

# The DLR Environment Library for Multi-Disciplinary Aerospace Applications

Lâle Evrim Briese<sup>1</sup>   Andreas Klöckner<sup>1</sup>   Matthias Reiner<sup>1</sup>

<sup>1</sup>Institute of System Dynamics and Control, DLR German Aerospace Center, Oberpfaffenhofen, Germany,  
Lale.Briese@dlr.de · Andreas.Kloeckner@dlr.de · Matthias.Reiner@dlr.de

## Abstract

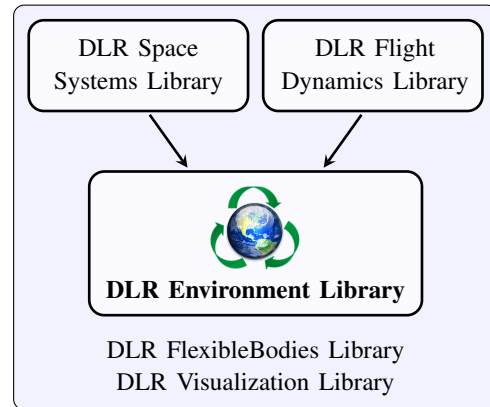
Environment models are vital elements for any type of vehicle dynamics simulations, such as aircraft or satellites. Recently, applications have been developed, where these previously unrelated regimes of operation need to be integrated, for example in end-to-end simulations of launch vehicles. This paper therefore introduces the new DLR Environment Library, which implements common models of planets, geospheres, currents, kinematics, and physical effects for such applications. It provides a set of environment models with minimal dependencies, complete compatibility to the Modelica Standard Library, and convenient drag & drop usage. The DLR Environment Library is expected to immensely aid developing end-to-end simulation models integrating components from DLR's SpaceSystems and FlightDynamics Libraries. In particular, it will importantly decrease modeling errors due to its consistent environment models.

*Keywords: environment modeling, gravitational models, planet models, atmosphere models, kinematic state models, space mission simulation, multi-disciplinary modeling*

## 1 Introduction

Modeling of environmental effects is highly relevant for vehicle simulations, such as aircraft (Klöckner et al., 2013; Looye, 2008), satellite (Reiner and Bals, 2014; Pulecchi et al., 2006), or launch vehicle simulations (Acquatella, 2016). While these domains have mostly been treated as independent in the past, latest developments point towards even more integrated simulation needs. For instance, reusable launchers will require accurate modeling of aircraft-like and satellite-like flight phases. Especially, combined multi-disciplinary simulations, including several vehicle types like launch vehicles and satellites with corresponding environmental conditions as well as ground stations, are of great interest within end-to-end space mission simulations.

For several years, the Institute of System Dynamics and Control at the DLR German Aerospace Center has been developing Modelica-based libraries for the modeling and simulation of flight vehicles (DLR FlightDynamics Library) and satellites (DLR SpaceSystems Library) as shown in Figure 1. These libraries can operate either in-

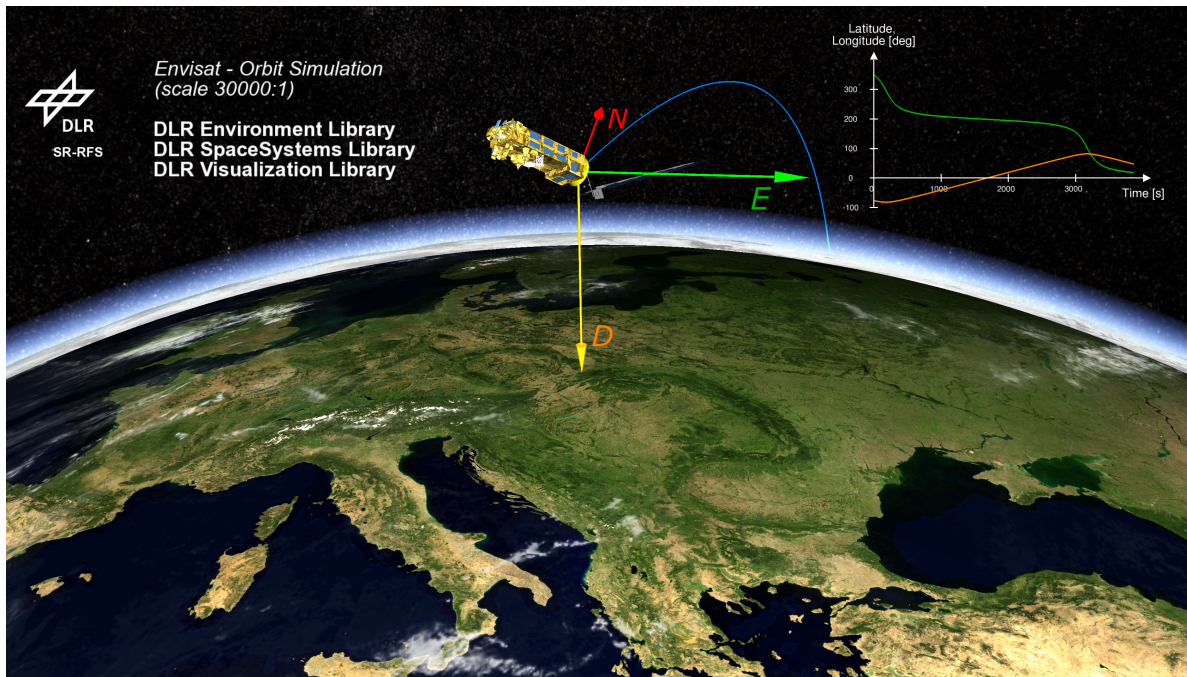


**Figure 1.** An overview of the interaction of application-based libraries with the new DLR Environment Library.

dependently from each other or in combination with other libraries. For example, the DLR FlexibleBodies Library is used for modeling flexible structures and the DLR Visualization Library is used for visualizing multibody systems.

Although these libraries share one common need for the modeling of environmental effects, there have been different application- and library-specific environment models for each library. Certainly, not every application requires the same level of detail or the same type of environment models for the specific design regime. For example, a satellite system in Low Earth Orbit (LEO) can neglect gravitational effects of another planet and a flight vehicle with a cruise flight altitude of 40.000ft is hardly influenced by the solar radiation pressure, unlike spacecraft in a deep space environment.

