Readme file for paper AIAA 2008-6770

SGP4 Orbit Determination

David A. Vallado[1]
Center for Space Standards and Innovation, Colorado Springs, Colorado, 80920

Paul Crawford[2]
University of Dundee, Dundee, UK

Notes:

2008-08-28

- Baseline version to accompany the differential correction paper.
- There are three test modes of operation:
- **1. TLE tests**
  This test simply inputs a TLE file (twoline.txt), perturbs the initial state, and then tries to recover the initial TLE. Since the answer is known, differences are given to the original TLE. Recognize that these tests contain some difficult tests and not all will converge.
- **2. Ephemeris conversion tests**
  This test inputs an ephemeris file ("dot-e" format from STK which is simply a header and points of secs from epoch, and space delimited position and velocity vectors. See also http://www.agi.com/resources/help/online/source/extfile/stk/stk/importfiles-02.htm) in ECI (J2000), TEME, or ECEF (Fixed) and finds a representative TLE. Example files are given for a GPS satellite (gpsd6n22.e) and a numerically generated satellite (sat1.e) that showed some problems with various fit spans as discussed in the paper.
- **3. Observation conversion tests**
  This test inputs an observation file ("plain" format from the paper which is simply a header and space delimited points of date, and the observation values), and finds a representative TLE. The test case (geos5.inp) is from the example discussed in the paper. The baseline does not solve this accurately for all configurations and additional work is needed to determine what is causing the difficulties.

- There are two modes of program operation:
- 1. Default – command-line arguments
  User enters parameters as command line arguments. Some additional parameters are needed for the test modes as shown below. The parameters are as follows (Note opsmode of afspc is set for all cases inside sgp4).
  s, b – matrix inversion solution method, SVD or LU-back substitution
  r, x, t – test mode of observations, external ephemerides, or TLE test
  e, t, v – statetype, equinoctial, two-line Keplerian, or position and velocity vectors
  0.001 – percentchg, the percentage change for the finite differencing
  0.0000001– deltaamtchg, the amount in finite differencing below which to set a value to avoid divide by zeros.
  0.0002– rmsepsilon, the rms tolerance to stop the iterations
  filename
  72 – number of points per period if TLE test, number of points to fit if Ephemeris Test, last ob if observation test.
  7 – 6 or 7 state size. If solving for BStar, use 7. GEOs usually do better with 6.

  Test mode 1: c:\testdc s t e .001 .0000001 .0002 twoline.txt 72 7
  Test mode 2: c:\testdc s x e .001 .0000001 .0002 gpsd6n22.e 300 7
  User then inputs point spacing (every point, every 2nd, etc), and the total number of points to process
  Test mode 3: c:\testdc s r e .001 .0000001 .0002 geos5.inp 20 7
  the 20 is currently ignored in favor of processing all obs

[1] Senior Research Astrodynamicist, Center for Space Standards and Innovation, 7150 Campus Dr, Suite 260, dvallado@centerforspace.com, AIAA Associate Fellow.
[2] Principal Engineer, psc@sat.dundee.ac.uk.

- 2. Interactive – enter commands from a prompt
  User is prompted for the inputs. Inputs will vary with the different test mode.

Revision Notes:

None.

Code Notes:

The C++ executable needs a Borland C++ specific DLL to run (cc3260mt.dll). If you do not have Borland C++, simply recompile the code with your specific compiler.

Contents:

testdc.cpp	Main program for example use
sgp4unit.cpp (.h)	sgp4 mathematical theory
sgp4io.cpp (.h)	I/O routine for TLE data (twoline2rv conversion)
debug*.cpp	Debug routines to print out intermediate variables at the end of each function call
ast2body.cpp (.h)	Basic astrodynamics two-body routines – Kepler, etc.
asttime.cpp (.h)	Time routines
astmath.cpp (.h)	Basic math functions
astiod.cpp (.h)	Initial orbit determination routines and observation conversions – rv_razel, gibbs, etc.
eopspw.cpp (.h)	EOP and Space weather functions for the CelesTrak files – uses CelesTrak files (two provided)
coordfk5.cpp (.h)	Coordinate transformations for IAU-76/FK5 – uses nut80.dat