D-SPOSE Software User Manual

Luc B. M. Sagnières

Aerospace Mechatronics Laboratory
Department of Mechanical Engineering
McGill University

November 2020

This report provides information on the application of the Debris SPin/Orbit Simulation Environment (D-SPOSE v1.0.1). The software was developed by Luc Sagnières within the Aerospace Mechatronics Laboratory at McGill University, Montreal, QC, Canada, under the supervision of Prof. Inna Sharf.

1 Introduction

D-SPOSE was developed for space debris remediation purposes for the analysis and prediction of the rotational motion of large space debris in order to determine, to the highest degree of accuracy possible, the evolution of the rotational parameters of uncontrolled space objects over a time scale of years. This tool would benefit the space debris community, for example, by being able to predict the future attitude state of Active Debris Removal (ADR) targets, long before mission launch.

The developed software includes a coupled orbit-attitude propagator, simulating the evolution of the orbit and orientation of a satellite, and considers a widespread list of external gravitational and non-gravitational perturbations, including a high-order gravitational acceleration, the gravity-gradient torque, third-body perturbations from the Sun and the Moon, aerodynamic drag and torque, direct, Earth-emitted, and reflected radiation pressure and torque, the eddy-current torque, and internal energy dissipation. The developed model was tested and validated against past observations of the evolution of the angular motion of uncontrolled space objects, namely several spherical geodetic satellites, for which an abundant amount of observations exist.

For a defined spacecraft, with a tessellated surface geometry model and parameters, such as mass, moments of inertia, magnetic properties and surface optical coefficients, and initial conditions, *i.e.*, angular velocity, orientation, position, and velocity, all of which are inputted by the user, D-SPOSE will propagate the state of the satellite in question under the external perturbations requested by the user for the defined propagation length. It will output information about the state of the satellite throughout the simulation, as well as information about the perturbations.

Refer to the following conference papers, journal articles, and Ph.D. thesis (especially Chapter 2) for more information on the underlying dynamics:

- Sagnières, L. B. M. (2019), *Modeling and Simulation of Long-term Rotational Dynamics of Large Space Debris*, Ph.D. Thesis, McGill University
- Sagnières, L. B. M. and I. Sharf (2019), Long-term rotational motion analysis and comparison to observations of the inoperative Envisat, *J. Guid. Control Dyn.*, 42 (2), 364-376, doi:10.2514/1.G003647
- Sagnières, L. B. M., I. Sharf, and F. Deleflie, Validation of a novel coupled orbit-attitude propagator by comparison to SLR data and light curves, *69th International Astronautical Congress*, Abstract ID 46562, Bremmen, Germany, September 2018

- Sagnières, L. B. M., I. Sharf, and F. Deleflie (2020), Simulation of long-term rotational dynamics of large space debris: A TOPEX/Poseidon case study, *Adv. Space Res.*, 65, 1182-1195, doi:10.1016/j.asr.2019.11.021

The software and documentation can be found on the McGill University Aerospace Mechatronics Laboratory GitHub: `https://github.com/McGill-AML`. Any questions, bugs, or suggestions should be sent to luc.sagnieres@mail.mcgill.ca.

2 Installation

2.1 System Requirements

D-SPOSE was developed on macOS. It requires Fortran, C++, and C compilers. More information can be found in the README.md provided.

2.2 Package Contents

The package, which can be downloaded from the McGill University Aerospace Mechatronics Laboratory GitHub, contains the following:

- README.md: a readme file containing references
- makefile: a makefile for the command line compilation
- doc: a folder containing the documentation
  - D-SPOSE_software_manual.pdf: this software manual
  - Sagnieres_PhD_thesis.pdf: the PhD thesis mentioned above
  - Sagnieres_JGCD_2019.pdf: the JGCD journal article mentioned above
  - Sagnieres_IAC2018.pdf: the IAC2018 conference paper mentioned above
  - Sagnieres_ASR_2020.pdf: the ASR journal article mentioned above
- input: a folder for the user-defined simulation input files
- output: a folder where simulation output files are generated
- data: a folder containing all of the data input for environmental models
  - albedo_emissivite_CERES_9x9.txt: CERES albedo and IR coefficients
  - albedo_emissivity_ECMWF_9x9.txt: ECMWF albedo and IR coefficients
  - albedo_emissivity_stephens_9x9.txt: Stephens (1981) albedo and IR coefficients
  - ap_input.txt: Ap indices for NRLMSISE-00
  - DTCFILE.TXT: data file for JB2008
  - dwm07b104i.dat data file for HWM14
  - EGM2008_TideFree_Coefficients.txt: coefficients for EGM2008
  - eop.txt: Earth orientation parameters
  - gd2qd.dat: data file for HWM14
  - hwm123114.bin: data file for HWM14
  - igrf12coeffs.txt: coefficients for IGRF-12
  - moon_ephemeris.txt: Moon ephemerides
  - nut80.txt: 1980 IAU theory of nutation parameters

- solar_input.txt: F10.7 indices for NRLMSISE-00
- SOLFSMY.TXT: data file for JB2008
- sun_ephemeris.txt: Sun ephemerides
- wmm_coef.txt: coefficients for WMM

- src: a folder containing the source function and header files, seperated in subfolders dependent of language
  - aero_drag.c(.h): calculates the aerodynamic acceleration and torque for the entire surface model
  - aero_force.c(.h): calculates the force due to aerodynamic drag for a specific surface
  - aero_torque.c(.h): calculates the torque due to aerodynamic drag for a specific surface
  - albedo_calc.c(.h): calculates the acceleration and torque due to reflected and emmitted Earth radiation
  - angle2quat.c(.h): converts Euler angles into a quaternion
  - anglevelprop.c(.h): propagates angular velocity as a function of the sum of external torques
  - check_inputs.c(.h): calculates the lengths of the input files
  - crossprod.c(.h): performs a vector cross product
  - days2mdh.c(.h): converts the day of year to month, day of month, hour of day, minute of hour, and second of minute
  - dcm2angle.c(.h): converts a rotation matrix to Euler angles
  - dotprod.c(.h): performs a vector dot product
  - ecef2eci.c(.h): rotates a vector from the ECEF frame to the ECI frame
  - ecef2lla.c(.h): calculates the latitude, longitude, and altitude from a position vector in the ECEF frame
  - eddy_torque.c(.h): calculates the eddy-current torque
  - gaus_coef_wmm.c(.h): reformats the WMM coefficients for later use
  - gaus_coef.c(.h): reformats the IGRF-12 coefficients for later use
  - get_density.c(.h): outputs the atmospheric density at the satellite location using desired atmospheric model
  - grav_potential.c(.h): calculates the Earth gravitational potential energy
  - gravity_field.c(.h): calculates the acceleration due to the a non-spherical Earth
  - hwm14.f90: calculates the wind velocity according to HWM14
  - invertmat.c(.h): inverts a 3x3 matrix
  - load_inputs.c(.h): loads the information contained in the user-input files and data input files
  - load_teme.c(.h): loads the initial conditions and initializes orbit
  - JB2008.for: calculates density according to JB2008
  - magnetic_field_wmm.c(.h): calculates the magnetic field at satellite position according to WMM
  - magnetic_field.c(.h): calculates the magnetic field at satellite position according to IGRF-12
  - main.c: this is the main D-SPOSE script
  - matrixmult.c(.h): multiplies two 3x3 matrices
  - matxvec.c(.h): multiplies a 3x3 matrix by a 3x1 vector
  - moon_potential.c(.h): calculates the potential energy due to the Moon