In general, most environment models are stored inside subpackages of application libraries, providing just the minimal amount of data needed for the realistic simulation of the desired application. For this purpose, most environment models take into account gravitational acceleration, atmospheric parameters, and specific influences which are relevant for use cases as presented for example in Reiner and Bals (2014), Looye (2008) or Pulecchi et al. (2006). The advantages of application- and library-dependent environment models are clearly the reduction of the level of detail and the simplification of complex environmental effects. This leads to a smaller amount of available models for individual purposes and consequently to less required maintenance.



**Figure 2.** Visualization of Envisat in an orbit simulation using the DLR Environment Library.

A major disadvantage of this strategy arises when multiple application-based libraries must be used for one comprehensive problem. The combination of several multi-disciplinary libraries can then lead to redundancies or discrepancies in the overall environmental formulation. This is especially the case if two or more application libraries are combined for the end-to-end simulation of e.g. reusable launch vehicles. For example, depending on the specific application, coordinate systems or rotation sequences can be defined differently and therefore can lead to often unmanageable errors. Also, different gravitational models can result in mismatched data of the Inertial Measurement Unit (IMU) of each system, finally increasing the overall error between vehicles in a multi-disciplinary simulation.

To prevent these problems, it has been decided to build a common library, based on knowledge of environmental effects inside application libraries. The overall goal of the library is to provide a modular, non-redundant and user-friendly formulation of environmental effects. It has to be compatible with the Modelica Standard Library (MSL) as well as the application libraries developed at the Institute of System Dynamics and Control.

Within this paper, the new Environment Library is presented. In Section 2, an overview of the library is given, including its purpose, its main characteristics, its basic structure as well as the verification of its compatibility with the MSL. Based on this section, some selected features of the library are further introduced in Section 3. The functionality of the models is demonstrated within Section 3 with specific examples provided by the application libraries mentioned before. The main advantages of the proposed library are summarized in Section 4.

## 2 Overview of the Library

For multi-disciplinary end-to-end simulations regarding all kinds of vehicles (from Earth-based flight and launch vehicles to spacecraft in deep space environment), specific but consistent environmental conditions have to be considered. The library is developed to fulfill these multi-disciplinary requirements and is therefore based on environment models from two application-related Modelica-based libraries developed by the Institute of System Dynamics and Control:

- DLR SpaceSystems Library (SSL) (Reiner and Bals, 2014), and
- DLR FlightDynamics Library (FDL) (Looye, 2008; Klöckner et al., 2014a)

The library in its current version is fully tested within the simulation environment Dymola 2017. Although it is designed as a stand-alone library, it is based on the Modelica Standard Library (3.2.2) and builds on the DLR Visualization Library (1.4) for optional visualization as presented in Figure 2 for an orbit simulation of the environmental satellite Envisat. The DLR Visualization Library is not required for the functionality of the provided environment models, but it enables drag & drop visualization of all parts of the simulation. With the visual effects, a better understanding of the overall model behaviour can be provided, especially when flight dynamics are considered.

The main characteristics of this library regarding its basic structure and the implementation of the provided models were determined by considering the following goals.

- **Modularity**

Models inside the library shall be able to work independently. A limited amount of interdependent models is considered acceptable.

- **Adaptivity**

The library shall be able to grow and has to allow individual modification. The library shall adapt easily to changes in other used libraries (e.g. MSL).

- **Generality**

The library shall be seen as a common ground for multiple disciplines. It shall therefore provide models which can adapt to requirements of multiple disciplines without affecting unrelated disciplines.

- **Reusability**

Models provided by the library shall be usable in different domains, at the best as drag & drop models.

- **User-friendliness**

The library and its models shall be designed without the need of intensive maintenance or without excessive user interaction or configuration.

- **Simplicity**

The amount of configurable parameters shall be reduced as much as possible and conditional changes of the model behaviour shall be implemented in separate models instead of using enumerated types.

## 2.1 Basic Structure of the Library

An overview of the top-level structure of the Environment Library is shown in Figure 3. The library provides a documentation with information about the library itself including contact information, references, release notes and a tutorial for beginners. Additionally, examples to demonstrate the functionality of the provided models are implemented. Environment models are stored in the main subpackages. Further dependencies between the main subpackages are kept minimal in order to achieve maximum modularity.

The main subpackages for the modeling of planets, geospheres and currents are created based on an object-oriented structure as shown in Figure 4. All main subpackages contain the package `BaseClasses` in which partial models for each discipline are implemented. These partial models provide specific functions to be accessed from anywhere within the simulation model corresponding to the `inner` and `outer` concept in Modelica. From these partial models, drag & drop models in the top-level of each subpackage are extended. This planet-independent library structure enables a highly modular, user-friendly, object-oriented and consistent modeling of environmental conditions. These advantages are vitally important for multi-disciplinary simulations regarding multiple vehicles in different environments. Especially, many separate models instead of one general model containing all possible

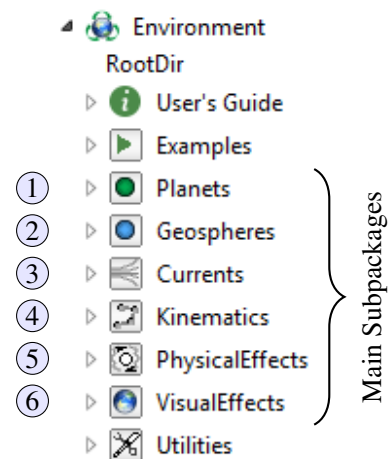


Figure 3. An overview of the top-level library structure.

