# Neutral Density Uncertainty Quantification Software Tool Manual

*Version 1, 17 September 2023*

*Author: C. Siemes, Delft University of Technology, c.siemes@tudelft.nl*

## Introduction

Accelerometer and GNSS receiver data allow for retrieving neutral mass density observations. In the case of accelerometers, the observations represent the in situ density, whereas density observations derived from GNSS receiver data require averaging along the orbit. The accuracy of the density observations is difficult to assess because it depends on many input parameters describing the satellite geometry and surface properties, atmospheric conditions, radiation fluxes, and the accuracy of the accelerometer and GNSS receiver data. We developed a software tool that propagates the uncertainty from input parameters to density observations. Optionally, the software accounts for averaging along the orbit.

The software tool performs the following steps. First, it simulates the desired orbit and satellite attitude. Then, it evaluates the radiation pressure and aerodynamic force along the orbit based on radiation flux data and thermosphere models and derives the density (Doornbos, 2011). As part of the radiation pressure evaluation, the software tool calculates the temperature of the satellite surfaces (Siemes et al., 2023). The temperature, acceleration, and density signals are the Taylor point in linearizing the nonlinear functions relating the input parameters to density. After the linearization, the software tool performs the formal error propagation and generates figures illustrating the results. A detailed description of the method will be provided in a scientific publication. That description will clarify the meaning of the parameters mentioned in the remainder of the manual.

## License

**External software**

This software uses several external source codes and libraries, which are subject to their own licenses. These source codes and libraries are the following:

1) SOFA library

This software uses the SOFA library (contained in the subdirectory external/SOFA_library) for the conversion of time values and for constructing the transformation from the celestial to the terrestrial reference frame following the IERS conventions. The SOFA library is available under license, where the license text is included in its source code files. This software implements an interface (contained in the subdirectory external/SOFA_interface) to the SOFA library via Matlab's mex functionality, which was not provided or endorsed by SOFA. Acknowledgement:

Software Routines from the IAU SOFA Collection were used. Copyright (c) International Astronomical Union Standards of Fundamental Astronomy (http://www.iausofa.org)

2) NRLMSISE-00 C code implementation by Dominik Brodowski

Dominik Brodowski created a C code implementation of the NRLMSISE-00 model, which is available in the public domain and is contained in the subdirectory external/NRLMSISE-00. The C code can be accessed here: https://www.brodo.de/space/nrlmsise/ (last accessed on 7 January 2024)

3) The "interp.f" source code by Ch. Bizouard, Observatoire de Paris

The function implements the interpolation of Earth orientation parameters from the IERS (contained in the subdirectory external/IERS_interface). It can be accessed here: ftp://hpiers.obspm.fr/iers/models/interp.f (last accessed on 7 January 2024)

**Installation**

The software tool was developed using Matlab release 2023a without toolboxes. Though not tested, we expect the software tool to work on other recent Matlab releases. Usage of external source code has been kept to a minimum. The only external source code used is the code for the NRLMSISE-00 model, provided by Dr. Dominik Brodowski, who translated the original Fortran code into C code, noting that C code is easier to integrate into Matlab than Fortran code. More information is provided on the webpage: http://www.brodo.de/english/pub/nrlmsise/

Follow the instructions of the SOFA library in the subdirectory http://www.iausofa.org to compile the SOFA library contained in the subdirectory external/SOFA_library. Typically, you need to edit the ‘makefile’ in external/SOFA_library/20190722/c/src and then run ‘make’ on the shell.

Several files were to be downloaded from the webpage https://github.com/c0d3runn3r/nrlmsise and placed into the subdirectory 'NRLMSISE-00':

- nrlmsise-00.h
- nrlmsise-00.cpp
- nrlmsise-00_data.cpp

Open Matlab, switch to the subdirectory 'external/NRLMSISE-00', and run the 'compile_nrlmsise00_mex_wrapper.m' to compile the source code.

After downloading and compiling the source code of the NRLMSISE-00 model, the software tool can be run by entering the command 'simulator' in the command window. The simulations are modified by editing the settings file 'simulator_settings.m' as explained in the following sections. The result of the simulation is a set of figures that are stored in the directory specified in the settings file.

Similarly, switch to subdirectories 'external/IERS_interface' and 'SOFA_interface' and use the provided Matlab scripts to compile the external code.

**Simulation settings**

The simulation settings are specified in the file 'simulator_settings.m'. The user needs to edit this file to modify the simulations. Listing 1 gives an overview of the high-level simulation parameters. The first simulation parameter sets the length of the time series in units of days. We recommend simulating at least four orbits, i.e., one-fourth of a day (0.25 days). The time step specifies the time between two successive samples in the time series. Since the satellite's temperature is calculated from one sample to the next based on a discretized differential equation, the time step must be carefully selected. A small time step will result in a long computation time, whereas a large time step will result in inaccurate temperatures. We recommend using a time step of 5 – 60 s. Instead of using simulated satellite orbit, attitude, and mass data, one can use specify an input file with real satellite data. Finally, the user can specify the output directory where the figures illustrating the simulation results will be stored.

```
SIM.period = 0.2;                % period = length of the simulated time series (d)
SIM.time_step = 60;              % time step between two samples (s)
SIM.consider_averaging = true;   % switch that controls whether average density
                                 % will be simulated (true or false)
SIM.use_real_data = true;        % switch for using real or simulated data
DATA.INPUT_FILE = './my_input_data/GA_20031101.txt'; % input file (optional)
SIM.OUTPUT_DIR = './my_output';  % directory where figures will be stored
```

*Listing 1: Parameters specifying the simulation settings.*

**Orbit**

Orbit is specified by the altitudes of the perigee and apogee, the inclination, the eccentricity, the longitude of the ascending node, the argument of perigee, and the true anomaly (cf. Listing 3). The meaning of these orbital parameters is sketched in Figure 1. We refer to Montenbruck and Gill (2012), Chapter 2 for a more detailed description of the orbital parameters. Note that we define the altitudes of the perigee and apogee as the distance from the surface of a sphere with radius R = 6,378,136.6 m to the perigee and apogee.