- moon.c(.h): outputs Moon position at current time
- norm_coef.c(.h): reformats the EGM2008 coefficients for later use
- norm.c(.h): outputs the norm of a 3x1 vector
- nrlmsise-00_data.c: NRLMSISE-00 data
- nrlmsise-00.c(.h): calculates density according to NRLMSISE-00
- nutation.c(.h): outputs rotation matrix considering Earth nutation
- orbit2inertial.c(.h): converts position and velocity vectors to rotation matrix from SCO frame to inertial frame
- orbital2state.c(.h): converts orbital parameters to state vector
- polarm.c(.h): outputs rotation matrix considering polar motion
- precess.c(.h): outputs rotation matrix considering Earth precession
- propagation.c(.h): calculates the changes in state vector due to external perturbations
- quat2dcm.c(.h): converts a quaternion to a rotation matrix
- quatnormalize.c(.h): normalizes a quaternion
- sc_geometry.c(.h): loads the spacecraft geometry
- sc_parameters.c(.h): loads the spacecraft parameters
- SGP4.cpp(.h): contains the SGP4 procedure
- shadow_function.c(.h): calculates the portion of the sun not blocked by Earth
- sidereal.c(.h): outputs rotation matrix due to Earth's rotation
- srp_force.c(.h): calculates the force due to solar radiation for a specific surface
- srp.c(.h): calculates the solar radiation acceleration and torque for the entire surface model
- state2orbital.c(.h): converts state vector to orbital parameters
- sun_potential.c(.h): calculates the potential energy due to the Sun
- sun.c(.h): outputs Sun position at current time
- surface.h: defines the surface structure containing information about each surface
- t2doy.c(.h): converts seconds since January 1, 2000, 00:00:00 to year, day of year, and days since January 1, 2000
- teme2ecef.c(.h): converts position and velocity from TEME to ECEF
- third_body.c(.h): calculates the acceleration due to a third body
- tle2rv.cpp: this script converts the TLE input to initial position and velocity values using SGP4
- transpose.c(.h): calculates the transpose of a 3x3 matrix
- tt2utc.c(.h): converts terrestrial time to UTC
- vectors2angle.c(.h): calculates the minimum angle made from two vectors
- wind.c(.h): calculates the wind speed according to desired model

• matlab: a folder containing MATLAB functions for handling the output files
  - load_propagation.m: a script which will load the propagation output text file into a n x 14 array containing the 14 parameters of the state vector (time, velocity, position, angular velocity, orientation) for the n output time steps.
  - load_geometry.m: a script which will load the geometry output text file and plot the surface geometry model
  - state2coe.m: a function that converts position and velocity into orbial elements

2.3 Compilation

The compilation is done from the command line by executing the makefile (using command 'make test'), outputting two executables:

- tle2rv_exec: this converts the TLE input file to osculating initial conditions using SGP4
- dspose_exec: this runs the D-SPOSE simulation

This command above will also generate test output files associated with the sample input files provided. More information can be found in the README and in Section 3.5.

3 Operation

3.1 Input Files Format

Five user-input files are requested in order to run D-SPOSE. These contain information on the initial conditions, the dynamics model parameters, time information for the propagation, spacecraft parameters, and the spacecraft surface geometry model.

3.1.1 TLE.txt

This file contains a Two-Line Element in the standard format which will be used to initialize the orbit of the spacecraft and the initial epoch of the propagation using SGP4.

The sample 'model_parameters.txt' input file for the test scenario outlined below is the following:

```
%%%%% Input Two-Line Element for Propagation Initialization
% Initializes start time and orbital parameters through SGP4
1 27386U 02009A   13127.45641789  .00000089  00000-0  44087-4 0  3555
2 27386  98.4303 196.5510 0001204  87.6383 272.4946 14.37606880585289
```

3.1.2 time_parameters.txt

This file contains information on the time parameters for the propagation. The first line contains the requested propagation length: number of days, hours, minutes, and seconds.

The second contains the requested output step. This will decide how often the state vector is outputted to the propagation file.

The third line contains the time step for the propagation. The numerical method used is the fixed time step Runge-Kutta Dormand Prince numerical integration method, where the same time step is used for the orbital and attitude integration.

The sample 'model_parameters.txt' input file for the test scenario is the following:

```
%%%%% Time Parameters
% Line 1 is propagation length (Days / Hours / Minutes / Seconds)
% Line 2 is output step (Days / Hours / Minutes / Seconds)
% Line 3 is time step in seconds
0 0 0 10
0 0 0 1
0.1
```

3.1.3 model_parameters.txt

This file contains information on the model parameters. The user can decide to include (1) or exclude (0) any of the 14 perturbations listed in the first data line.

