User Manual 3.3 Properties and models: Mass and Forces

De Wiki
Aller à : navigation, rechercher
User Manual 3.3 Properties and models: Mass and Forces

Sommaire

Introduction

Scope

The models presented here are useful for forces computations.

Javadoc

The AeroModel and the DirectRadiativeModel are available in the package fr.cnes.sirius.patrius.assembly.models.

The associated properties are available in the package fr.cnes.sirius.patrius.assembly.properties.

Features Description

Aero model

The aerodynamic model (see AeroModel) is based on an instance of Assembly. It implements the DragSensitive interface which makes it suitable for drag force models. It provides the drag acceleration as well as its partial derivatives.

The model supports two types of shapes to represent an assembly from the aerodynamic point of view: sphere and facet.

Thus, in order to create an aerodynamic model, the assembly must have :

- Parts with mass properties (MassProperty) and
- Parts with either :
  - an AeroFacetProperty : defining the normal and tangential aerodynamic coefficients (Cn and Ct) on a facet.
  - an AeroSphereProperty : to define a drag coefficient (Cx) on a spherical shape.

Parts that have aerodynamic properties are not required to have a mass property (eg the solar panels may have aerodynamic properties without mass properties). The acceleration applied on the assembly is the sum over the forces provided by each of those parts divided by the total mass of the assembly.

Given an initial orbit bulletin, `initialSpacecraftState`, a user sample is given below:

```
// create an assembly builder
final AssemblyBuilder builder = new AssemblyBuilder();

// main part with mass (bus)
builder.addMainPart("bus");
builder.addProperty(new MassProperty(5000.), "bus");

// solar panel with aerodynamics properties
builder.addPart("solarPanel", "bus", transform);

// the AERO_SPHERE property (simplified representation)
builder.addProperty(new AeroSphereProperty(2., 1.), "solarPanel");

// get the generated assembly
final Assembly assembly = builder.returnAssembly();

// link the assembly to the Orekit frames tree, for forces computation
assembly.initMainPartFrame(initialSpacecraftState);
```

```
// create an aerodynamic model
final AeroModel aeroModel = new AeroModel(assembly);
```

Tabulated aero model

The aerodynamic model (see GlobalAeroModel) is a complex aerodynamic model based on a generic user-defined drag coefficient. The drag coefficient is provided using an implementation of DragCoefficientProvider. It returns absorption, specular and diffuse parts of drag coefficients in satellite frame.

It is based on an instance of Assembly. The main part of the assembly must have a AeroProperty

Drag Lift model

The DragLift model is based on an instance of Assembly. It implements the DragSensitive interface and provides the drag and lift acceleration (see corresponding JavaDoc).

In order to create an aerodynamic model based on the DragLift model, the assembly must have :

- Parts with mass properties (MassProperty) and
- Parts with Aero global properties (see AeroGlobalProperty)

Given an initial orbit bulletin, initialSpacecraftState, a user sample is given below:

```
// create an assembly builder
final AssemblyBuilder builder = new AssemblyBuilder();

// main part with mass (bus)
builder.addMainPart("bus");
builder.addProperty(new MassProperty(5000.), "bus");

// Add an AeroGlobal property to the assembly
builder.addProperty(new AeroGlobalProperty(2, 0, new ConstantFunction(1.)),
"bus");

// get the generated assembly
final Assembly assembly = builder.returnAssembly();

// link the assembly to the Orekit frames tree, for forces computation
assembly.initMainPartFrame(initialSpacecraftState);

final DragLiftModel dragLift = new DragLiftModel(assembly.returnAssembly());
```

Properties for aero computation

AeroFacetProperty

An aerodynamic facet property contains an attribute of the MAT_GEO_Facets Facet type and two attributes for the two drag coefficients (normal coefficient, tangent coefficient). It is a required property to create aerodynamic models. This class also defines a facet from an orientation and an area. This property is required to create aerodynamic models and allows the part to contribute to the atmospheric pressure applied on an assembly.

The property type associated is **AERO_FACET**.

AeroSphereProperty

An aerodynamic sphere property contains three attributes:

- the sphere radius;
- the atmospheric height scale factor.
- the drag coefficient

This property is required to create aerodynamic models and allows the part to contribute to the atmospheric pressure applied on an assembly.
The atmospheric height scale factor is used only when computing the partial derivatives with respect to position, using an approximated formula; it is not compulsory for the user to set its value (when not, the derivatives with respect to position will be zero).

The property type associated is **AERO_SPHERE**.

AeroGlobalProperty

This class allows the user to define the aerodynamic properties of the different parts that constitute the assembly. The attributes of an AeroGlobal property are:

- C,,x,,: the drag coefficient

- C,,z,,: the lift coefficient

- S: the surface used for the computation of aerodynamics' forces.

