions are Fourier series in commonly used fast variables
(mean longitude, eccentric longitude, and true longitude). The coefficients of these Fourier series
are functions of the slowly varying mean elements \((a, h, k, p, \text{ and } q)\) and have frequency content
similar to the mean elements. The interpolation strategy for the DSST is based largely on the
slowly varying nature of these mean elements and the Fourier coefficients. Also, the motion of
the non-singular equinoctial mean elements is more linear in time and this has positive implica-
tions for orbit determination processes, particularly covariance estimation and propagation, based
on the semi-analytical theory [49].

For the geopotential perturbations, the DSST is based on a general formulation of the
geopotential in terms of the equinoctial elements [9]. This general formulation makes use of the
Jacobi polynomials and the Hansen coefficients. The classical Keplerian elements do not enter
into the theory. Further the formulation for the geopotential terms is based on recursion relations.
tends the DSST to the J2000 coordinate system.

One of the issues of the accessibility to the DSST relates to documentation. While description
of the algorithms is in the public domain, it is distributed among several items as illustrated by the
reference list for this paper. Don Danielson at the Naval Postgraduate School attempted to im-
prove this situation by producing a single integrated mathematical description of the DSST [17].
Danielson documented the top level functions in the DSST; Danielson did introduce modifi-
cations and extensions to the algorithms particularly in the automatic model truncation algorithm.
We observe that [17] was generated nearly 20 years after the original algorithms were constructed
and that when [17] was published, it had not been used as the basis for an independent software
development.
DSST applications include studies of the long-term motion of artificial satellites done in the MIT community [18 through 21]. The DSST also has been the basis of orbit propagation capabilities at The Aerospace Corporation [22 through 24]. More recently, the DSST has been used in precision orbit determination analyses [25] including Weighted Least Squares estimation of mean elements [26] and unique Kalman Filters designed to estimate mean elements directly from tracking data [27 through 30].

This wide range of applications supports the argument for the inclusion of the DSST in the Orekit package. Since Orekit employs the Java object-oriented programming platform [31, 32], migration of the DSST to Java is necessary for the DSST to be included in Orekit. Migration of the DSST to an object-oriented programming platform also was identified earlier as a demonstration task for the Open Source Software Suite for Space Situational Awareness and Space Object Catalog project [33].

Our approach to verify the Orekit DSST implementation is to compare specific Orekit DSST solutions with the corresponding GTDS DSST solutions assuming both solutions have the same initial conditions (mean elements) and physical force models. The Orekit software implementation employs the Java programming language and the Eclipse IDE [34]. The GTDS program (including the DSST) is compiled using the Intel Fortran compiler. Both Orekit and GTDS operate on a Linux PC. This Linux PC operates under the Ubuntu 11.10 64-bit server OS. Both Orekit and GTDS are comprehensive astrodynamical programs including comprehensive treatment of time systems, coordinate systems, Earth orientation parameters, force models, Earth Gravity fields, and Lunar-Solar ephemerides. In our testing, we are comparing the Orekit and GTDS DSST solutions; it is assumed that the GTDS DSST is the truth. This assumption is based on the following test history:

- Comparison of GTDS DSST results with GTDS Cowell numerical integration results
- Comparison of GTDS DSST force model results with numerical results based on the DSST force model expressions constructed using the Maxima computerized symbolic algebra system [35, 36]
- Comparison of GTDS DSST orbit propagation results with Universal Semi-Analytic Method (USM) orbit propagations [37, 38]
- GTDS DSST orbit determination results obtained by processing satellite laser ranging observation data [25]
- GTDS DSST orbit determination results obtained by processing externally generated, very precise position and velocity data from several satellite reference sources [25]

Extensive testing of the GTDS DSST using the GTDS Cowell numerical orbit propagation process was accomplished at the Draper Laboratory around 1981. This came at the conclusion of the initial effort to include short-periodic models in the GTDS DSST. These included direct comparisons and Precise Conversion of Element (PCE) Differential Correction tests. The Maxima computer algebra system was used to build explicit analytical formulas for portions of the DSST [35] and evaluation of these formulas provided test data for the Fortran 77 DSST. The Maxima satellite theory system was originally built in 1978 by Eric Zeis and was updated to cur-
rent computer platforms by Zach Folcik in 2011 – 2012 \cite{36}. The USM program is a semi-analytical orbit propagator constructed by Dr. Vasilii S. Yurasov \cite{37}. USM was developed approximately concurrently (early 1980s) with the DSST but the two efforts proceeded totally independently. Thus comparison of GTDS DSST and USM results provides insight as to GTDS DSST errors. Reference \cite{25} illustrates the results of processing very high accuracy observation data with GTDS DSST.

The roadmap of this paper is as follows. In Section 2, we describe mathematical and software preliminaries including the Generalized Method of Averaging (GMA) and design details of the GTDS DSST and Orekit DSST designs necessary for the subsequent testing. In Section 3, we compare the results of Orekit numerical integration with GTDS Cowell numerical integration. Reasonable agreement between the Orekit and GTDS numerical integrators is necessary to demonstrate compatibility of the time, coordinate system, numerical integration, and physical force model processes in the two orbit software programs. Section 4 gives results for the zonal harmonics cases. Section 5 gives results for the Lunar-Solar point mass cases. Section 6 starts the discussion of the tesseral resonance testing. Conclusions and Future Work end the paper.

MATHEMATICAL AND SOFTWARE PRELIMINARIES

The DSST is based on the Generalized Method of Averaging (GMA) with non-canonical elements. Nayfeh (Section 5.2.3 of \cite{39}) provides an introduction to the GMA. The GMA treats both the mean element motion and the short-periodic motion. In this paper, the focus is on the mean element equations of motion with multiple physical perturbations. Particularly, the small parameters are the zonal harmonics in the non-spherical gravitational potential field, the lunarsolar point mass factors, and the resonant tesseral harmonics in the non-spherical gravitational potential field. The resonant tesseral harmonics are those terms where commensurability between the mean motion of the satellite and the central body rotation rate converts a periodic term in the disturbing potential into a slowly varying term in the disturbing potential. These slowly varying terms survive the averaging process of the GMA.