The second line deals with dynamics parameters and includes, in the following order: the largest degree and order considered in the spherical harmonics expansion of the geopotential in the calculation of the gravitational acceleration; the largest degree and order considered in the spherical harmonics expansion of the geopotential in the calculation of the gravity-gradient torque; the drag coefficient for the aerodynamic drag and torque calculation; the ap index; and the F10.7 index. The last two are for input into NRLMSISE-00 if that is the chosen atmospheric density model. If set as 0, then actual values for the time frame considered, included in the 'ap_input.txt' and 'solar_input.txt' data files, will be used; if not, then they will be considered as the constant user-input value for the entire propagation time frame.

In the third line, the user can decide which environmental model to use: the atmospheric density model (NRLMSISE-00 or JB2008); the magnetic field model (IGRF-12 or WMM); the wind model (co-rotating or HWM14); and the albedo model (Stephens (1981), CERES, or ECMWF).

The fourth line contains the parameters for the Kane Damper if internal energy dissipation is included: the moment of inertia considered diagonal with $I_x = I_y = I_z$, in kg m^2, and the damping coefficient, in kg m^2 s^{-1}.

In the final line, the user can decide if he wants the work parameters and perturbations output files outputted for the simulation (1=yes; 0=no).

The sample 'model_parameters.txt' input file for the test scenario is the following:

```
%%%%% Model Parameters
% Values: 1 to include; 0 to exclude
% Line 1: Aero-drag; Aero-torque; Grav-model; Grav-torque; Eddy-torque; Sun acc.;
    Moon acc.; SRP acc.; Albedo acc.; IR acc.; SRP torque; Albedo torque; IR torque; Kane
    damper
% Line 2: l_max for grav_a; l_max for grav_g; drag coefficient; Ap, F10.7 (for use with
    NRL; if 0 then use actual data)
% Line 3: Atmospheric density model: 1 for NRLMSISE-00; 2 for JB2008; Magnetic Field
    model: 1 for IGRF-12; 2 for WMM; Winds: 0 for co-rotating winds; 1 for HWM14; Albedo
    & IR: 1 for Stephens; 2 for CERES; 3 for ECMWF
% Line 4: damper moment of inertia (kg m^2); damping coefficient (kg m^2 s^-1)
% Line 5: Output work file if 1, no if 0; Output perturbations if 1, no if 0;
1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2.2 0 0
1 1 1 1
10 0.1
1 1
```

3.1.4 sc_parameters.txt

This file contains information about the spacecraft parameters and the initial rotational parameters. The first line contains the mass of the spacecraft, in kg, and the number of surfaces contained in the tessellated surface geometry model in 'sc_geometry.txt'.

The second line contains the initial angular velocity of the body frame with respect to the inertial frame, in $^\circ$ s^{-1}, with components in the body-fixed frame as defined by the surface geometry model.

The third line contains the initial orientation of the body-fixed frame, parametrized as Euler angles, in $^\circ$, in a 3-2-1 rotation. The last parameter on this line indicates whether this orientation is with respect to the inertial frame (set as 1) or the orbital frame (set as 2). The orbital frame is defined with the x_{sco}-axis in the spacecraft velocity direction, the z_{sco}-axis pointing towards Earth in the orbital plane, and the y_{sco}-axis completing the right hand rule.

The fourth to sixth lines contain the matrix representation of the inertia tensor in the body-fixed frame, in kg m^2. The seventh to ninth lines contain the matrix representation of the magnetic tensor in the body-fixed frame, in S m^4.

The sample 'sc_parameters.txt' input file for the test scenario is the following:

```
%%%%% Spacecraft Parameters
% Line 1 is m (kg), number of surfaces
% Line 2 is initial angular velocity of body frame w.r.t. inertial frame in body frame
    (° s^-1)
% Line 3 is initial attitude of body frame (Euler angles with 3-2-1 rotation) (°)  w.r.t.
    (1) inertial frame or (2) orbital frame
% Lines 4-6 are the inertia matrix (kg m^2)
% Lines 7-9 are the magnetic tensor (S m^4)
7827.867 16
0 0 2.9
180 0 81.5 2
17023.3 397.1 -2171.4
397.1 124825.7 344.2
-2171.4 344.2 129112.2
0931500 0 0
0 1059000 0
0 0 1059000
```

3.1.5 sc_geometry.txt

This file contains the tessellated surface geometry model as well as information on the optical coefficients of each surface. Each line contains the information required for one triangular surface. The first three columns are the inward surface normal of the surface, $\mathbf{n} = [n_x\ n_y\ n_z]^T$, in the body-fixed frame.