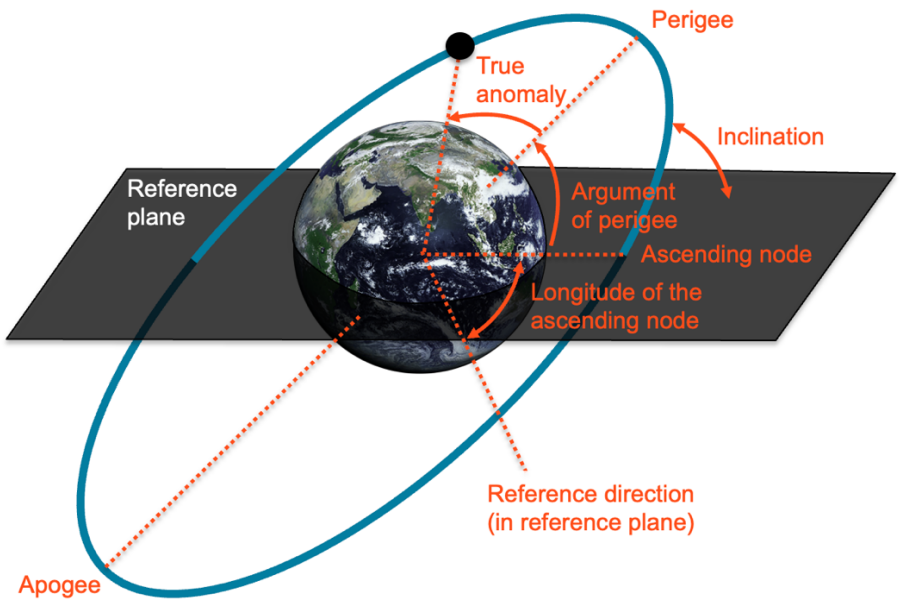


*Figure 1: Parameters specifying the initial state of the orbit.*

The orbital parameters specified in the settings (cf. Listing 2) describe the initial state, equivalent to the initial position and velocity. In the simulations, we account for the secular drifts in the longitude of the ascending node and the argument of perigee due to the J2 effect. Atmospheric drag is not accounted for in the simulations. Thus, if different altitude regimes need to be explored, the user must manually specify those manually in the settings.

```
ORBIT.altitude_at_apogee = 400e3;      % apogee (m)
ORBIT.altitude_at_perigee = 400e3;     % perigee (m)
ORBIT.inclination = 89;                % inclination (°)
ORBIT.argument_of_perigee = 0;         % argument of perigee (°)
ORBIT.longitude_of_ascending_node = 0; % longitude of the ascending node (°)
ORBIT.true_anomaly = 0;                % true anomaly (°)
```

*Listing 2: Parameters specifying the orbit.*

**Satellite attitude**

The satellite attitude is defined by the attitude control law. In the simulator, we implemented the most common attitude control law for satellites carrying accelerometers, where the satellite body reference frame is defined as follows:

- The x-axis is pointing in the direction of the satellite's velocity relative to the atmosphere, where the relative velocity is composed of the satellite velocity and the velocity of the corotating atmosphere.
- The y-axis is perpendicular to the relative velocity and the satellite's position vector.
- The z-axis is perpendicular to the x- and y-axis.

**Satellite model**

The satellite is represented by a panel model, where the panels define the outer geometry of the satellite. Each panel is augmented with parameters that describe its properties (cf. Listing 3): area, outward-pointing normal vector (a unit vector perpendicular to the panel), material ID, temperature, coefficients of absorption, diffuse, and specular reflection of visible light and infrared radiation, heat conductivity to the satellite body, and heat capacitance. The material IDs identify which panels have the material and, therefore, the same absorption and reflection coefficients. This information is necessary to account for situations where radiation pressure errors cancel. In addition, we specify the energy accommodation coefficient, which applies to all panels, the satellite body's temperature, heat capacitance, heat generation, and the satellite mass.

The satellite geometry is defined by the panels' areas and normal (cf. Figure 2). The satellite geometry and mass are the basis for all subsequent modeling.

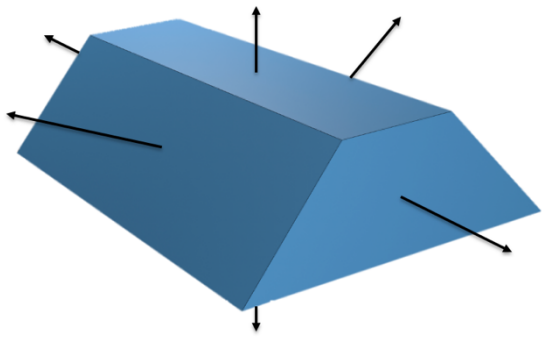

*Figure 2: Panel model of the GRACE satellites. The blue faces represent the panels, and the black arrows indicate the panels' normal vectors.*

We can subdivide the satellite model into three components:

1. Surface properties for modeling the radiation pressure from external radiation sources

   External radiation sources are solar radiation, albedo, and Earth infrared radiation (cf. Figure 3). The absorption and reflection coefficients represent the surface properties related to radiation pressure from external radiation sources. They define the fraction of radiation that is absorbed, diffusely reflected, and specularly reflected, separately for visible light and infrared radiation. An intuitive description is available in Doornbos (2011).