McClain \cite{10} provides a derivation of the mean element equations of motion for the case of two perturbing functions. The osculating equations of motion are given in Eq (1). The quantities $\epsilon$ and $\nu$

\[
\frac{d a_i}{d t} = \epsilon F_i(a, \ell) + \nu G_i(a, \ell) \quad (i = 1, 2, \ldots, 5)
\]

\[
\frac{d \ell}{d t} = n + \epsilon F_6(a, \ell) + \nu G_6(a, \ell)
\]

(1)

are the small parameters. The $a_i (i = 1, 2, \ldots, 5)$ are the equinoctial elements a, h, k, p, and q. The phase angle $\ell$ is the mean longitude. The GMA involves the assumption of a transformation from the mean
elements to the osculating elements and an assumed form for the mean element equations of motion.

\[ \frac{d\tilde{a}_i}{dt} = \sum_{j=0}^{N} \sum_{k=0}^{M(j)} c_{ij}^{jk} B_{ij,k} + O\left(\epsilon^{N+1}\right) \]

\[ \frac{d\tilde{\ell}}{dt} = \tilde{n} + \sum_{j=0}^{N} \sum_{k=0}^{M(j)} c_{ij}^{jk} B_{ij,k} + O\left(\epsilon^{N+1}\right) \]

The identity transformation is Eq.(2) and the assumed mean element equations of motion are Eq.(3). In Eq.(2), the \( \psi_{i,j,k} \) are functions of the slow equinoctial elements and the mean longitude and are \( 2\pi \) periodic in the mean longitude. In Eq.(3), the \( B_{i,j,k} \) are functions of the slow equinoctial elements.

At the conclusion of McClain’s derivation, the mean element equations of motion to first order reduce to

\[ \frac{d\tilde{a}_i}{dt} = \epsilon \langle F_i(\tilde{a}, \tilde{\ell}) \rangle_{\tilde{\ell}} + \nu \langle G_i(\tilde{a}, \tilde{\ell}) \rangle_{\tilde{\ell}} \quad (i = 1, 2, \ldots, 5) \]

\[ \frac{d\tilde{\ell}}{dt} = \tilde{n} + \epsilon \langle F_6(\tilde{a}, \tilde{\ell}) \rangle_{\tilde{\ell}} + \nu \langle G_6(\tilde{a}, \tilde{\ell}) \rangle_{\tilde{\ell}} \]

Eq.(1) through Eq.(4) generalize to an arbitrary number of perturbing functions in a straightforward manner. The mean element equations of motion in the Orekit DSST software correspond to the generalization of Eq.(4).

The Fortran 77 DSST exists in two forms:

4. As an option within the MIT GTDS orbit determination system [25]
5. As the DSST Standalone Orbit Propagator Package [40]

In order to achieve the test results reported in this paper, we need to understand the design of the Fortran 77 DSST. Because the DSST Standalone Orbit Propagator Package is much simpler than the more comprehensive GTDS orbit determination system, we choose to describe the design of the DSST Standalone as this avoids the distraction of non-DSST modules. Development of the Standalone started in 1983 and its evolution is well described in [40]. Figures 1 through 6 describe key aspects of the Fortran 77 DSST design; understanding this design will help with the testing process. Figure 1 gives a top level view of the DSST Standalone. The overall architecture of the Fortran 77 DSST Standalone follows the standard GTDS orbit propagator design:

- Initialization of constants (GETENV, MP_READ_PMEF, GET_CSCONS, READ_EPOT)
- Initialization of quantities that depend on the initial time (epoch) (INTANL)
- Initialization of quantities that depend on the values of the orbit elements at the epoch time (this portion of the initialization would be repeated at the start of each Differential Correction iteration) (BEGANL)
- Output at request time (ORBANL)

Figures 2 and 3 expand on the initialization of the quantities that depend on the initial time. Figure 3 is important for two reasons that relate to the testing of the Orekit DSST capability. Subroutine AVrint is the driver for the algorithms to automatically tailor the DSST to the particular orbit. Subroutines INREAD and NKREAD support the usage of the modified Newcomb operators [41] in the tesseral resonance model. This usage is highlighted because it is unique in artificial satellite theory.

Figure 4 describes the architecture of the mean element rate evaluation for the conservative perturbations. Subroutine AVRANL (Figure 4) is the driver for the mean element rates. Subroutine PZONAL (Figure 5) is the driver for the zonal harmonic contributions to the element rates. Subroutine PTHIRD (also Figure 5) is the driver for the third body point mass contributions to the element rates. Subroutine PTESRS (Figure 6) is the driver for the tesseral resonance contributions to the element rates. We note that the tesseral resonance model is much more complex than either the zonal or the lunar-solar point mass models (compare Figures 5 and 6) and this observation contributed to our organization of the testing effort.

The Fortran 77 DSST Standalone design diagrams given in Figures 1 through 6 were generated using the Windows SmartDraw VP tool.

When CS Communications & Systemes decided to include the DSST in the Orekit package, they concluded that Danielson [17] was the only single document with a complete mathematical description of the DSST.

The data flow from the original Fortran 77 DSST developers to the 2011 Orekit DSST development is illustrated in Figure 7. Danielson [17] has a very extensive reference list of contributions to the Fortran 77 DSST up to early February 1995.
Figure 8 describes the interface between Flight Dynamics applications, the Orekit library, the Apache Commons Math library, and Java. The chart also illustrates, in a pictorial sense, how the DSST fits into Orekit.

CS Communications & Systemes used [17] as a reference for the Orekit DSST design and code. Initially, the Orekit DSST functional composition was significantly different from the Fortran 77 DSST. This was particularly the case for the Orekit Hansen coefficients; all the Hansen coefficient mean element rate calculations for zonal harmonics, lunar-solar point masses, and tesseral resonance were included in a single task. This contrasts with the Fortran 77 DSST standalone where the zonal harmonics, lunar-solar point masses, and tesseral resonance Hansen coefficient calculations are contained in each of the respective subroutine groupings (PZONAL, PTHIRD, and PTESRS).

During the Orekit DSST vs. Fortran 77 DSST testing process from late 2011 to the present, the design of the Orekit DSST has evolved. The computation of the Hansen coefficients is one area of such design evolution: it is no longer a single task, it is now split in three tasks, one for each contribution (zonal, third-body, and resonant tesseral). The current Orekit propagator design is given Figures 9, 10, and 11 (Orekit available propagators class diagram, DSST propagator class diagram, and DSST propagation sequence diagram, respectively). These diagrams were generated with the PlantUML plug-in [42] from a configuration file written by hand.