Then, each set of three columns, 4-6, 7-9, 10-12, represent the position of the three vertices in the body-fixed frame, in m, $\mathbf{r}_1 = [x_1\ y_1\ z_1]^T$, $\mathbf{r}_2 = [x_2\ y_2\ z_2]^T$, $\mathbf{r}_3 = [x_3\ y_3\ z_3]^T$.

Finally, the last columns are the coefficients of diffuse reflection, specular reflection, and absorption in the visible and infrared spectra, respectively, $\sigma_{rd,\mathrm{alb}}$, $\sigma_{rs,\mathrm{alb}}$, $\sigma_{a,\mathrm{alb}}$, $\sigma_{rd,\mathrm{ir}}$, $\sigma_{rs,\mathrm{ir}}$, and $\sigma_{a,\mathrm{ir}}$.

The sample 'sc_geometry.txt' input file for the test scenario is not shown here due to its size, but the file header is as follows:

```
%%%%% Spacecraft Geometry
% Each row is one triangular surface
% Columns 1-3 are inward surface normal
% Columns 4-6, 7-9, 10-12 are coordinates of the vertices in body-fixed frame (m)
% Columns 13-18 are the coefficients of diffuse reflection, specular reflection, and
    absorption in (1) visible and (2) infrared spectrum
```

3.2 Execution

D-SPOSE is run from the command line. First, the TLE input needs to be converted to osculating initial conditions by running the 'tle2rv_exec' executable file. This reads the 'TLE.txt' input file and outputs the 'initial_conditions.txt' file. Second, the D-SPOSE simulation is started by executing 'dspose_exec'. This reads the input files 'model_parameters.txt', 'sc_parameters.txt', 'time_parameters.txt', 'sc_geometry.txt', and 'initial_conditions.txt', and outputs the propagation output file containing the user-input information and simulation output, the geometry file for housekeeping, and the work parameters and perturbations files if requested.

3.3 Output Files Format

3.3.1 propagation_v1.0.0_YYYY-MM-DD_HH-MM-SS.txt

The propagation output file contains information on the state vector throughout the simulation, but also contains more information in its header. The input parameters are reproduced, and a rotation matrix to rotate a vector from the inertial frame used for the propagation, the TEME frame at the input TLE epoch, to the inertial J2000 frame.

The following is the header for the test scenario propagation output file:

```
# VERSION NUMBER: 1.0.0
# START TIME: Tue Dec  4 10:34:22 2018
#
# TIME PARAMETERS: 4875.000000 10.000000 57.000000 81.587031 0.000000 0.000000 0.000000
    10.000000 0.000000 0.000000 0.000000 1.000000 0.100000
# SPACECRAFT PARAMETERS: 7827.867000 16.000000 0.001243 7150732.992181 196.551000
    98.424919 68.491653 291.508762 0.000000 0.000000 2.900000 180.000000 0.000000
    81.500000 2.000000 17023.300000 397.100000 -2171.400000 397.100000 124825.700000
    344.200000 -2171.400000 344.200000 129112.200000 931500.000000 0.000000 0.000000
    0.000000 1059000.000000 0.000000 0.000000 0.000000 1059000.000000
# MODEL PARAMETERS: 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
    1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 2.000000
    2.000000 2.200000 0.000000 0.000000 2.000000 1.000000 1.000000 1.000000
    10.000000 0.100000 1.000000 1.000000
#
# Rotation Matrix from TEME to ECI (J2000) at propagation start:
# 0.9999946757676250 0.0029847622663145 0.0013189505737857
# -0.0029847221839813 0.9999955451833360 -0.0000323568752460
# -0.0013190412756830 0.0000284200019334 0.9999991296608296
#
# Time since Start (s) Velocity Components in TEME [v_i v_j v_k] (m/s) Position
    Components in TEME [r_i r_j r_k] (m) Angular Velocity Components in Body Frame
    [w_x w_y w_z] (rad/s) Orientation Quaternion [q_0 q_1 q_2 q_3]
```