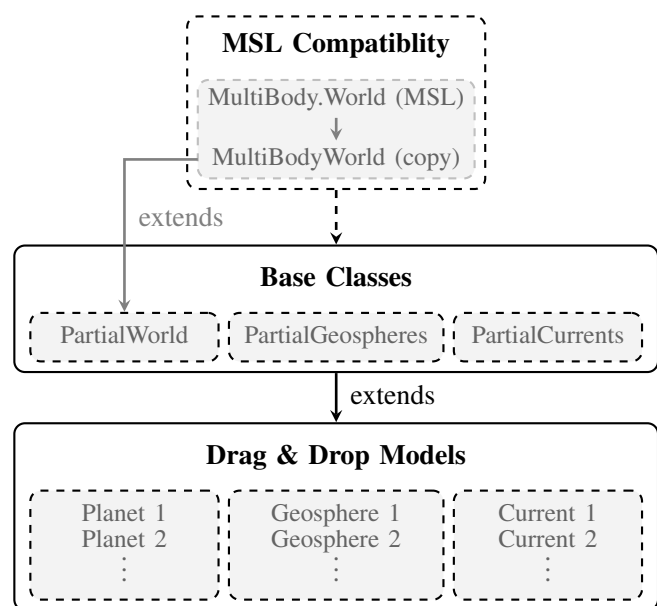


Figure 4. An overview of the object-oriented library structure.

environmental effects, offer a better understanding of the overall model behaviour. A brief summary of these subpackages is given below:

- **Planets ①**

This subpackage contains generic MSL-based planet models, providing relevant planet frames, the global simulation time, the time-dependent rotation angle of rotating planets, the planet constellation inside the solar system as well as the advanced replaceable `gravityAcceleration` function.

- **Geospheres ②**

Models which represent general geospheres (e.g. atmosphere) are included inside this package, providing inner models with replaceable functions similar to the `gravityAcceleration` function to calculate specific geospheric parameters as well as the mean current of the geosphere.



- **Current** ③

Inside this package, components for modeling of additional currents like user-defined wind are stored, which are designed as drag & drop models to induce currents (or wind effects in terms of the atmosphere) to a connected body frame.

- **Kinematics** ④

The package `Kinematics` contains several functions to describe kinematic relationships and coordinate transformations as well as models for automatic state selection for MSL-based bodies.

- **Physical Effects** ⑤

Inside the package `PhysicalEffects` specific models representing certain physical effects are implemented which provide for example forces due to the solar radiation pressure or information about the geomagnetic field.

- **VisualEffects** ⑥

Several models to visualize the Earth, the Moon and the Sun are implemented inside this package. They are based on components in the DLR Visualization Library. With the visualization software *SimVis* (Bellmann, 2009) and the provided high-resolution visualization data of the Earth, the simulation results for a certain problem can be shown in a realistic and easily recognizable way (see Figure 2).

Subpackage-specific functions are implemented inside the `Functions` packages, whereas components which are not intended for further usage are stored inside the `Internal` packages. Using common base classes for all drag & drop models, two possible modeling approaches can be followed by the user (van der Linden et al., 2014). First, the partial models from the `BaseClasses` packages can be placed inside generic simulation models such that they can be replaced dynamically for each application model. Second, the drag & drop models from the top level can be used as fully functional and stand-alone components of application models.

The `Utilities` package provides environment related constants, enumeration types, icons as well as general functions. Inside the subpackage `User`, individual user options can be stored as additional constants. Especially for large files that cannot be saved inside the `Resources` folder, like the visualization data of the Earth, a modifiable path name to the source directory can be supplied and managed by each user individually.

### 3 Selected Features of the Library

Within this section, some of the main subpackages and features of the Environment Library are emphasized. Selected use cases are provided in thematically related subsections to demonstrate the functionality of the presented models as well as the wide range of available model variants inside the library.

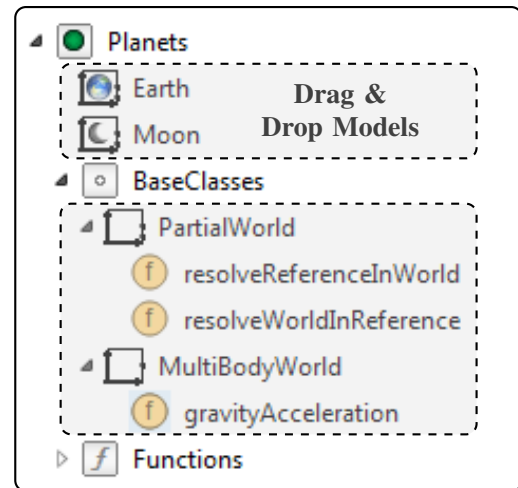


Figure 5. An overview of the subpackage `Planets`.

#### 3.1 The Planets subpackage

The `Planets` subpackage as shown in Figure 5 provides planet models, which are compatible to the standard multi-body world component of the MSL (Otter et al., 2003). This facilitates the switching to enhanced world models in application libraries without changing the application library structure or code.

The model `MultiBodyWorld` with the replaceable function `gravityAcceleration` is a modified copy of the original world component with two important changes. On the one hand, the parameters have been rearranged in additional tabs without changing their content. This is done in order to provide a better overview of the parameters inside the world component and to enhance the user-friendliness of the overall model with a thematically structured graphical user interface. On the other hand, the equations to define the position and orientation of the frame `frame_b` in the original world component are no longer defined as equations. Instead, they are integrated as variable declarations to the definition of the frame itself such that these values can be changed while using an `extends` statement. With this new modeling approach for an extended world model and its modified frame definition, moving planets within the solar system can be integrated into the simulation taking into account their influences on each other.

From this modified world component a partial model `PartialWorld` for planetary objects is extended introducing an additional frame and two internal functions to resolve any vector from the inertial frame `frame_b` to this new moving reference frame. Because the world component is defined as an `inner` model, it is possible to call the two internal functions inside the `PartialWorld` from anywhere inside the simulation model under the condition that an `outer` command referencing the world component is used. Although this modeling strategy is applicable to any rotating planetary object, the implementation of the Earth will be explained further in this section.



The Earth component is extended from the partial model `PartialWorld`. The frame `frame_b` is defined as an inertial frame (*Earth Centered Inertial*, ECI), whereas the new additional frame represents the *Earth Centered Earth Fixed* (ECEF) coordinate system with an attitude depending on the simulation time and the Earth's angular velocity. Additionally, the parameter `gravityType` is redefined with Earth-specific gravity types and the function `gravityAcceleration` is redeclared with the corresponding functions to calculate the gravity acceleration vector of the Earth.