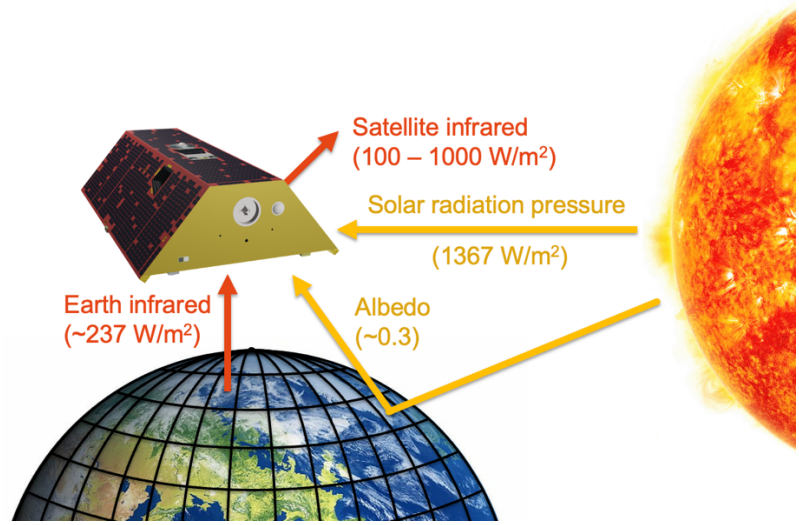


*Figure 3: Sources of radiation pressure.*

2. Thermal satellite model

   The thermal satellite model is used for calculating the panel temperatures, from which the software tool calculates the radiation pressure due to the satellite's thermal emissions according to the Stefan Boltzmann law. The panels gain heat by absorbing radiation and lose heat by emitting heat into cold space (cf. Figure 4). The heat capacitance controls how fast the panels gain and lose heat. The satellite body conductively exchanges heat with the panels and generates its own heat through the onboard electrical systems. For the acceleration due to thermal emissions and its uncertainty, it is most important that the thermal model replicates the maximum temperatures well.

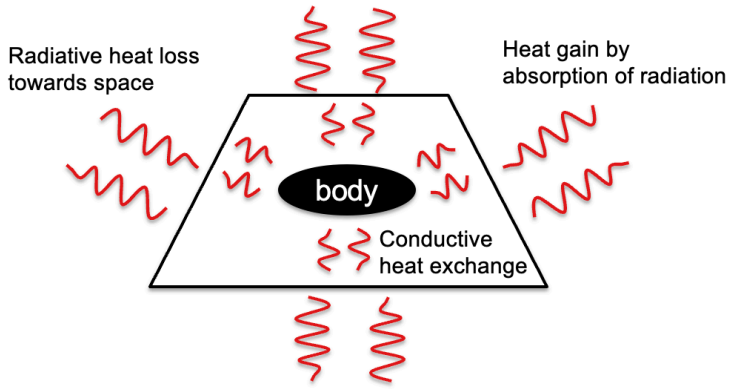


*Figure 4: Thermal satellite model.*

3. Gas-surface interaction model

We use the gas-surface interaction model based on diffuse reflections with incomplete accommodation (DRIA) based on Sentman (1961). As the acronym suggests, incident particles are reemitted fully diffusely (cf. Figure 5), where the angular probability distribution follows Lambert's cosine law. The energy accommodation coefficient specifies the velocity of the reemitted particles. An energy accommodation coefficient of zero means that the reemitted velocity is the same as the incident velocity. In contrast, an energy accommodation coefficient of one means that the velocity of the reemitted particle corresponds to the satellite surface temperature, which is usually much lower than the incident velocity. At 200 – 400 km altitudes, we expect the energy accommodation coefficient to be close to one; above 400 km altitude, it will be smaller than one (Bernstein and Pilinski, 2022).