Next we need to discuss the establishment of an Orekit DSST test environment on the first author’s Linux PC. The following steps were followed:

1. Installing the Eclipse IDE [34] and java
2. Install the GIT configuration management tool [43]
3. Clone the Apache Commons Math library
4. Clone the Orekit library
5. Define the path dependencies and establish Commons-Math and Orekit as java projects under Eclipse

The Orekit Training document [44] discusses the procedures to clone the Apache Commons-Math library and to clone the Orekit library.

Next we executed the default test case and compared the results with those obtained in Toulouse; they agreed exactly.

Our test procedures primarily are multiple revolution orbit propagations in which we hope to demonstrate similar results with both the Orekit and GTDS DSST orbit propagators. Currently we use plots of the element histories and differences to demonstrate similarity as described in Figure 12. We also had to investigate differences in intermediate quantities. To evaluate the intermediate quantities, we executed Orekit DSST under the java interactive debugger and GTDS DSST under the Intel Fortran 77 Interactive Debugger (IDB).
CALIBRATION OF OREKIT NUMERICAL INTEGRATION

Our primary test procedure involves comparison of DSST orbit propagations made with Orekit and GTDS (Figure 12). Both Orekit DSST and GTDS DSST inherit their treatment of time, coordinate systems, and force model parameters from the overall Orekit and GTDS frameworks, respectively. For our tests to provide useful results, the overall Orekit and GTDS software frameworks should be compatible. The issue of compatibility between two comprehensive astrodynamics software programs was first investigated by Schutz et al [45]. Nouel et al [46] also considered the comparison of alternative orbit determination software. More recently, Vallado considered the compatibility of different flight dynamics programs in the context of AGI’s Satellite Tool Kit/High Precision Orbit Propagator (STK/HPOP) [47].

We chose the Starlette case from [45] to investigate the compatibility of the Orekit and GTDS numerical integration. We assume the epoch orbit elements given in Table 1. We constructed a GTDS geopotential file that includes the Eigen-5C geopotential [48]. We modified the GTDS GM to be $398600.4415 \text{ km}^3\text{sec}^{-2}$ and the radius of the Earth to be $6378.13646 \text{ km}$; these values were the defaults in Orekit. In GTDS, we used an integration stepsize of 30 seconds with the usual 12th order summed Cowell/Adams predict-correct algorithm. We integrated over a 14 day interval and generated output state vectors and orbital elements in the J2000 coordinate system on a 15 minute grid.

Table 1: Starlette Orbit Calibration Test for Orekit and GTDS Numerical Integration ($J_2$ only)

<table>
<thead>
<tr>
<th>Osculating Keplerian Element</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch</td>
<td>2000 November 22, 0 hr 0 min 0.0 sec UTC</td>
</tr>
<tr>
<td>Semi-major Axis</td>
<td>7336.0 km</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.02</td>
</tr>
<tr>
<td>Inclination</td>
<td>49.7 deg</td>
</tr>
<tr>
<td>Right Ascension of the Ascending Node</td>
<td>0.0 deg</td>
</tr>
<tr>
<td>Argument of Perigee</td>
<td>0.0 deg</td>
</tr>
<tr>
<td>Mean Anomaly</td>
<td>100.0 deg</td>
</tr>
<tr>
<td>J2000 Coordinate System</td>
<td></td>
</tr>
<tr>
<td>Eigen-5C geopotential</td>
<td></td>
</tr>
</tbody>
</table>

We included just the $J_2$ perturbation with polar motion in the initial calibration exercise although Table IV in [45] suggests several additional cases. Such additional calibration cases might involve tesseral geopotential harmonics and lunar-solar ephemerides.

The numerical results for this calibration exercise are given in Figures 13 through 21. Figures 13, 14, and 15 give the Radial, Cross-Track, and Along-Track position differences between Orekit and GTDS. The differences are small: growing to about 1.5 m at 14 days. However, there
is a secular, growing trend and we will need to consider that further. Figures 16 and 17 give the osculating semi-major axis time histories and differences, respectively. The Figure 16 semi-major axis histories nearly overlap each other. The Figure 17 semi-major axis differences exhibit a linear growth compatible with the along-track positional differences. Moreover, there is an apparent bias in the mean value of the semi-major axis difference (about 0.5 x 10^{-3} meters). We suspect that this bias is related to the observed along-track difference. Figures 18 and 19 give the osculating equinoctial element h time histories and differences, respectively. The equinoctial element h exhibits behavior consistent with the long-periodic motion expected in the equinoctial h and k elements. The Figure 18 osculating element h histories nearly overlap each other. The equinoctial element h difference also exhibits a small initial bias. Figures 20 and 21 give the osculating equinoctial element p time histories and differences, respectively. The motion in the equinoctial element p is much larger in accord with the nodal drift. Again, the Orekit and GTDS equinoctial p element histories nearly overlap. When we look at the equinoctial p difference, it appears to be quadratic.

While we would like to further improve the agreement between the Orekit and GTDS numerical integrators, we deemed the current agreement to be sufficient to initiate comparisons of the Orekit and GTDS DSST implementations.

ZONAL HARMONICS

We chose an orbit with mean semi-major axis, eccentricity, and inclination approximating a near-circular, sun-synchronous, repeat ground-track configuration to test the zonal harmonics capabilities of the Orekit DSST. This orbit has a mean altitude near 820 km. Some of the very early testing of the Semi-analytical Satellite Theory by Ken Duck at NASA GSFC in the mid 1970s employed similar orbital configurations. We use the mean elements from Table 2 to initialize our Orekit and DSST orbit propagations.

We exercised the error analysis flow given in Figure 12. The resulting plots are given in Figure 22. We obtained good qualitative agreement in the p and q equinoctial elements but only partial agreement in the h and k equinoctial elements. GTDS DSST and Orekit DSST h, k, p, and q equinoctial elements all exhibited the characteristics of a perturbed linear oscillator as we expected and was discussed by Zeis in [35]. In addition, we observe that the h and k oscillations in Figure 22 have the same magnitude for GTDS and Orekit. However, the h and k angular velocities are quite different between GTDS and Orekit.