With this modeling concept, components from the MSL can be used with the new world definition. This can be demonstrated with the example *Double Pendulum* placed on the Earth's surface. The original world component inside the MSL and the modified `MultiBodyWorld` are based on the same gravity model and therefore provide the same results as presented in Figure 6. Additionally, the resulting acceleration of the component `boxBody2` using a more advanced gravitational model based on the `Earth` is presented. It is shown, that the behaviour of the *Double Pendulum* changes significantly over time depending on the chosen gravitational model, demonstrating the consequences of using different gravitational models within end-to-end simulations.

Some planet-specific features implemented inside the extended planet models are further introduced using the `Earth` component as an example.

### Absolute Simulation Time

The new world component adds an absolute simulation time `julianDate`, which is initialized with parameters provided by the user either in Julian date format or in years, months, days and hours with minutes and seconds as fractions.

### Rotation of the Earth

The Earth's rotation angle at a certain time is determined as a function of the absolute simulation time as proposed inside the *Naval Observatory Vector Astrometry Software* (NOVAS) (Bangert et al., 2011). The transformation matrix from the ECI frame to the rotating ECEF frame can either be defined as a simplified rotation between these frames using only the rotation angle (ERA) around the Earth's rotation axis ( $z$ -axis) with respect to the ECI frame, or it can be calculated considering the nutation and precession depending on the Julian date as well as the difference in seconds between *Universal Time* and *Universal Coordinated Time* (Bangert et al., 2011). The leap seconds are automatically computed based on tabular data (Astronomical Almanac, 2010), but can also be provided by the user as input values.

### Gravity Acceleration

To calculate the gravity acceleration vector  $\mathbf{g}_0 \in \mathbb{R}^3$  with respect to the inertial frame, the `Earth` model provides a `gravityAcceleration` function comparable to the stan-

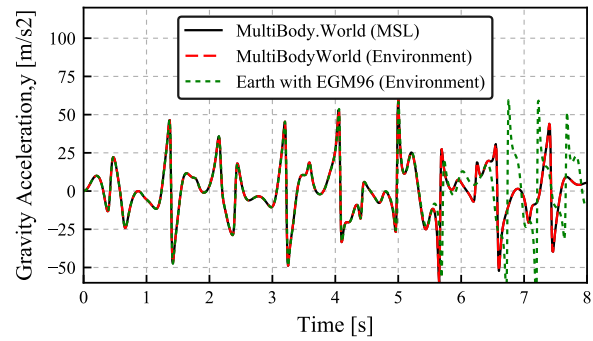


Figure 6. Compatibility of the planet models with the MSL.

dard world component. The basic gravity types from the original world component are still available to maintain the compatibility to the MSL. Additionally, more precise gravity models like the EGM96 (Lemoine et al., 1998) and the Vinti Order 6 (Bate et al., 1971) gravity models can be chosen for the calculation of the gravity acceleration vector  $\mathbf{g}_E \in \mathbb{R}^3$  of the Earth. The EGM96 model uses terms up to the second degree of the zonal harmonic coefficients of the gravitational potential as discussed in Reiner and Bals (2014). Those are only dependent on the symmetrical mass distribution along the  $z$ -axis of the Earth. The Vinti Order 6 potential function takes into account the perturbation accelerations due to the Earth's nonsphericity based on Bate et al. (1971).

If needed, the gravity acceleration from the Moon and the Sun can be considered inside the precise gravity models. Therefore, the current positions of the Moon  $\mathbf{r}_M$  and the Sun  $\mathbf{r}_S$  with respect to the Earth are calculated analytically with low precision formulae for planetary positions (van Flandern and Pulkkinen, 1979). As an alternative, they can also be obtained from the DE405 ephemeris files (Standish, 1998). Relying on the current Julian date as an input parameter, the DE405 ephemeris coefficients are extracted from an external C-code to calculate the positions of the Moon and the Sun.

The total gravity acceleration vector  $\mathbf{g}_0$  is finally calculated as the sum of the gravity acceleration from the Earth  $\mathbf{g}_E$ , the Moon  $\mathbf{g}_M$  and the Sun  $\mathbf{g}_S$ . The gravity acceleration vectors  $\mathbf{g}_M$  and  $\mathbf{g}_S$  are calculated according to the Equation (1) depending on the position  $\mathbf{r}_{M,S} \in \mathbb{R}^3$  and the gravitational constant  $G_{M,S}$  of the Moon or the Sun.

$$\mathbf{g}_{M,S} = G_{M,S} \left( \frac{\mathbf{r}_{M,S} - \mathbf{r}_E}{\|\mathbf{r}_{M,S} - \mathbf{r}_E\|^3} - \frac{\mathbf{r}_{M,S}}{\|\mathbf{r}_{M,S}\|^3} \right) \quad (1)$$

The absolute gravity acceleration for different gravity models is shown in Figure 7 for the given latitude of the International Space Station (ISS). The position of the ISS is calculated from orbital elements provided by *NORAD Two-Line Element sets* (TLE) (NASA, 2011) for a given point in time as implemented inside the SSL. The models provide very similar results, except for the expected disturbances.

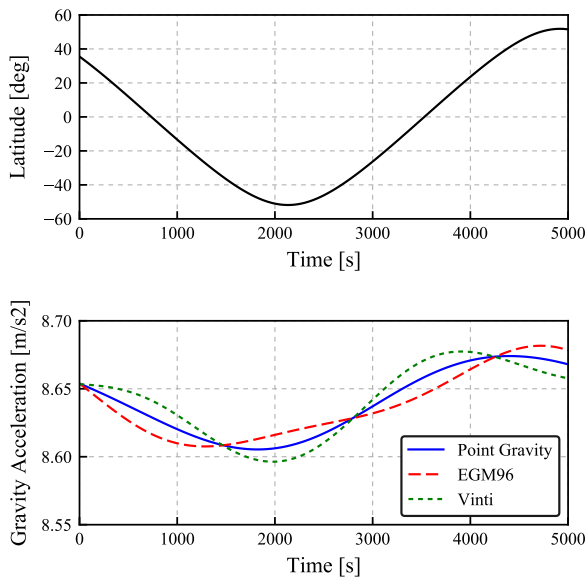


Figure 7. Comparison between different gravity models (ISS).