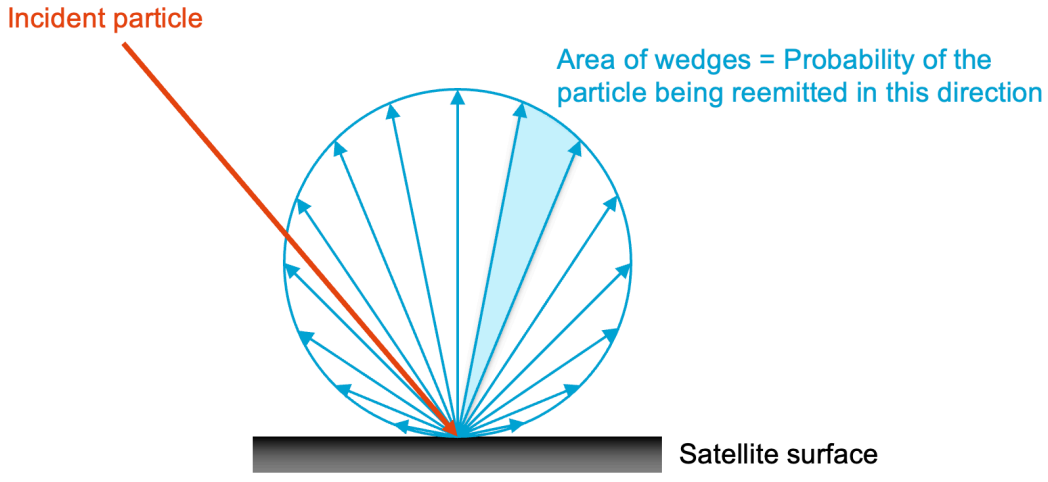


*Figure 5: Diffuse reemission of particles on the satellite surface.*

```
% panel normal vectors (dimension = #panels x 3, dimensionless quantity)
SAT.np = [1.0  0.0       0.0      % front
         -1.0  0.0       0.0      % rear
          0.0  0.766044 -0.642787 % starboard outer
          0.0 -0.766044 -0.642787 % port outer
          0.0  0.0       1.0      % nadir
          0.0  0.0      -1.0];    % zenith

% panel material IDs (dimension = #panels x 1, dimensionless quantity)
SAT.mp = [1; 1; 2; 2; 3; 2];

% panel areas (dimension = #panels x 1, unit = m2)
SAT.Ap = [0.9551567; 0.9551567; 3.1554792; 3.1554792; 6.0711120; 2.1673620];

% energy accommodation coefficient of DRIA model (dimensionless quantity)
SAT.alpha_E = 0.85;

% initial temperature of panels (dimension = 1 x #panels, unit = K)
SAT.Tw = [300,300,300,300,300,300];

% initial satellite body temperature (unit = K)
SAT.Tb = 297;

% reflection coefficients for visible light
% (dimension = #panels x 1, dimensionless quantity)
SAT.ca_vis = [0.34; 0.34; 0.65; 0.65; 0.12; 0.65]; % absorption
SAT.cs_vis = [0.40; 0.40; 0.05; 0.05; 0.68; 0.05]; % specular reflection
SAT.cd_vis = 1 - SAT.ca_vis - SAT.cs_vis;          % diffuse reflection

% reflection coefficients for infrared light
% (dimension = #panels x 1, dimensionless quantity)
SAT.ca_ir = [0.62; 0.62; 0.81; 0.81; 0.75; 0.81]; % absorption
SAT.cs_ir = [0.23; 0.23; 0.03; 0.03; 0.19; 0.03]; % specular reflection
SAT.cd_ir = 1 - SAT.ca_ir - SAT.cs_ir;            % diffuse reflection

% heat conductivity of panels toward satellite body
% (dimension = #panels x 1, unit = W/K)
SAT.kp = [0.1; 0.1; 0.1; 0.1; 0.5; 0.1];

% heat capacitance of panels
% (dimension = #panels x 1, unit = J/K)
SAT.Cp = [1000; 1000; 5000; 5000; 10000; 5000];

% heat capacitance of satellite body (unit = J/K)
SAT.Cb = 1e5;

% heat generation inside satellite body (unit = W)
SAT.Pb = 70;

% satellite mass (kg)
SAT.ms = 476.8471;
```

*Listing 3: Parameters specifying the satellite properties.*

**Uncertainty parameters related to the radiation pressure**

We consider the uncertainty in the input parameters of the satellite model and the radiation fluxes (cf. Listing 4). The uncertainty of the initial temperatures refers to the temperature at the first epoch. The temperatures are updated after every time step, and the uncertainty of the initial temperatures is propagated accordingly. This means that the uncertainty of the temperatures at a later epoch will be different from the one specified in the settings. Typically, the initial temperatures and their uncertainty will have an insignificant effect after several orbits (unless unrealistic initial values are specified). Therefore, we recommend simulating at least four orbits, to overcome the dependence on the initial temperatures.

Shadowing effects will not play a role if the satellite has a convex shape. Shadowing effects might affect the radiation pressure and aerodynamic force if the satellite has a concave shape or appendices like deployable solar arrays. We recommend increasing the uncertainty of the area of the affected panels to account for the uncertainty induced by shadowing effects.

Contrary to many other parameters, the uncertainty in the radiation fluxes is specified as a fraction of the signal. This ensures that the radiation pressure from distant albedo or Earth infrared radiation pixels has a minor influence on the uncertainty while keeping the calculations simple.

```
% standard deviation of panel heat conductivity toward satellite body
% (dimension = #panels x 1, unit = W/K)
RP_COV_INPUTS.kp_std = SAT.kp/4;

% standard deviation of panel heat capacitance
% (dimension = #panels x 1, unit = J/K)
RP_COV_INPUTS.Cp_std = SAT.Cp/4;

% standard deviation of satellite body heat capacitance (unit = J/K)
RP_COV_INPUTS.Cb_std = SAT.Cb/4;

% standard deviation of heat capacitance of satellite body (unit = J/K)
RP_COV_INPUTS.Pb_std = SAT.Pb/4;

% standard deviation of panel areas (m2)
RP_COV_INPUTS.Ap_std = SAT.Ap/50;

% standard deviation of satellite mass (kg)
RP_COV_INPUTS.ms_std = 2;

% standard deviation of reflection coefficients for visible light per material
% (must be specified in ascending order of material IDs)
% (dimension = #materials x 1, dimensionless quantity)
RP_COV_INPUTS.ca_vis_std = [0.1; 0.1; 0.1]; % absorption
RP_COV_INPUTS.cd_vis_std = [0.1; 0.1; 0.1]; % specular reflection
RP_COV_INPUTS.cs_vis_std = [0.1; 0.1; 0.1]; % diffuse reflection

% standard deviation of reflection coefficients for infrared radiation per material
% (must be specified in ascending order of material IDs)
% (dimension = #materials x 1, dimensionless quantity)
RP_COV_INPUTS.ca_ir_std = [0.1; 0.1; 0.1]; % absorption
RP_COV_INPUTS.cd_ir_std = [0.1; 0.1; 0.1]; % specular reflection
RP_COV_INPUTS.cs_ir_std = [0.1; 0.1; 0.1]; % diffuse reflection

% standard deviation of initial panel temperatures
% (dimension = 1 x #panels, unit = K)
RP_COV_INPUTS.T_std = [20 20 20 20 20 20];

% standard deviation of initial satellite body temperature (K)
RP_COV_INPUTS.Tb_std = 20;

% standard deviation of radiation fluxes as a fraction of the signal
% (dimensionless quantity)
RP_COV_INPUTS.Ps_fraction = 0.001; % solar radiation
RP_COV_INPUTS.Pa_fraction = 0.1;   % albedo (value applies to all pixels)
RP_COV_INPUTS.Pi_fraction = 0.1;   % Earth infrared radiation (idem)
```