The user has two possibilities when constructing a new AeroGlobal property:

- Variable aerodynamics' coefficients (of the ParamDiffFunction type) and a constant surface,

- Constant aerodynamics' coefficients and a variable surface (of the ParamDiffFunction type).

Direct radiative model

The radiation pressure model (see DirectRadiativeModel) is based on an instance of Assembly. It implements the radiative pressure interface (see RadiationSensitive). This class provides the acceleration applied on the assembly.

The acceleration is computed from all the parts of the assembly and finally a coefficient k0 is multiplied to the acceleration force. Its default value (if not given) is 1.0.

The model supports two types of shapes to represent an assembly from the radiative point of view: sphere and facet.

Thus, in order to create a radiative model, the assembly must contain parts with the required properties:

- RadiativeProperty : to define the thermo-optical coefficients.

- RadiativeSphereProperty or RadiativeFacetProperty : to define a part with a spherical shape or with a facet.

The acceleration applied on the assembly is the sum over the accelerations provided by each part. The parts with the properties described above can contribute to the computation.

```
A user sample is given below:
// we assume that an assembly is defined, as well as input parameters for the
computation
// (instance of a SpacecraftState and the incoming radiation flux)

// create a radiative model
final double k0 = 0.98;
final DirectRadiativeModel radiativeModel = new
DirectRadiativeModel(assembly, k0);

// compute the acceleration due to the radiation pressure
final Vector3D computedAcc =
radiativeModel.radiationPressureAcceleration(spacecraftState, flux);
```

Rediffused radiative model

The rediffused solar pressure model (see RediffusedRadiativeModel) is based on an instance of Assembly. It implements the rediffused radiative pressure interface (see RediffusedRadiationSensitive). The class RediffusedRadiationPressure implements the OREKIT interface (see ForceModel) and provides the acceleration applied on the assembly (method "redistributedRadiationPressureAcceleration"). The acceleration is computed from all the parts of the assembly.

The model supports two types of shapes to represent an assembly from the radiative point of view: sphere and facet.

Thus, in order to create a rediffused radiative model, the assembly must contain parts with the required properties:

- RadiativeProperty : to define the albedo thermo-optical coefficients.
- RadiativeIRProperty : to define the infrared thermo-optical coefficients
- RadiativeSphereProperty or RadiativeFacetProperty : to define a part with a spherical shape or with a facet.

The acceleration applied on the assembly is the sum over the accelerations provided by each part. The parts with the properties described above can contribute to the computation.

A user sample is given below:

```
// create a rediffused radiative model (k0 albedo = 1; k0 infrared = 1)
final IEmissivityModel model = new KnockeRiesModel();
final FactoryManagedFrame itrfFrame = FramesFactory.getITRF();
final CelestialBody sun = CelestialBodyFactory.getSun();
final RediffusedRadiativeModel rm = new RediffusedRadiativeModel(false,
false, 1, 1, assembly);

// create a rediffused radiative force instance (with 10 coronas, 10
meridians)
final RediffusedRadiationPressure r = new RediffusedRadiationPressure(sun,
itrfFrame, 10, 10, model, rm);

// compute the acceleration due to the rediffused radiation pressure
final Vector3D computedAcc = r.computeAcceleration(spacecraftState);
```

Radiative properties

RadiativeProperty

A radiative property contains three attributes, for the three thermo-optical coefficients of a part. It is a required property to create radiative models or rediffused radiative models (albedo pressure).

The property type associated is **RADIATIVE**.

RadiativeIRProperty

A infrared radiative property contains three attributes, for the three thermo-optical coefficients of a part. It is a required property to create rediffused radiative models (infrared emissivity pressure).

The property type associated is **RADIATIVEIR**.

RadiativeFacetProperty

A radiative facet property contains an attribute of the MAT_GEO_Facets Facet type. This class defines a facet from an orientation and an area. This property is required to create radiative or rediffused radiative models and allows the part to contribute to the radiation pressure, albedo pressure or infrared emissivity pressure applied on an assembly.

The property type associated is **RADIATIVE_FACET**.

RadiativeSphereProperty

A radiative sphere property contains an attribute for the sphere radius. This property is required to create radiative or rediffused radiative models and allows the part to contribute to the radiation pressure, albedo pressure or infrared emissivity pressure applied on an assembly.

The property type associated is **RADIATIVE_SPHERE**.

Mass model

The assembly mass model implements the MassProvider interface and is able to interact with a Numerical propagator with regards to changing masses (such as that of the propellant tank). To date, the only implemented class is MassModel.