### 3.2 The Geospheres subpackage

The subpackage `Geospheres` is implemented to provide specific types of geosphere models, like atmosphere models which can be used in combination with flight or launch vehicles, respectively.

All main geosphere models are extended from the partial model `PartialGeosphere`. This model contains two replaceable functions, which can be called from anywhere inside a simulation model. Both functions require the position of an object with respect to the rotating reference frame. The outputs of the function `baseProperties` are the absolute pressure, temperature, density and speed of sound, corresponding to the `BaseProperties` implementation inside the MSL package `Media`. The output of the second replaceable function `meanCurrent` is the velocity of the geosphere-specific current.

Similar to the implementation of planets, the user can choose between two modeling concepts using either the `PartialGeosphere` or the stand-alone geosphere models from the top-level of this subpackage. The replaceable functions `baseProperties` and `meanCurrent` can be re-declared with advanced functions for each specific geosphere model.

In terms of atmosphere models, geodetic parameters such as latitude, longitude and altitude have to be calculated from the given input position. For this purpose, consistent kinematic functions inside the `Kinematics` package are used (see Section 3.5) approximating the shape of the planet according to the World Geodetic System '84 (WGS'84) (NIMA, 2000). Optionally, the user can decide if the geoid undulation between the calculated altitude and the *Mean Sea Level* shall be taken into account. For this reason, the geoid information based on the EGM96 model is computed with an external C-code (Lemoine et al., 1998).

In the Environment Library, different geosphere models for the Earth's atmosphere are implemented. For example, a constant atmosphere with user-provided parameters or a user-defined atmosphere with input values based on tabular data can be chosen. For the latter option, the tabular data is interpolated using the altitude of the object. Other geosphere components use standard atmosphere models as explained in the following list:

- **StandardAtmosphere (ISA)**

Within this component, two atmosphere models covering several regimes are implemented. The *Two Zones Model* by Schänzer (1969) can be chosen especially for flight vehicles with an altitude up to 40.000ft. The *Three Zones Model* (NASA, 2015) can be used if atmospheric conditions between the troposphere and the upper stratosphere are needed. This model is based on atmospheric measurements with separate curve fits for the troposphere, the lower and the upper stratosphere.

- **StandardAtmosphere76**

This component is based on the U.S. Standard Atmosphere model from 1976 where the atmospheric parameters can be determined for altitudes from -5 km up to 1000 km. For altitudes above 50 km, the data for this atmosphere model is based on rocket and satellite measurements (NASA, 1976).

- **NRLMSISEAtmosphere**

The NRL-MSISE-00 model is a highly accurate empirical model developed by the *U.S. Naval Research Laboratory* (NRL) (Picone et al., 2001). It is primarily used by spacecraft due to its accuracy in altitudes above 100 km and its range from the ground to the exosphere. The density and temperature at a certain position are computed using the NRL-MSISE-00 database with an external C-code. As inputs, the Julian date provided by the world component as well as the geodetic parameters latitude, longitude and altitude are required.

For all atmosphere models, the mean current is based on a logarithmic approach to determine the velocity vector in the Earth's boundary layer with respect to the ground. In Figure 8, a comparison between the provided atmosphere models is shown for the ascent phase of a generic launch vehicle depending on its current altitude. All atmosphere models provide similar results for the atmospheric density but significantly different results for example for the temperature corresponding to the approximation methods used inside particular atmosphere models. Especially in multi-disciplinary simulations, one common atmosphere model instead of many application-specific atmosphere models for several vehicles can therefore reduce errors in the overall model behaviour.

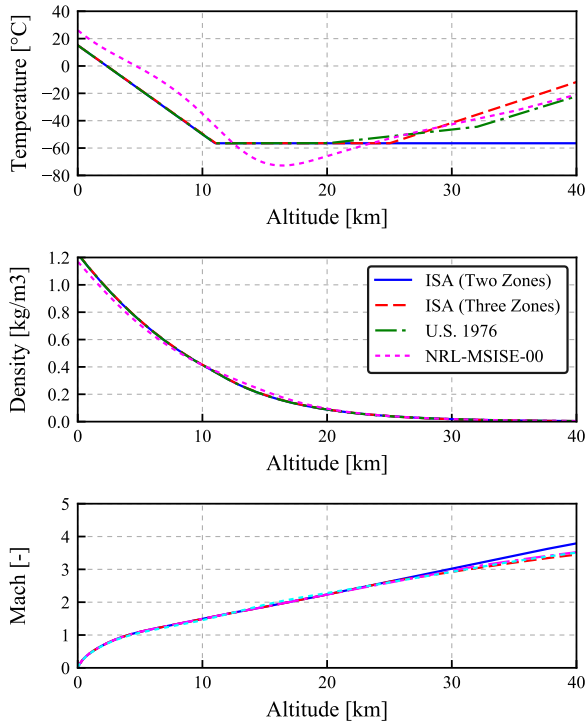


Figure 8. Comparison between different atmosphere models.

### 3.3 The Currents subpackage

Currents are used as a supplement to geosphere models as presented in Section 3.2. While geosphere models provide mean currents for any location with their internal function `meanCurrent`, a current model provides simplified models of local flow velocities such as turbulence or gusts. Each submodel (e.g. aircraft or spacecraft) can have its own local current model. All currents retrieve the mean current from the geosphere and add local effects to it.

A simple example is the continuous Dryden turbulence model (MIL-STD-1797A, 1990), which adds a low-pass-filtered white noise to the mean current (The MathWorks, 2016). Such an approach is illustrated in Figure 9 for a user-defined wind profile, where the filtered noise is added to the mean current from the geosphere model. This approach makes it possible to also cover distributed flow effects, such as wake vortices or delayed turbulence.