*Listing 4: Parameters specifying the uncertainty of radiation pressure input parameters.*

**Uncertainty parameters related to the aerodynamic force**

We consider the uncertainty of the following input parameters for the aerodynamic force: atmospheric temperature, mass fraction of atmospheric constituents, satellite velocity relative to the atmosphere, energy accommodation coefficient, panel areas, and satellite mass (cf. Listing 5). The satellite velocity relative to the atmosphere is the combined effect of the uncertainty in the satellite velocity and thermosphere wind. Unless a piece of debris is considered, the latter will usually be much larger than the first. Further, we note that the panel area's uncertainty usually equals the uncertainty value used for radiation pressure. Since we use the NRLMSISE-00 model to represent the atmosphere (Picone et al., 2002), the atmospheric constituents are hydron (H), helium (He), atomic nitrogen (N), atomic oxygen (O), diatomic nitrogen ($N_2$), diatomic oxygen ($O_2$), Argon (Ar), and atomic oxygen ions ($O^+$).

```
% standard deviation of atmospheric temperature as a fraction of the signal
% (dimensionless quantity)
AERO_COV_INPUTS.Ta_fraction = 0.2;

% standard deviation of atmospheric mass density of constituents as a fraction
% of the signal (dimension = #constituents x 1, dimensionless quantity,
% sequence is He, O, N2, O2, Ar, H, N, O+)
AERO_COV_INPUTS.rho_fraction = [0.2; 0.2; 0.2; 0.2; 0.2; 0.2; 0.2; 0.2];

% standard deviation of energy accommodation coefficient
% (dimensionless quantity)
AERO_COV_INPUTS.alpha_E = 0.05;

% standard deviation of panel areas (dimension = #panels x 1, unit = m2)
AERO_COV_INPUTS.Ap_std = RP_COV_INPUTS.Ap_std;

% standard deviation of satellite mass (unit = kg)
AERO_COV_INPUTS.ms_std = RP_COV_INPUTS.ms_std;

% standard deviation of satellite velocity relative to atmosphere
% (dimension = 3 x 1, unit = m/s, satellite body reference frame)
AERO_COV_INPUTS.vr_std = [50; 50; 20];
```

*Listing 5: Parameters specifying the uncertainty in the aerodynamics input parameters.*

**Uncertainty parameters related to accelerometry and GNSS tracking**

We consider accelerometer noise and the accuracy of the GNSS tracking. The accelerometer noise is the instrument's precision, typically best in the 1 – 100 mHz frequency range. The GNSS tracking noise will propagate to acceleration noise conceptually via double-differencing of positions and evaluating a gravity field model at a noisy position. GNSS tracking noise is typically at m-level for onboard navigation solutions or when only single-frequency or code observations are available. A cm-level GNSS tracking accuracy can only be achieved when multi-frequency carrier phase observations are processed with precise orbit determination software on the ground.

The following parameters specify the accelerometer and GNSS tracking noise (cf. Listing 6): The standard deviation of the accelerometer noise, separately for the x-, y-, and z-components of the satellite body reference frame, the standard deviation of the x-, y-, and z-components of the GNSS-derived position in the local orbit reference frame, including the correlations between the components, the period used for estimating the accelerometer bias (typically 1 day), the averaging period for GNSS-derived density observations, and the slope of the GNSS-tracking noise amplitude spectral density.

The slope parameter accounts for time correlations in the GNSS tracking noise. A slope of 0.0 corresponds to uncorrelated positions noise (white noise), whereas a slope parameter smaller than 0.0 increases the noise at low frequencies, as illustrated in Figure 6. This should be considered in the context of averaging, effectively resulting in a noise cut-off at the frequency corresponding to the averaging period. The noise variance after double-differencing and averaging is equal to the area under the acceleration noise power spectral density, i.e., to the square of the area highlighted in Figure 6 in color.