Our next debugging effort was to consider the mean element rates at the epoch time. We used the Fortran IDB debugger to gain access to equinoctial mean element rates available in ANAVR (the GTDS version of Standalone subroutine AVRANL) after the first call to the zonal harmonics subroutine PZONAL (see Figure 4). We also used the java debugger to gain access to the epoch time zonal harmonic mean equinoctial element rates computed in the class
Table 2: LEO Test Case for Orekit DSST and GTDS DSST (J\textsubscript{2}-only and J\textsubscript{2} through J\textsubscript{36} cases)

<table>
<thead>
<tr>
<th>Mean Keplerian Element</th>
<th>2011 December 12, 11 hr 57 min 20.0 sec UTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-major Axis</td>
<td>7204.53584810944 km</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.0012402238462686</td>
</tr>
<tr>
<td>Inclination</td>
<td>98.74341600466741 deg</td>
</tr>
<tr>
<td>Right Ascension of the Ascending Node</td>
<td>43.3299011079034 deg</td>
</tr>
<tr>
<td>Argument of Perigee</td>
<td>111.199017507663 deg</td>
</tr>
<tr>
<td>Mean Anomaly</td>
<td>68.6685250972562 deg</td>
</tr>
</tbody>
</table>

J2000 Coordinate System

Eigen-5C geopotential

\[ \text{DSSTCentralBody.java}^* \]. The epoch values of the p and q element rates agreed at the level of three to four significant digits between Orekit and GTDS. There was no agreement in the h and k equinoctial element rates between Orekit and GTDS. However, we did observe that the ratio of the time derivative of the mean equinoctial h element divided by the time derivative of the mean equinoctial k element was consistent between GTDS and Orekit. The h-dot/k-dot ratio agreed between GTDS and Orekit at the same level as the agreement for p-dot and q-dot.

Next we used Matlab to construct the J\textsubscript{2}-rates using the Macsyma-based formulas given by Zeis [35]. These agreed with the GTDS rates.

At this point we decided to undertake a detailed review of the DSST algorithms used for the zonal harmonics rates. While Orekit was coded from the Danielson report [17], the original analyses in [8] and [11] are also relevant. Equation (1) in Section 3.1 of [17] suggests that we should focus on the partial derivatives of the zonal potential with respect to h and k. Equation (6) in Section 3.1 of [17] gives these partial derivatives. When an obvious typographical error is corrected, the equations are:

\[
\frac{\partial U}{\partial h} = -\frac{\mu}{a} \sum_{s,n} (2 - \delta_{0,s}) \left(\frac{R}{a}\right)^n J_n V_{n-s} Q_{n-s} \left( K_{0}^{-n-1,s} \frac{\partial G_s}{\partial h} + h \chi^3 G_s \frac{dK_{0}^{-n-1,s}}{d \chi} \right)
\]

\[
\frac{\partial U}{\partial k} = -\frac{\mu}{a} \sum_{s,n} (2 - \delta_{0,s}) \left(\frac{R}{a}\right)^n J_n V_{n-s} Q_{n-s} \left( K_{0}^{-n-1,s} \frac{\partial G_s}{\partial k} + k \chi^3 G_s \frac{dK_{0}^{-n-1,s}}{d \chi} \right)
\]

\[ (5) \]

* We are referring to the Orekit architecture as of June 2012.
We focus our attention on the Hansen coefficient kernels \( K_{-n-1,s} \) and particularly on the derivatives of the Hansen coefficient kernels \( \frac{dK_{-n-1,s}}{d\chi} \) because the other factors come into the \( p \) and \( q \) element rates. Our review of the Hansen coefficient formulas for the Zonal harmonics is summarized in Table 3.

### Table 3: Zonal Harmonic Model Reconciliation

<table>
<thead>
<tr>
<th>Document</th>
<th>Averaged Potential Model</th>
<th>Partial Derivatives of the averaged potential</th>
<th>Averaged Equations of motion</th>
<th>Hansen coefficient initialization</th>
<th>Hansen coefficient recursion</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIAA 75-9 (Cefola &amp; Broucke) [1975] [8]</td>
<td>75-9: Eq (118), p.15</td>
<td>75-9: See discussion on p. 15</td>
<td>74-170: Eq (51) thru (55)</td>
<td>75-9: Eq (81), Eq (82)</td>
<td>75-9: Eq (80)</td>
</tr>
<tr>
<td>AIAA 74-170 (Cefola, Long, &amp; Holloway)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yellow Book (W. McClain) [1978] [13]</td>
<td>Eq (3-183) which references (3-181) and (182)</td>
<td>Eq (3-187a) thru (3-187g), (3-188a-f), (3-189)</td>
<td>Eq (3-202a) thru Eq (3-202f)</td>
<td>Eq (3-193a,b)</td>
<td>Eq (3-193c)</td>
</tr>
<tr>
<td>Semi-analytic Satellite Theory (Danielsen et al, Naval Postgraduate School) [1995] [17]</td>
<td>Eq (3) in Section 3.1, p. 55</td>
<td>Eqs (6), (8), (9) in Section 3.1, p. 55</td>
<td>Eq (1) in Section 3.1, p. 55; This follows from Eq (10), Section 2.2, p. 16</td>
<td>Eq (6), Section 2.7.3, p. 50</td>
<td>Eq (6), Section 2.7.3, p. 50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. 74-170 does not give a differential equation for the perturbed portion of the mean longitude rate.

The formula for the recursion for the derivative of the Kernel, Eq (7) in Section 3.1 of [17] is essential. This relation is obtained by differentiating Eq. (6) in Section 2.7.3 of [17]. We carried out this differentiation and obtained:

\[
\frac{dK_{-n-1,s}}{d\chi} = \frac{(n - 1)\chi^2}{(n + s - 1)(n - s - 1)} \left[ (2n - 3)\frac{dK_{-n,s}}{d\chi} - (n - 2)\frac{dK_{-n+1,s}}{d\chi} \right] + \frac{2}{\chi} K_{-n-1,s} \quad n > s+1
\]

Clearly, there is sign discrepancy between the result just derived and Eq (7) in Section 3.1 of [17]. We used the **Java** debugger to verify that the Orekit DSST employed Eq (7) in Section 3.1 of [17].
The fix for this problem was to modify Orekit DSST class **HansenCoefficients.java** (again referring to the Orekit DSST architecture of June 2012). With the corrected code, Figures 23 through 26 were produced. In these figures, the Fortran 77 DSST and the Orekit DSST with the modified code produced consistent results. We also exercised this same test case with a J$_2$ through J$_{36}$ geopotential. This produced Figures 27 through 30. Again the Fortran 77 DSST and the Orekit DSST were in accord. It was interesting to note that the h and k element histories were significantly impacted by the 36 x 0 geopotential.