Like for the geosphere models, the user has the option to also use local wind or gust effects based on tabular data, which interpolates the velocity components according to the position of the object connected to the current's frame.

### 3.4 The PhysicalEffects subpackage

The subpackage `PhysicalEffects` provides stand-alone drag & drop models to automatically induce forces and torques due to physical effects on the attached frames. In contrast to the previous subpackages, these models are not based on a common partial model. However, all models fulfill the same goals as defined in Section 2 for instance in terms of modularity, simplicity and user-friendliness. Selected features of this subpackage are described below.

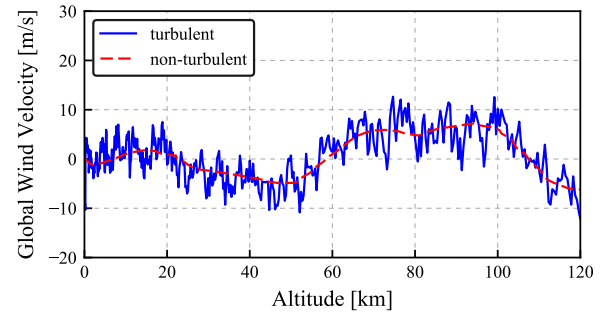


Figure 9. Application of a turbulent current on a wind profile.

### Gravity Gradient Torque

The gravity gradient torque is modeled as a torque  $\tau_a$  that acts on a connected frame with the position  $\mathbf{r}_{0,a}$  and the rotational transformation matrix  $\mathbf{T}_a$ . The torque is caused by the allocation of the mass with respect to its center and depends on the inertia tensor  $\mathbf{I}_B \in \mathbb{R}^{3 \times 3}$  (Larson and Wertz, 1999). In Equation (2), the gravity acceleration vector  $\mathbf{g}_0 \in \mathbb{R}^3$  is a function of the position  $\mathbf{r}_{0,a}$  and the Julian date  $t_J$  which is retrieved for the position of the connected frame directly from the world component.

$$\tau_a = \left( \mathbf{T}_a \mathbf{g}_0(\mathbf{r}_{0,a}, t_J) \frac{3}{\|\mathbf{r}_{0,a}\|} \right) \times \left( \mathbf{I}_B \mathbf{T}_a \frac{-\mathbf{r}_{0,a}}{\|\mathbf{r}_{0,a}\|} \right) \quad (2)$$

### Solar Radiation Pressure

The effect of the solar radiation pressure is modeled as a force  $\mathbf{f}_{sp}$  that acts on the connected frame. Shadows of the Moon and the Sun are considered with the shadow factor  $\chi_{sp} \in [0, 1]$  using a cylindrical shadow model. The equations are implemented as proposed in Montenbruck and Gill (2000). Required parameters are the effective area  $A_{sp}$  of the solar radiation pressure and its normal vector  $\mathbf{n}_{sp}$  as well as the coefficient of reflectivity of the material  $\xi_{sp} \in [0, 1]$  (total absorption to total reflection) as shown in Equations (3) to (5). The distance between the Sun and the frame is defined as  $\mathbf{d}_{sp} \in \mathbb{R}^3$ . The solar radiation pressure  $p_\odot$  is assumed to be constant for spacecraft near Earth.

$$c_{sp,\theta} = \frac{\mathbf{n}_{sp}}{\|\mathbf{n}_{sp}\|} \frac{\mathbf{d}_{sp}}{\|\mathbf{d}_{sp}\|} \quad (3)$$

$$\mathbf{c}_{sp,R} = \left[ (1 - \xi_{sp}) \frac{\mathbf{d}_{sp}}{\|\mathbf{d}_{sp}\|} + 2 \xi_{sp} c_{sp,\theta} \frac{\mathbf{n}_{sp}}{\|\mathbf{n}_{sp}\|} \right] \quad (4)$$

$$\mathbf{f}_{sp} = -\chi_{sp} p_\odot A_{sp} c_{sp,\theta} \mathbf{c}_{sp,R} \frac{AU^2}{\|\mathbf{d}_{sp}\|^2} \quad (5)$$

### Geomagnetic Field

The geomagnetic field can be computed for a connected frame by the `GeoMagneticField` component, using the US/UK World Magnetic Model (WMM) from 2010 or 2015 (Maus et al., 2010). The model provides a magnetic field vector  $\mathbf{B}_m \in \mathbb{R}^3$  that depends on the latitude, longitude and altitude of the component as well as the current



Julian date provided by the world model. The output vector is calculated in the local *North-East-Down* (NED) frame which can be transformed to any other coordinate system like ECI or ECEF with the provided functions inside the package `Kinematics` (see Section 3.5). The resulting geomagnetic field can be used for example in simulations with magnetic actuators or *Inertial Navigation Systems* (INS).

### Atmospheric Drag for Spacecraft

The atmospheric drag is caused by friction with the remainder of the atmosphere depending on the altitude. Like the solar radiation pressure, the atmospheric drag is modeled as a force and torque element acting on the attached frame which should be located at the center of mass of the object. The density  $\rho$  of the atmosphere is provided by the NRL-MSISE-00 model due to its accuracy in near Earth orbit. The drag force  $\mathbf{f}_{ad}$  and torque  $\boldsymbol{\tau}_{ad}$  can be computed using Equations (6) and (7) where  $\mathbf{v}_{rel}$  is the relative velocity of the object with respect to the rotating Earth.

$$\mathbf{f}_{ad} = -0.5 c_d A_{ad} \rho \|\mathbf{v}_{rel}\| \mathbf{v}_{rel} \quad (6)$$

$$\boldsymbol{\tau}_{ad} = 0.5 c_d A_{ad} \rho \|\mathbf{v}_{rel}\|^2 \left[ \frac{\mathbf{v}_{rel}}{\|\mathbf{v}_{rel}\|} \times \mathbf{T}_a \mathbf{d}_{cp} \right] \quad (7)$$

Required parameters are the drag coefficient  $c_d$ , the effective area  $A_{ad}$  and the vector from the center of pressure to the center of mass  $\mathbf{d}_{cp}$ , resolved in the attached frame.