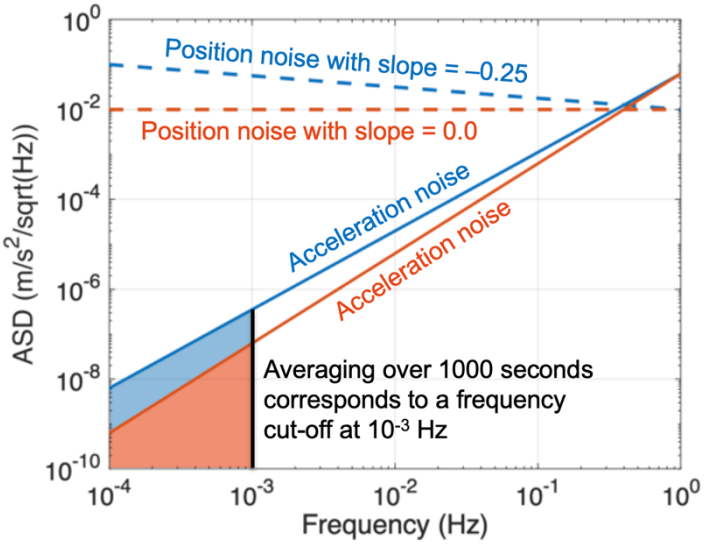


*Figure 6: Slope of GNSS position noise accounting for time-correlations. The acceleration noise conceptually resulting from double-differencing and averaging will have a variance proportional to the square of the colored area.*

```
% standard deviation (precision) of accelerometer measurements
% (dimension = 3 x 1, unit = m/s2, satellite body reference frame)
ACC.acc_std = [1e-9; 1e-9; 1e-9];

% standard deviation of position noise
% (unit = m, local orbit reference frame)
ACC.sx = 0.01; % along-track direction
ACC.sy = 0.01; % cross-track direction
ACC.sz = 0.05; % zenith direction

% correlation between components of position noise (dimensionless quantities)
ACC.rho_xy = 0.0; % along-track - cross-track
ACC.rho_xz = 0.9; % along-track - zenith
ACC.rho_yz = 0.0; % cross-track - zenith

% period that is used for estimating the accelerometer bias (unit = s)
ACC.dt_gravity_averaging_acc = 86400;

% averaging window width for accelerations derived from GNSS tracking
ACC.averaging_period = 1;      % unit = ACC.averaging_unit
ACC.averaging_unit = 'orbits'; % accepted units are 'orbits', 'seconds', 'days'

% slope of position noise in amplitude spectral density (dimensionless quantities)
ACC.position_noise_asd_slope = -0.25;

% sampling period of the GNSS tracking data (unit = s)
ACC.dt_gnss_tracking = 1.0;
```

*Listing 6: Parameters specifying the uncertainty in accelerometry and GNSS tracking.*

The inertial satellite position and velocity vectors define the local orbit reference frame (L-frame). The x-axis is in the direction of the velocity vector, the y-axis is orthogonal to the position and velocity vectors, and the z-axis completes a right-hand coordinate system. By construction of the frame, the z-axis points away from Earth. Considering a near-circular orbit, GNSS tracking noise is most conveniently specified in the local orbit reference frame, where the vertical z-component usually has a higher noise level than the horizontal x- and y-components.

The axes of the satellite body frame (B-frame) are, in principle, arbitrarily defined. However, the most common definition is that the axes are aligned with the main satellite body axes such that the x- and z-axes roughly point into the flight and nadir (toward Earth) directions, respectively, and the y-axis completes a right-hand coordinate system. Since the accelerometer is rigidly connected to the satellite, we specify the accelerometer noise in the B-frame.

For a near-circular orbit, the x-axes of the L- and B-frames point approximately in the same direction, and the y- and z-axes of the L- and B-frames point approximately in opposite directions. Small differences in the pointing result from the varying satellite attitude, driven by the attitude control law. The L- and B-frames are illustrated in Figure 7. For the GRACE mission, one satellite assumes an attitude as described here, and the other satellite is rotated by 180° about the z-axis to enable the ranging between the satellites.

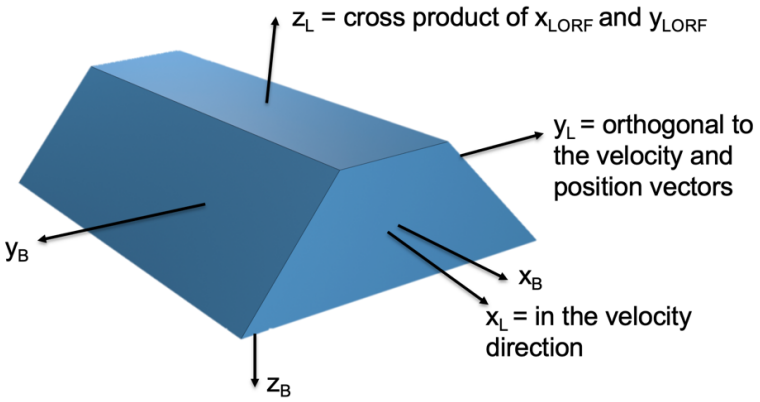


*Figure 7: Local orbit reference frame and satellite body reference frame.*

**Output figures**

The software tool will evaluate the acceleration and density signals along the orbit and determine the noise level of the in situ density observations derived from accelerometer data and the averaged density observations derived from GNSS tracking data. The results are illustrated in several figures, which place the signals and noise levels into context and compare in situ and averaged quantities. The first set of figures (cf. left column of Figure 8) shows the radiation pressure acceleration and its standard deviation. This is useful to understand how the averaging reduces the standard deviation compared to the in situ observations. Typically, the averaging strongly reduces the standard deviation in the along-track direction, whereas it is less effective in the cross-track and vertical directions. For a more convenient comparison, the standard deviations of the in situ and averaged radiation pressure acceleration are shown together in a separate set of figures (cf. right column of Figure 8). The in situ and averaged density observations are illustrated in the same way in Figure 9.

Since the software tool evaluates all signals and noise levels, we can analyze the contributions from different noise sources. Here, we consider the radiation pressure, the GNSS tracking noise, and the total noise. The GNSS tracking noise includes the accelerometer noise in the case of the in situ observations, otherwise not. The noise level is presented as a percentage of the density signal, as illustrated in Figure 10, which gives additional insights into the achievable signal-to-noise ratio.