3.3.2 perturbations_v1.0.0_YYYY-MM-DD_HH-MM-SS.txt

In the perturbations file, the 14 external perturbations are evaluated at each output time step. Components of the accelerations are produced in the inertial frame while components of the torques are produced in the body-fixed frame.

```
# Perturbations Output File
# Accelerations in ECI; Torques in Body-Fixed Frame
# Time since Start (s) Aerodynamic Acceleration (m s-2) Aerodynamic Torque (N m)
    Gravitational Acceleration (extra terms) (m s-2) Gravity-Gradient Torque (N m)
    Eddy-Current Torque (N m) Sun Acceleration (m s-2) Moon Acceleration (m s-2)
    Solar Radiation Acceleration (m s-2) Albedo Acceleration (m s-2) IR Acceleration
    (m s-2) Solar Radiation Torque (N m) Albedo Torque (N m) IR Torque (N m) Kane Damper
    Torque (N m)
```

3.3.3 parameters_v1.0.0_YYYY-MM-DD_HH-MM-SS.txt

The header for the parameters file is as follows. This file contains information on the energy of the spacecraft and on the work done by the external perturbations.

```
# Time since Start (s) Non-Conservative Translational Work (J) Non-Conservative
    Rotational Work (J) Work from Gravity-Gradient Torque (J) Work from Aspherical Earth
    Potential (J) Work from Sun Acceleration (J) Work from Moon Acceleration (J) Earth
    Potential Energy (J) Sun Potential (J) Moon Potential (J)
```

3.3.4 geometry_v1.0.0_YYYY-MM-DD_HH-MM-SS.txt

The same geometry file as used as input is outputted for housekeeping purposes.

3.4 Software Limits

The are some internal limits to the software, which are outlined here along with some recommendations:

- The earliest epoch for a TLE to be used as initial conditions is January 1, 2000. Any TLE before this will produce an error message.
- Earth Orientation Parameters are included until September 26, 2017. More recent values can be appended as shown in Section 3.6. These are currently only used for the calculation of the exact rotation matrix between the TEME frame of the initial TLE epoch used as inertial frame in D-SPOSE and the J2000 frame. If a TLE is used as initial conditions with an epoch after the date of the latest EOP parameters, an error message will occur.
- JB2008 contains a hard lower limit on its density output of 90 km. Anything below will produce a "nan" value for atmospheric density. No hard upper limit exists but a recommendation of 4000 km is put forward. No hard limits exist for NRLMSISE-00.
- Actual solar and geomagnetic indices for NRLMSISE-00 are included until May 31, 2017. More recent values can be appended as shown in Section 3.6.
- Actual solar and geomagnetic indices for JB2008 are included until August 1, 2017. The input file can be updated as shown in Section 3.6.
- Sun and Moon ephemerides are included until early 2030. More recent values can be appended as shown in Section 3.6.
- Error messages will appear if an incorrect input value is present in the input files.
- For long propagation periods (years) and small time steps (seconds), the software can take a very long time to run (days). It is recommended to run D-SPOSE on a cluster.
- If the chosen time step is not small enough (recommended to be on the order of seconds), results can easily be flawed. It is recommended that the user performs a work-energy balance of the results using the information contained in the propagation output file to calculate the energy of the system and the information in the work output file to calculate the work done by the perturbations.
- Two other torques haven't been included yet: the residual magnetic torque and the YORP effect. These should be added in future updates.

3.5 Test Scenario

During the compilation using the makefile ('make test' command), the software will also be tested. Using the sample input files located in the "input/" folder, a short simulation will be run, outputting files in the "output/" folder. The user can check whether D-SPOSE is running correctly by comparing the results obtained to the results provided in the "output/expected-output" folder, which should be the same.