### 3.5 The Kinematics subpackage

The Environment library includes a comprehensive toolset for coordinate transformations and kinematics simulation. These can be used for `MultiBody` models to flexibly

- compute kinematic states in different notations,
- define different notations of continuous states, and
- constrain the kinematics to lower-order models.

In order to compute kinematic states of a `MultiBody` model, functions are provided which transform the standard state set of a `MultiBody` frame (i.e. position  $\mathbf{r}_0$ , attitude  $\mathbf{R}, \mathbf{T}$ , and rotational velocity  $\mathbf{R}, \mathbf{w}$ ) to a broad variety of different notations. The structure of the provided models and functions reflects this distinction in separate collections of conversion functions. Since many notations, such as WGS'84 positions or an aircraft's attitude, are given relative to the planet's reference system, a further discrimination is made between conversions in the inertial and the world reference system (see Figures 10 and 11).

The conversion functions are used by the provided sensor models, which retrieve the states of a frame. The implemented functions to calculate the kinematic relationships are also generalized such that they can be used to compute the required parameters given in any other geodetic system instead of the WGS'84. The simplified and user-friendly structure of the kinematic functions provides

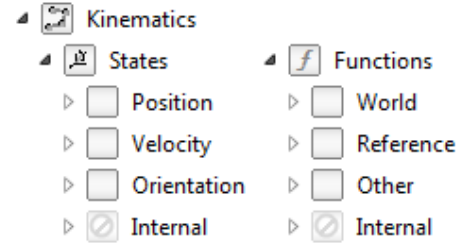


Figure 10. An overview of the package `Kinematics`.

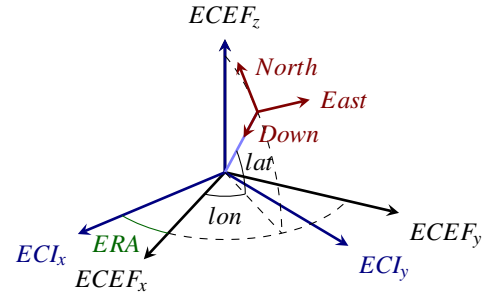


Figure 11. An overview of some basic coordinate systems.

a better overview for the user. Common functions can reduce errors due to different implementations of coordinate systems within application-based libraries.

In addition to computing the states in a required notation, the library provides models to define the very same notations as actual continuous time states of a `MultiBody` model by simply dragging the component into a model and connecting it to a frame connector. This is accomplished by first, defining the desired states with `stateSelect=StateSelect.always`, second, transforming the desired states into standard `MultiBody` notation, and finally, setting the frame variables to the result. As for the transformation functions, there are components to set position, velocity and attitude states independently from each other.

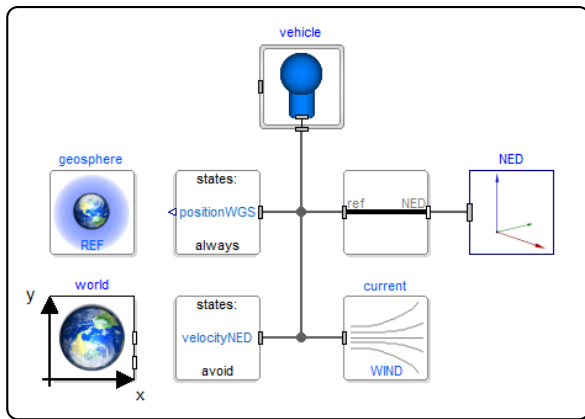
Finally, constraint models are provided, which interface seamlessly with the standard `MultiBody` models. This includes a generalization of the quasi-steady flight kinematics inside the FDL to general `MultiBody` models. Therefore, any six degrees of freedom (DOF) model can be transformed effectively into a three DOF model removing the rotational states. The transformation is accomplished by explicitly setting  $\mathbf{R}, \mathbf{w} = \{0, 0, 0\}$  and creating new unknown variables for the orientation  $\mathbf{Q}$ . The model thus interrupts the usual flow of calculation (conceptually a double integration from torques to rates and attitude) as shown in Equations (8) to (10).

$$\text{der}(\mathbf{R}, \mathbf{w}) := \mathbf{f}(\mathbf{t}) \quad \Rightarrow \quad \mathbf{0} := \mathbf{f}(\mathbf{t}), \quad (8)$$

$$\text{der}(\mathbf{Q}) := \mathbf{f}(\mathbf{R}, \mathbf{w}) \quad \Rightarrow \quad \text{no kinematics}, \quad (9)$$

$$\mathbf{t} := \mathbf{f}(\mathbf{Q}, \mathbf{R}, \mathbf{w}) \quad \Rightarrow \quad \mathbf{Q} := \mathbf{f}(\mathbf{t}). \quad (10)$$

By rooting `frame_a.R` in the model, the kinematics equations in the `MultiBody` components are disabled and by setting  $\mathbf{R}, \mathbf{w} = \{0, 0, 0\}$ , the dynamics equations implicit



**Figure 12.** An overview of the structure of a simulation model using the presented environment and kinematics models.

itly force the torques to be zero. In order to achieve a well-defined model, external torque equations are required to compute the unknown attitude variables  $Q$ .

As for the usage of the models provided by the subpackage `Kinematics`, the structure of a simulation model for a generic vehicle is shown representatively in Figure 12 in combination with the previously presented models within the DLR Environment Library.

## 4 Conclusion

The DLR Environment Library is a Modelica-based library for modeling environmental effects for application-specific libraries. Over the past years, environment models, optionally including planet definitions or atmospheric parameters, have been developed independently and in a smaller scale within each application library. These separate developments have induced problems due to redundant declarations, mismatched level of detail, accuracy and precision within multi-disciplinary projects. With the DLR Environment Library, these problems are solved, as introduced in previous sections.

Especially, the new library and modeling concept based on an object-oriented library structure provides several advantages as listed below:

- modular, reusable and comprehensible structure,
- easily adaptable to new requirements & applications,
- consistent definition of environmental conditions,
- simple, user-friendly and understandable models,
- reduced maintenance demands.