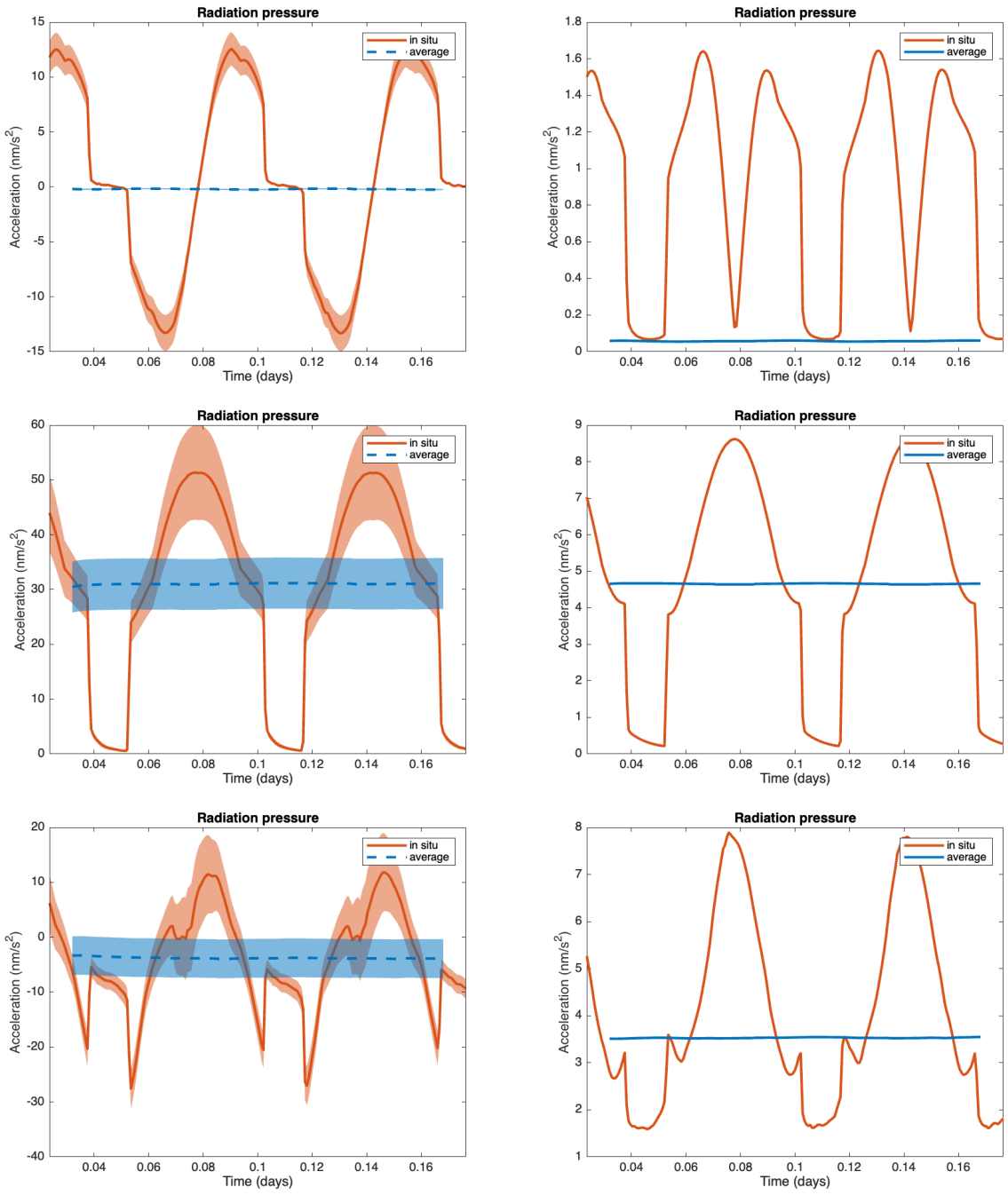


*Figure 8: Components of the radiation pressure acceleration and their standard deviation (left) and standard deviation (right). The top, middle, and bottom rows show the radiation pressure acceleration's x-, y-, and z-components, respectively.*

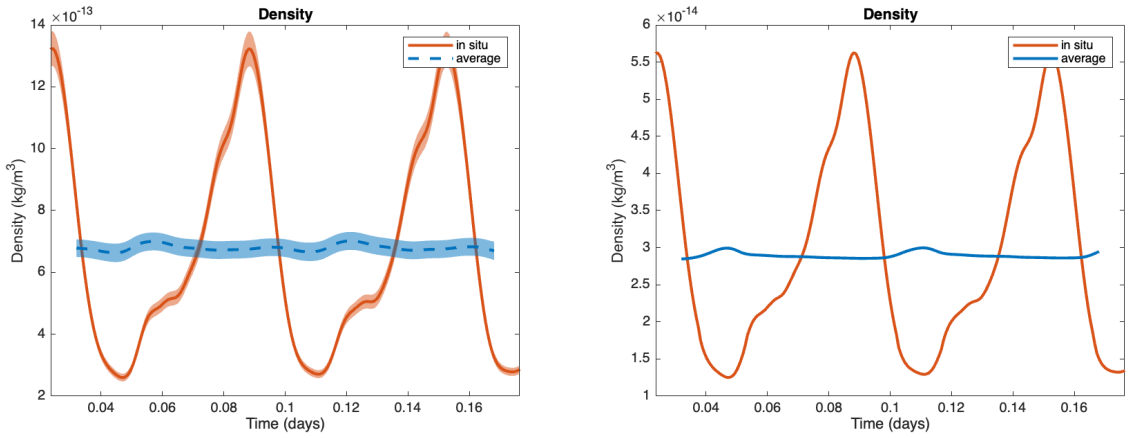


*Figure 9: Density observations and their standard deviation (left) and standard deviation (right).*

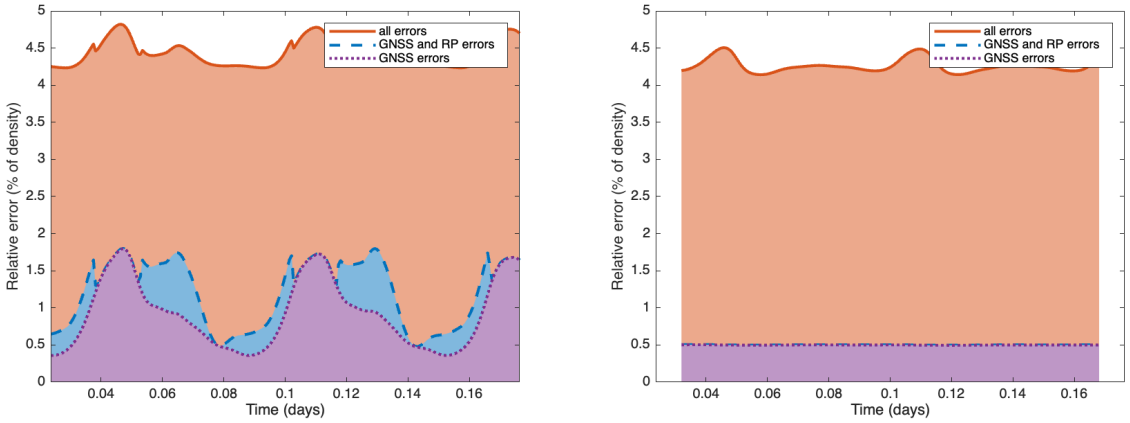


*Figure 10: Density standard deviation and the individual contributions for in situ density observations (left) and averaged density observations (right).*