Perhaps most important, this test procedure taught us that we needed to include review of the analytical formulas from [17] in our verification procedure.

One final note: consideration of the J$_2$ element motion formulas from Zeis [35] led us to the simple formula for the ratio of h-dot/k-dot. This formula is a consequence of fact that the mean eccentricity is constant for the J$_2$ perturbation:

$$\frac{h - \dot{h}}{k - \dot{k}} = -\frac{k}{h}$$  \hspace{1cm} (7)

This formula did reproduce the h-dot/k-dot ratio observed with GTDS DSST and Orekit DSST.

**LUNAR-SOLAR POINT MASS TEST CASES**

We selected the Sirius 24 hr orbit from [38] as our test case for lunar-solar point masses. Some of the authors of this paper had previously compared the GTDS DSST orbit propagator and the Universal Semianalytical Method (USM) orbit propagator due to V. S. Yurasov [37] for this case. The Lunar-Solar point masses cause strong orbital element motion in the eccentricity and the inclination for this case. The assumed mean elements are given in Table 4. These elements give the ‘regional’ ground-track illustrated in Figure 31. The Sirius orbit employs an argument of perigee of 270 degrees resulting in coverage in the Northern Hemisphere. The Sirius orbit employs the critical inclination.

Based on our experience with zonal harmonics, we constructed a table to trace the Hansen coefficient formulas in the several source documents. For the third-body point masses, these results are given in Table 5.
Table 4: Sirius 24 hr Orbit Test Case for Orekit DSST and GTDS DSST

Mean Keplerian Elements

<table>
<thead>
<tr>
<th>Epoch</th>
<th>2000 July 1, 0 hr 0 min 0.0 sec UTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-major Axis</td>
<td>42163.393 km</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.2684</td>
</tr>
<tr>
<td>Inclination</td>
<td>63.435 deg</td>
</tr>
<tr>
<td>Right Ascension of the Ascending Node</td>
<td>285.0 deg</td>
</tr>
<tr>
<td>Argument of Perigee</td>
<td>270.0 deg</td>
</tr>
<tr>
<td>Mean Anomaly</td>
<td>344.0 deg</td>
</tr>
</tbody>
</table>

J2000 Coordinate System
Eigen-5C geopotential (J2)
Selected Lunar-Solar Point Mass terms

Table 5: Lunar-Solar Point Mass Model Reconciliation

<table>
<thead>
<tr>
<th>Document</th>
<th>Averaged Potential Model</th>
<th>Partial Derivatives of the averaged potential</th>
<th>Averaged Equations of motion</th>
<th>Hansen coefficient initialization</th>
<th>Hansen coefficient recursion</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIAA 75-9 (Cofola &amp; Broude) [1973] [8]</td>
<td>Eq (91), p. 13</td>
<td>Eq (101) thru Eq (105), p. 14</td>
<td>Eq (51) thru Eq (55)</td>
<td>Eq (63), Eq (64)</td>
<td>Eq (62)</td>
</tr>
<tr>
<td>AIAA 74-170 (Cofola, Long, &amp; Holloway)</td>
<td>Eq (4-92), p 4-41</td>
<td>Eq (4-95a) thru Eq (4-95f), (4-96) thru (4-99)</td>
<td>Eq (3-171a) thru Eq (3-171f)</td>
<td>Eq (4-107)</td>
<td>Eq (4-108)</td>
</tr>
<tr>
<td>Yellow Book (W, McClain) [1978] [11]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semi-analytic Satellite Theory (Danielsen et al, Naval Postgraduate School) [1995] [17]</td>
<td>Eq (1) in Section 3.2, p. 60</td>
<td>Eq (2) in Section 3.2, p. 60</td>
<td>Eq (1) in Section 3.1, p. 55; This follows from Eq (10), Section 2.2, p. 26</td>
<td>Eq (8), Section 2.7.3, p. 50</td>
<td>Eq (7), Section 2.7.3, p. 50</td>
</tr>
</tbody>
</table>

We found confusion about the initialization of the Hansen coefficient recursions for the third-body perturbations {Eq. (7) in Section 2.7.3 of [17] and Eq.(3) in Section 3.2 of [17]}.  

14
When these issues were resolved, we obtained the results given in Figures 32 through 40. Figures 32 and 33 give the eccentricity and inclination time histories for the $J_2$ plus lunar point mass case. Both exhibit a long period or secular signal with 14 day oscillations superimposed. Figures 34 and 35 give the eccentricity and inclination time histories for the $J_2$ plus solar point mass case. Both exhibit a long period or secular signal with 180 day oscillations superimposed. In these figures, the GTDS and Orekit signals are closely aligned. Figures 36 and 37 give the equinoctial element $h$ and $k$ time histories for the same case ($J_2$ plus solar point mass). There are small differences between the GTDS DSST and Orekit DSST signals. Figures 38 and 39 give the $h$ and $k$ equinoctial element differences for this case; these differences exhibit the same 180 day period plus secular/long-period structure. These small differences may relate to differences between the truncation algorithms employed in GTDS and Orekit.

Figure 40 gives Fortran 77 DSST Standalone and Orekit results for the argument of perigee and the equinoctial $h$ and $k$ elements for the $J_2$ plus lunar and solar point masses case. The first part of Figure (40) for the argument of perigee compares well with the argument of perigee time history for the Sirius case given in [38]. Figure 40 was generated in Toulouse at CS Communications & Systems.