Although the development of general terrain, weather and aerodynamic models is in progress, these packages have been excluded from the content of this paper, since the models are not yet fully implemented and tested inside the Environment Library. The implementation of all planets within the solar system is also planned for the future.

## Acknowledgements

We would like to thank Dr. Gertjan Looye (DLR German Aerospace Center) for his contributions to former implementations of environment models inside the DLR FlightDynamics Library.

## References

- P. Acquatella. Launch Vehicle Multibody Dynamics Modeling Framework for Preliminary Design Studies. *6th International Conference on Astrodynamics Tools and Techniques (ICAAT)*, 2016.
- Astronomical Almanac. *The Astronomical Almanac for the Year 2011*. United Kingdom Hydrographic Office, 2010. ISBN: 978-07-0774-103-1.
- J. Bangert, W. Puatua, G. Kaplan, J. Bartlett, W. Harris, A. Fredericks, and A. Monet. User's Guide to NOVAS Version C3.1. Technical report, U.S. Naval Observatory, 2011.
- R. Bate, D. Müller, and J. White. *Fundamentals of Astrodynamics*. Dover Publications, Inc., 1971. ISBN: 0-486-60061-0.
- T. Bellmann. Interactive Simulations and advanced Visualization with Modelica. In *Proceedings of the 7th International Modelica Conference*, pages 541–550, 2009. doi:10.3384/ecp09430056.
- A. Klöckner, M. Leitner, D. Schlabe, and G. Looye. Integrated Modelling of an Unmanned High-Altitude Solar-Powered Aircraft for Control Law Design Analysis. In *Advances in Aerospace Guidance Navigation and Control - Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control*, pages 535–548. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38252-9.
- A. Klöckner, G. Looye, R. Müller, R. Kuchar, F. Re, and M. Leitner. Object-Oriented Aircraft Modeling with the DLR FlightDynamics library. In *9th AIRTEC 2014 International Congress*, 2014a.
- A. Klöckner, F. L. J. van der Linden, and D. Zimmer. Noise Generation for Continuous System Simulation. In *Proceedings of the 10th International Modelica Conference*, pages 837–846, 2014b. ISBN: 978-91-7519-380-9.
- A. Klöckner, A. Knobloch, and A. Heckmann. How to Shape Noise Spectra for Continuous System Simulation. In *Proceedings of the 11th International Modelica Conference*, pages 411–418, 2015. ISBN: 978-91-7685-955-1.
- W. J. Larson and J. R. Wertz. *Space Mission Analysis and Design*, volume 3. Microcosm Press and Kluwer Academic Publishers, 1999. ISBN: 1-881883-10-8.
- F. G. Lemoine, S. C. Kenyon, J. K. Factor, and R. G. Trimmer et al. The Development of the Joint NASA GSFC and National Imagery and Mapping Agency NIMA Geopotential Model EGM96. Technical report, National Aeronautics and Space Administration (NASA), 1998.

- G. Looye. The New DLR Flight Dynamics Library. In *Proceedings of the 6th International Modelica Conference*, volume 1, pages 193–202, 2008.
- S. Maus, S. Macmillan, S. McLean, B. Hamilton, A. Thomson, M. Nair, and C. Rollins. The US/UK World Magnetic Model for 2010-2015. Technical report, National Oceanic and Atmospheric Administration (NOAA), 2010.
- MIL-STD-1797A. *Flying Qualities of Piloted Aircraft*. U.S. Department of Defense, 1990. Military Standard.
- O. Montenbruck and E. Gill. *Satellite Orbits - Models, Methods and Applications*. Springer Verlag, Heidelberg, 2000. ISBN: 978-3-642-63547-2.
- NASA. U.S. Standard Atmosphere, 1976. Technical report, National Aeronautics and Space Administration, 1976.
- NASA. Definition of Two-line Element Set Coordinate System, 2011. National Aeronautics and Space Administration, [http://spaceflight.nasa.gov/realddata/sightings/SSapplications/Post/JavaSSOP/SSOP\\_Help/tle\\_def.html](http://spaceflight.nasa.gov/realddata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html).
- NASA. Earth Atmosphere Model, 2015. National Aeronautics and Space Administration, <https://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html>.
- NIMA. World Geodetic System 1984 - Its Definition and Relationships with Local Geodetic Systems. Technical report, National Imagery and Mapping Agency, 2000.
- M. Otter, H. Elmqvist, and S. Mattsson. The New Modelica MultiBody Library. In *Proceedings of the 3rd International Modelica Conference*, pages 311–330, 2003.
- J. M. Picone, A. E. Hedin, and A. C. Aikin D. P. Drob. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Journal of Geophysical Research*, 107, 2001. doi:10.1029/2002JA009430.
- T. Pulecchi, F. Casella, and M. Lovera. A Modelica Library for Space Flight Dynamics. In *Proceedings of the 5th International Modelica Conference*, pages 107–116, 2006.
- M. J. Reiner and J. Bals. Nonlinear inverse models for the control of satellites with flexible structures. In *Proceedings of the 10th International Modelica Conference*, pages 577–587, 2014. doi:10.3384/ECP14096577.
- G. Schänzer. *Einführung in die Flugphysik*. Institut für Flugführung, TU Braunschweig, 1969. Lecture notes.
- E. M. Standish. JPL Planetary and Lunar Ephemerides, DE405 / LE405. Technical report, Jet Propulsion Laboratory, 1998.
- The MathWorks. Dryden Wind Turbulence Model, 2016. <http://de.mathworks.com/help/aeroblks/drydenwindturbulencemodelcontinuous.html>.
- F. L. J. van der Linden, C. Schlegel, M. Christmann, G. Regula, C. I. Hill, P. Giangrande, J.-C. Maré, and I. Egaña. Implementation of a Modelica Library for Simulation of Electromechanical Actuators for Aircraft and Helicopters. In *Proceedings of the 10th International Modelica Conference*, pages 757–766, 2014. doi:10.3384/ECP14096757.
- T. C. van Flandern and K. F. Pulkkinen. Low-Precision Formulae for Planetary Positions. *The Astrophysical Journal Supplement Series*, 41:391–411, 1979.