**Limitations**

The software tool uses a database of space weather indices, namely the F10.7 solar radio flux and ap indices, as proxies for solar and geomagnetic activity. Further, a database of Earth orientation parameters is used for constructing the reference frame transformation between the Earth-fixed and celestial (inertial) reference frames. The database spans the period from January 2000 to December 2022. Therefore, the simulation period must fall into this date range. Running the Matlab script 'show_f107_ap.m' will generate a figure that can be explored to select a period with the desired solar and geomagnetic activity (cf. Figure 11).

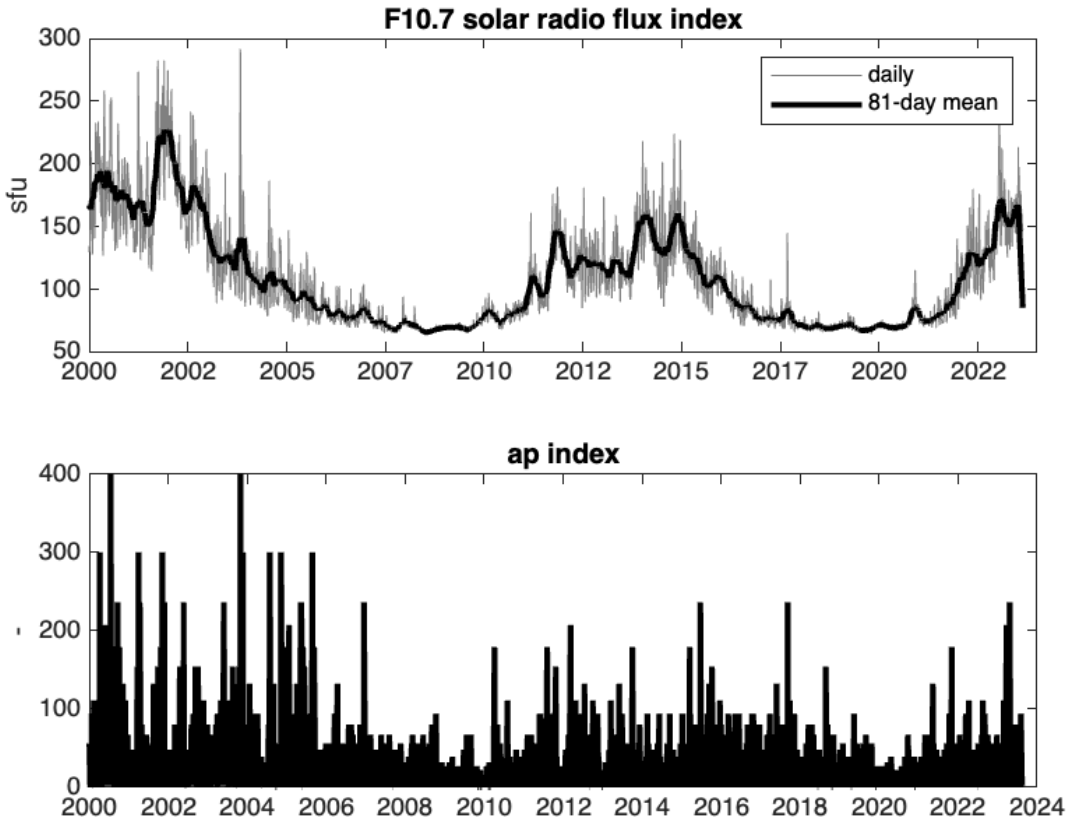


*Figure 11: F10.7 solar radio flux and ap index as proxies for solar and geomagnetic activity, respectively.*

**References**

V Bernstein, M Pilinski (2022) Drag coefficient constraints for space weather observations in the upper thermosphere. Space Weather 20: e2021SW002977. https://doi.org/10.1029/2021SW002977

E Doornbos (2011) Thermospheric density and wind determination from satellite dynamics. Ph.D. Thesis, Department of Astrodynamics and Satellite Missions, Delft University of Technology. Available at http://resolver.tudelft.nl/uuid:33002be1-1498-4bec-a440-4c90ec149aea

O Montenbruck, E Gill (2012) Satellite orbits. Springer. ISBN 978-3-540-67280-7. https://doi.org/10.1007/978-3-642-58351-3

JM Picone, AE Hedin, DP Drob, AC Aikin (2002) NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. J Geophys Res Space Phys 107(A12): 1468. https://doi.org/10.1029/2002JA009430

LH Sentman (1961) Free molecule flow theory and its application to the determination of aerodynamic forces. Technical Report. Lockheed Missiles and Space Co Inc, Sunnyvale, CA.

C Siemes, J van den IJssel, P Visser (in preparation) Uncertainty of thermosphere mass density observations derived from accelerometer and GNSS tracking data.

C Siemes, C Borries, S Bruinsma, I Fernandez-Gomez, N Hładczuk, J van den IJssel, T Kodikara, K Vielberg, P Visser (2023) New thermosphere neutral mass density and crosswind datasets from CHAMP, GRACE, and GRACE-FO. J. Space Weather Space Clim. 2023, 13, 16. https://doi.org/10.1051/swsc/2023014