**TESSERAL RESONANCE TESTING**

The tesseral resonance is much more complex than the previous models (zonal harmonics and third-body point masses) we have considered. We can see this complexity by reviewing the mean element equations of motion and the disturbing potential for the tesseral resonance from Section 3.3 of [17].

\[
\begin{align*}
\dot{a} &= \frac{2a}{A} \frac{\partial U}{\partial \lambda} \\
\dot{h} &= \frac{B}{A} \frac{\partial U}{\partial k} + \frac{k}{AB} \left( pU_{\alpha \gamma} - IQ_{\alpha \beta} \right) - \frac{hB}{A(1+B)} \frac{\partial U}{\partial \lambda} \\
\dot{k} &= -\left[ \frac{B}{A} \frac{\partial U}{\partial h} + \frac{h}{AB} \left( pU_{\alpha \gamma} - IQ_{\alpha \beta} \right) + \frac{kB}{A(1+B)} \frac{\partial U}{\partial \lambda} \right] \\
\dot{p} &= \frac{C}{2AB} \left[ p \left( U_{h \gamma} - U_{\alpha \beta} \right) \frac{\partial U}{\partial \lambda} - U_{h \beta} \right] \\
\dot{q} &= \frac{C}{2AB} \left[ q \left( U_{h \gamma} - U_{\alpha \beta} \right) \frac{\partial U}{\partial \lambda} - IU_{\alpha \gamma} \right] \\
\dot{\lambda} &= \frac{2a}{A} \frac{\partial U}{\partial a} + \frac{B}{A(1+B)} \left( \frac{\partial U}{\partial h} + \frac{k}{\partial \lambda} \right) + \frac{1}{AB} \left( pU_{\alpha \gamma} - IQ_{\alpha \beta} \right) \\
\end{align*}
\]

(8)

\[
U = \text{Re} \left\{ \sum_{j} \sum_{m=1}^{M} \sum_{n=-N}^{N} \sum_{n=\text{max}(2,m,s)}^{N} \left( \frac{R}{a} \right)^{n-m} \Gamma_{ms}^{n} \Gamma_{ns}^{n} K_{j}^{n-1,s} p_{j}^{n} \right\} \\
\{ G_{ms}^{j} + i H_{ms}^{j} (C_{mn} - i S_{nm}) \exp[i(j \lambda - m \theta)] \}
\]

(9)
We can also gain insight with regard to the complexity of the resonance model by comparing the software designs given in Figures 5 and 6.

The tesseral resonance terms involve linear combinations of the mean longitude and the Greenwich Hour Angle. These resonant combinations are slowly varying due to the commensurability between the mean motion and the GHA rate. As a result, the partial derivatives of the potential with respect to the mean longitude \( \lambda \) must be considered. Resonance is the first case where the partial derivatives of the potential with respect to the mean longitude are not exactly zero.

In the disturbing potential [Eq.(9)], the following are sources of additional complexity:

- The determination of the \( j \) and \( m \) indices for the resonant coefficient pairs
- The complexity of the Hansen coefficients when the subscript is not equal to zero
- The Jacobi polynomials
- The two dimensional array of C,S geopotential coefficients

Initially, we intended to use GPS orbit cases with 8x8 geopotential from [50] to test the Orekit DSST tesseral resonance capability. However, we soon realized that we needed to fallback to simpler test cases for the initial test. For the initial test case, we used a geostationary orbit [51] perturbed by the 2 x 2 field. The elements for this case are given in Table 6.

<table>
<thead>
<tr>
<th>Table 6: GEO 24 hr Orbit Test Case for Orekit DSST and GTDS DSST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Keplerian Elements</td>
</tr>
<tr>
<td>Epoch</td>
</tr>
<tr>
<td>2007 April 16, 0 hr 46 min 42.4 sec UTC</td>
</tr>
<tr>
<td>Semi-major Axis</td>
</tr>
<tr>
<td>42166.258 km</td>
</tr>
<tr>
<td>Eccentricity</td>
</tr>
<tr>
<td>0.0001</td>
</tr>
<tr>
<td>Inclination</td>
</tr>
<tr>
<td>0.001 deg</td>
</tr>
<tr>
<td>Right Ascension of the Ascending Node</td>
</tr>
<tr>
<td>130.7562 deg</td>
</tr>
<tr>
<td>Argument of Perigee</td>
</tr>
<tr>
<td>315.4985 deg</td>
</tr>
<tr>
<td>Mean Anomaly</td>
</tr>
<tr>
<td>44.2377deg</td>
</tr>
<tr>
<td>J2000 Coordinate System</td>
</tr>
<tr>
<td>Eigen-5C geopotential</td>
</tr>
</tbody>
</table>

We note that introducing small eccentricity and inclination into Eq.(8) greatly simplifies the formulae.

In the testing to date, we have uncovered bugs associated with:

- initialization of the GHA rate,
- initialization of the $S_{nm}$ portion of the geopotential, and
- assembly of the total element rates due to resonance with the zonal and lunar-solar point mass rates

With these fixes to Orekit DSST, we obtained the results given in Figures 41 and 42 for the GEO case with 2x2 geopotential field and a 200 day interval. The Orekit DSST and the Fortran 77 Standalone DSST results agree closely for both the mean equinoctial and Keplerian elements.

We also exercised this version of Orekit DSST for a GPS orbit case [50]. The elements for this case are given in Table 7.

**Table 7: GPS 12 hr Orbit Test Case for Orekit DSST and GTDS DSST**

<table>
<thead>
<tr>
<th>Mean Keplerian Elements</th>
<th>2007 April 16, 0 hr 46 min 42.4 sec UTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch</td>
<td></td>
</tr>
<tr>
<td>Semi-major Axis</td>
<td>26559.89 km</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.0041632</td>
</tr>
<tr>
<td>Inclination</td>
<td>55.2 deg</td>
</tr>
<tr>
<td>Right Ascension of the Ascending Node</td>
<td>130.7562 deg</td>
</tr>
<tr>
<td>Argument of Perigee</td>
<td>315.4985 deg</td>
</tr>
<tr>
<td>Mean Anomaly</td>
<td>44.2377 deg</td>
</tr>
<tr>
<td>J2000 Coordinate System</td>
<td></td>
</tr>
<tr>
<td>Eigen-5C geopotential</td>
<td></td>
</tr>
</tbody>
</table>

For the GPS case with 2x2 geopotential field, we obtained the results given in Figures 43 and 44 for a 10 year interval. Again, the Orekit DSST and the Fortran 77 Standalone DSST results agree closely for both the mean equinoctial and Keplerian elements.