3.6 Updating Data Files

3.6.1 solar_input.txt

The file containing the F10.7 and Ap indices can be found through the following links: ftp://ftpsedr.cls.fr/pub/previsol/solarflux/observation/radio_flux_absolute_observation.txt and ftp://ftp.ngdc.noaa.gov/STP/GEOMAGNETIC_DATA/INDICES/KP_AP. From these, for every row containing information for one day, the appropriate F10.7 of the previous day, the F10.7 81-day average centred on current day, and the daily Ap value can be appended. The latest values in the "solar_input.txt" file might not correspond to the same values as indicated on the website for the same date as these values are corrected when more accurate information is obtained.

3.6.2 ap_input.txt

The file containing 3-hour Ap indices can be found through the following link: ftp://ftp.ngdc.noaa.gov/STP/GEOMAGNETIC_DATA/INDICES/KP_AP. From these, for every row containing information for one day, the appropriate 3-hour Ap value can be appended. The latest values in the "ap_input.txt" file might not correspond to the same values as indicated on the website for the same date as these values are corrected when more accurate information is obtained.

3.6.3 sun_ephemeris.txt

The Sun ephemerides can be obtained form the Miriade Ephemeris Generator of the IMCCE Virtual Observatory Solar System Portal: http://vo.imcce.fr/webservices/miriade/?forms. The target can be selected as "Sun". The epoch and number of time steps is inputted by the user, in Terrestrial Time (TT) as used in D-SPOSE. The time step needs to be set as 12-hour increments. The reference center needs to be "geocenter". The planetary theory is "INPOP". The reference plane is the "equator". The type of coordinates is "rectangular". The type of ephemeris is "Mean of date". From the generated output, the year, month, day, hour, and components of the Sun position in AU needs to be added to the columns in the "sun_ephemeris.txt" file.

The latest values in the "sun_ephemeris.txt" file might not correspond to the same values as indicated on the website for the same date as these values are corrected when more accurate information is obtained.

3.6.4 moon_ephemeris.txt

The Moon ephemerides can be obtained form the Miriade Ephemeris Generator of the IMCCE Virtual Observatory Solar System Portal: http://vo.imcce.fr/webservices/miriade/?forms. The target can be selected as "Moon". The epoch and number of time steps is inputted by the user, in Terrestrial Time (TT) as used in D-SPOSE. The time step needs to be set as 4-hour increments. The reference center needs to be "geocenter". The planetary theory is "INPOP". The reference plane is the "equator". The type of coordinates is "rectangular". The type of ephemeris is "Mean of date". From the generated output, the year, month, day, hour, and components of the Sun position in AU needs to be added to the columns in the "moon_ephemeris.txt" file.

The latest values in the "moon_ephemeris.txt" file might not correspond to the same values as indicated on the website for the same date as these values are corrected when more accurate information is obtained.

3.6.5 eop.txt

The Earth orientation parameters can be obtained from the website of the IERS (https://www.iers.org/IERS/EN/DataProducts/EarthOrientationData/eop.html). In the "Standard EOP data files" section, under "Rapid data and predictions", you can click on "latest version" associated with the "finals.data (IAU1980)" header. This will generate a text file containing the Earth Orientation Parameters needed. The current "eop.txt" text file can then be appended using the first four columns of the text in the web page containing the year, month, day, and modified Julian date. Then, in order, the columns contain the x and

y components of the polar motion, UT1-UTC, LOD, dPsi, and dEps. Each is followed by an error estimate on the web page that isn't included in "eop.txt". Keep in mind that LOD is is milliseconds in the online text file while it needs to be included in seconds in "eop.txt".

The latest values in the "eop.txt" file might not correspond to the same values as indicated on the website for the same date as these values are corrected when more accurate information is obtained.

3.6.6 SOLFSMY.TXT

This file can be downloaded directly from the JB2008 website (`http://sol.spacenvironment.net/jb2008/indices/SOLFSMY.TXT`) and updated in the "data/" folder. The latest values in the "SOLFSMY.TXT" file might not correspond to the same values as indicated on the website for the same date as these values are corrected when more accurate information is obtained.

3.6.7 DTCFILE.TXT

This file can be downloaded directly from the JB2008 website (`http://sol.spacenvironment.net/jb2008/indices/DTCFILE.TXT`) and updated in the "data/" folder. The latest values in the "DTCFILE.TXT" file might not correspond to the same values as indicated on the website for the same date as these values are corrected when more accurate information is obtained.