However, when we exercise a test case with multiple resonant coefficient C,S pairs such as a GPS 3x3 case, we do not achieve agreement between Orekit DSST and the Fortran 77 DSST Standalone.

Clearly we need to reconcile the expressions for the tesseral resonance in [11] and [17] similar to the way we discussed the analytical results for zonal harmonics and the lunar-solar point masses. We also need to further investigate test cases which marginally increase the complexity of the resonance potential.

One test case of interest is the GPS test case of Table 7 with the 4x4 geopotential. Figure 45 gives the Keplerian element histories for the GPS orbit with 4x4 geopotential. The difference in
the inclination time histories is of particular interest. The equinoctial element differences for this same case are given in Figure 46. The relatively large differences in the equinoctial p and q seem to correlate with the anomaly in the inclination time histories (Figure 45). We have also investigated the GPS resonance case using the RTVOP maxima utility [35, 36]. RTVOP allows us to specify the level of eccentricity truncation we desire. If we consider the GPS 4x4 case and look at the order zero in the eccentricity terms, then Table 8 results.

Table 8. Contributions to the GPS Resonant Equations of Motion with Order Zero in the Eccentricity

<table>
<thead>
<tr>
<th>C,S Pair</th>
<th>a-dot</th>
<th>h-dot</th>
<th>k-dot</th>
<th>p-dot</th>
<th>q-dot</th>
<th>lambda-dot</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,2</td>
<td>-</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3,2</td>
<td>yes</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>4,2</td>
<td>-</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4,4</td>
<td>yes</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Anomalies in the p and q element differences (Figure 46) suggest that the (3,2) p and q rates are not being assembled consistently. The GPS case has been exercised with the 3x3 geopotential and a similar difference in the inclination time histories is exhibited (Figure 47). We are continuing the investigation.

CONCLUSIONS

The validation of the open-source java Orekit DSST implementation using the previous Fortran 77 implementations was first discussed by Luc Maisonobe and Paul Cefola at the September 2011 Lille University workshop on Orbit Propagation and Orbit Determination organized by Florent Deleflie.

Subsequently this validation process was referenced in a written agreement.

Detailed comparisons between Orekit DSST and GTDS SST (and the Fortran 77 DSST Standalone Orbit Propagator) were begun in April 2012. An initial presentation on the test effort including the test plan was made at the 5th International Conference on Astrodynamics Tools and Techniques held at ESTEC/ESA, Noordwijk, The Netherlands, May-June, 2012.

The initial design and algorithms for the java Orekit DSST followed the report prepared by Don Danielson at the Naval Postgraduate School (NPS), Monterey, CA, USA in 1995. The Danielson report primarily intended to provide a single document for the existing Fortran 77 DSST; however, Danielson did introduce modifications to the notation and refinements to the DSST theory, particularly in the area of model truncation. More important, we need to understand that the Orekit DSST was the first instance of code being created from the Danielson report. Using the Danielson report as a ‘design-to’ and ‘code-to’ specification does put greater emphasis on precision and correctness. Specifically we had to learn that verifying the Orekit DSST models includ-
ed both review of the performance of the java code and review of the portions of the Danielson document from which the algorithms were drawn.

We cloned the various portions of the Orekit project to a custom-built Linux PC. This Linux machine also provides the platform on which we execute GTDS. On this machine, we access Orekit DSST through the Eclipse IDE. This has proven to be a simple approach for maintaining compatibility between the Orekit code in Toulouse and the code on Martha’s Vineyard.

Really what we are doing is porting a comprehensive capability (DSST) from the GTDS orbit determination system to the Orekit Flight Dynamics library. In order to sensibly test this port, we need to have compatibility in time, coordinate systems, force models, lunar-solar ephemeris, and Earth orientation parameters. So far we have done a calibration exercise involving the Orekit numerical integration and the GTDS Cowell integration with $J_2$ force model and J2000 coordinates; this tests time and coordinate systems and a very limited portion of the force model.

So far we have eliminated the major bugs in the mean element equations of motion for the zonal harmonics and the lunar-solar point mass perturbations. Testing of the tesseral resonance model is in progress. This testing has been achieved against a moving baseline as Orekit DSST has been fixed as bugs are uncovered. This moving baseline has been helpful because both the lunar-solar and tesseral resonance cases also involve the zonal harmonics.

**FUTURE WORK**

The following is a list of future work items:

1. Complete the testing of the tesseral resonance model
2. Introduce and test a $J_2$-squared term model for the mean element equations of motion
3. Extend the calibration effort to include additional terms and improved agreement
4. Introduce a larger range of test cases that better span the orbit element space
5. Introduce a quantitative accuracy metric for assembling the results of several orbit propagation test cases (this has been done earlier in a Space Situational Awareness context)
6. Iterate through the test cases to provide a regression test as the Orekit DSST software baseline evolves. This should be automated.
7. Develop a plan for testing the DSST short-periodic models

**ACKNOWLEDGMENTS**

The authors would like to acknowledge the support and encouragement of Mr. Nicolas Frouvelle of CS Communications & Systèmes, Toulouse, France. The authors would like to acknowledge the original developers of the DSST Satellite Theory and its implementation in Fortran 77 including Mr. Wayne McClain, Mr. Leo Early, Dr. Ron Proulx, Dr. Mark Slutsky, Dr. David Carter, and Mr. Rick Metzinger. The authors also acknowledge the efforts of Prof. Don
Danielson to prepare a single integrated document describing the DSST. The authors also acknowledge the several MIT Aeronautics and Astronautics Department graduate students who participated in the development, test, evaluation, and application of the Semi-analytical Satellite Theory.

The work of graduate student Barry Bentley at the University of Cambridge (UK) was partially supported by an ESA Summer of Code in Space (SOCIS) project during 2012. This SOCIS project was arranged by CS Communications & Systems.

The work of Zachary Folcik is sponsored by the Department of the Air Force under contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Dr. Vasily S. Yurasov (then at Space Research Center “Kosmos”) provided the USM program which was very helpful during the testing of the lunar-solar point mass model.

The numerical results described in this paper were generated on a multi-core PC with an Intel Core i7 960 Processor and an NVIDIA GEFORE GTX 580 graphical processor running the Linux Ubuntu 11.10 server distribution. This machine was designed, assembled, tested, and maintained by MV Tech. Inc., Vineyard Haven, MA.
REFERENCES


5. Minutes of first Orekit Project Management Committee (PMC) meeting, https://www.orekit.org/forge/attachments/318/PMC_meeting_050712.pdf


15. Fonte, Daniel J. Jr., Implementing a 50 x 50 Gravity Field Model in an Orbit Determination System, Master of Science Thesis, Department of Aeronautics and Astronautics, MIT, June 1993 (CSDL-T-1169).


33. Cefola, Paul J., Brian Weeden, and Creon Levit, **Open Source Software Suite for Space Situational Awareness and Space Object Catalog Work**, presented at the International Conference on Astrodynamics Tools and Techniques (ICATT), ESA/ESAC, Madrid, Spain, 3-6 May 2010


42. http://plantuml.sourceforge.net/ accessed 1 February 2013. (PlantUML is an open-source tool written in Java that uses simple textual descriptions to draw UML diagrams.)


44. Maisonobe, Luc, **Orekit Training**, CS Communications & Systemes, 2010-2012


Figure 1. Fortran 77 DSST Standalone Program Top Level Design

Figure 2. Fortran 77 DSST Standalone Program Initialization of the DSST integration parameters and Force Models
Figure 3. Fortran 77 DSST Standalone Program Initialization of the DSST integration parameters including the Newcomb Operators

Figure 4. Fortran 77 DSST Standalone Program Mean Element Rates
Figure 5. Fortran 77 DSST Standalone Program Software for the Zonal Harmonics and Lunar-Solar Point Mass Mean Element Rates

Figure 6. Fortran 77 DSST Standalone Program Software for the Tesseral Resonance Mean Element Rates
Figure 7. Data Flow from the Fortran 77 DSST Development to the java Orekit DSST Design

Figure 8. Design of Flight Dynamics Applications using the Orekit Library
Figure 9. Orekit Available Propagators Class Diagram
Figure 10. Orekit DSST Propagator Class Diagram
Figure 11. Orekit DSST Propagation Sequence Diagram
Figure 12. Orekit DSST vs. GTDS DSST Test Data Flow

Figure 13. Numerical-Integration based Radial coordinate difference for Starlette J2 case (Orekit & GTDS)
Figure 14. Numerical Integration-based Cross Track coordinate difference for Starlette J2 case (Orekit & GTDS)

Figure 15. Numerical Integration-based Along Track coordinate difference for Starlette J2 case (Orekit & GTDS)
Figure 16. Numerical Integration-based Semi-major Axis Time Histories for Starlette J2 case (Orekit & GTDS)

Figure 17. Numerical Integration-based Semi-major Axis Difference for Starlette J2 case (Orekit & GTDS)
Figure 18. Numerical Integration-based Equinoctial Element $h$ Time Histories for Starlette J2 case (Orekit & GTDS)

Figure 19. Numerical Integration-based Equinoctial Element $h$ Differences for Starlette J2 case (Orekit & GTDS)
Figure 20. Numerical Integration-based Equinoctial Element $p$ Time Histories for Starlette J2 case (Orekit & GTDS)

Figure 21. Numerical Integration-based Equinoctial Element $p$ Differences for Starlette J2 case (Orekit & GTDS)
Figure 22. DSST Mean Equinoctial Element $h$, $k$, $p$, and $q$ time histories for a LEO orbit over 365 days prior to the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ only)

Figure 23. DSST Mean Equinoctial Element $h$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ only)
Figure 24. DSST Mean Equinoctial Element $k$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ only)

Figure 25. DSST Mean Equinoctial Element $p$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ only)
Figure 26. DSST Mean Equinoctial Element $q$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ only)

Figure 27. DSST Mean Equinoctial Element $h$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ thru $J_{36}$)
Figure 28. DSST Mean Equinoctial Element $k$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ thru $J_{36}$)

Figure 29. DSST Mean Equinoctial Element $p$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ thru $J_{36}$)
Figure 30. DSST Mean Equinoctial Element $q$ time history for a LEO orbit over 365 days after the fix of Orekit class HansenCoefficients.java (Orekit & GTDS) ($J_2$ thru $J_{36}$)

Figure 31. Sirius 24 hr Case Ground-track
Figure 32. DSST Mean Eccentricity time histories for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Lunar point mass perturbation)

Figure 33. DSST Mean Inclination time histories for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Lunar point mass perturbation)
Figure 34. DSST Mean Eccentricity time histories for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Solar point mass perturbation)

Figure 35. DSST Mean Inclination time histories for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Solar point mass perturbation)
Figure 36. DSST Mean Equinoctial Element $h$ time histories for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Solar point mass perturbation)

Figure 37. DSST Mean Equinoctial Element $k$ time histories for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Solar point mass perturbation)
Figure 38. DSST Mean Equinoctial Element $h$ differences for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Solar point mass perturbation)

Figure 39. DSST Mean Equinoctial Element $k$ differences for the Sirius orbit over 730 days (Orekit & GTDS) ($J_2$ plus Solar point mass perturbation)
Figure 40. DSST Orbit Element Histories for the Sirius orbit over 365 days (Orekit & GTDS) ($J_2$ plus Lunar plus Solar point mass perturbation)

Figure 41. DSST Mean Equinoctial Element Histories for the GEO orbit over 200 days (Orekit & GTDS) (2x2 perturbation including tesseral resonance)
Figure 42  DSST Mean Keplerian Element Histories for the GEO orbit over 200 days (Orekit & GTDS) (2x2 perturbation including tesseral resonance)

Figure 43  DSST Mean Equinoctial Element Histories for the GPS orbit over 10 years (Orekit & GTDS) (2x2 perturbation including tesseral resonance)
Figure 44 DSST Mean Keplerian Element Histories for the GPS orbit over 10 years (Orekit & GTDS) (2x2 perturbation including tesseral resonance)

Figure 45 DSST Mean Keplerian Element Histories for the GPS orbit over 10 years (Orekit & GTDS) (4x4 perturbation including tesseral resonance)
Figure 46  DSST Mean Equinoctial Element Differences for the GPS orbit over 10 years (Orekit & GTDS) (4x4 perturbation including tesseral resonance)

Figure 47  DSST Mean Inclination History for the GPS orbit over 10 years (Orekit & GTDS) (3x3 perturbation including tesseral resonance)