An instance of MassModel can be created from an Assembly as long as the Assembly has parts that have a MassProperty. Upon creation, every one of these properties is associated to an additional equation (MassEquation) and an initial additional state (that of the MassProperty at instanciation time). Different parts can be associated to the same mass property (e.g. thrusters using the same tank).

```
String thruster = "thruster";

// create an assembly
AssemblyBuilder builder = new AssemblyBuilder();

builder.addMainPart(thruster);

MassProperty massProp = new MassProperty(9000.);

builder.addProperty(massProp, thruster);

Assembly assembly = builder.returnAssembly();

// create a mass model
MassModel model = new MassModel(assembly);
```

The user can call the `getTotalMass()` method to obtain the total mass of the spacecraft. Additionally, the user cal call the `getMass(String)` method to see the mass of a specific part. For example, in order to see the mass of the part called "thruster", the user can execute the following instruction:

```
final double partMass = model.getMass(thruster);
```

Upon creating a NumericalPropagator and a mass model, the user MUST pass along the additional equations to the propagator, to make sure that the propagation updates the mass properties of the spacecraft (e.g. in case there are any maneuvers involved) :

```
propagator.setMassProviderEquation(model);
```

The provided mass model should be the same as provided for force models to ensure propagation consistency.

In order to create force models (such as continuous impulse maneuvers) that modify the mass of a tank, the user must indicate which tank the force model uses. This is done by giving, at instanciation time, the MassProvider model as well as the name of the part that will "lose" mass :

```
ConstantThrustManeuver constant = new ConstantThrustManeuver(date, dt,
thrust, isp, direction, model, thruster);
propagator.addForceModel(constant);
```

By doing so, when the thrusters fire, the NumericalPropagator can update the mass of the correct part (thruster).

It should be noted that different force models can use the same propellant tank. The

NumericalPropagator will account for all mass variations, even in case some of the maneuvers overlap :

```
ConstantThrustManeuver constant1 = new ConstantThrustManeuver(date1, dt1,
thrust1, isp1, direction1, model, thruster);
ConstantThrustManeuver constant2 = new ConstantThrustManeuver(date2, dt2,
thrust2, isp2, direction2, model, thruster);
ConstantThrustManeuver constant3 = new ConstantThrustManeuver(date3, dt3,
thrust3, isp3, direction3, model, thruster);
propagator.addForceModel(constant1);
propagator.addForceModel(constant2);
propagator.addForceModel(constant3);
```

The IInertiaProperty implementations all use a MassProperty and all the "Inertia" models implement the MassProvider interface and associate each detected MassProperty to a MassEquation.

Mass property

A mass property simply contains an attribute for the mass of the considered part.

The property type associated is **MASS**.

Getting Started

Modèle:SpecialInclusion prefix=$theme sub section="GettingStarted"/

Contents

Interfaces

Interface	Summary	Javadoc
WallGasTemperatureProvider	Interface for wall gas temperature providers.	...
DragCoefficientProvider	Interface for drag coefficients.	...
AlphaProvider	Interface for alpha coefficient (used in Cook model).	...

Classes

Class	Summary	Javadoc
RadiativeProperty	This class is a part property for the PATRIUS assembly. It is the radiative characterization of a part.	...
RadiativeIRProperty	This class is a part property for the PATRIUS assembly. It is the infrared radiative characterization of a part.	...
RadiativeSphereProperty	This class is a part property for the PATRIUS assembly. It allows to define a sphere, that is a shape used for the radiative models.	...
RadiativeFacetProperty	This class is a part property for the PATRIUS assembly. It allows to define a facet, that is a shape used for the radiative models.	...

AeroSphereProperty	This class is a part property for the PATRIUS assembly. It allows to define a sphere, that is a shape used for the aerodynamic models.	...
AeroFacetProperty	This class is a part property for the PATRIUS assembly. It allows to define a facet, that is a shape used for the aerodynamic models.	...
MassProvider	Interface that represents a spacecraft that has parts that can change masses.	...
MassProperty	This class is a part property for the PATRIUS assembly. It allows to define a mass for a part.	...
AeroModel	This class is an aerodynamic model for the PATRIUS assembly.	...
DirectRadiativeModel	This class is an direct solar radiative model for the PATRIUS assembly.	...
RediffusedRadiativeModel	This class is an rediffused radiative model for the PATRIUS assembly.	...
CnCookModel	Cook model for drag normal coefficient.	...
CtCookModel	Cook model for drag tangential coefficient.	...
ConstantWallGasTemperature	Class for constant wall gas temperature.	...
CookWallGasTemperature	Class for wall gas temperature following Cook model.	...
GinsWallGasTemperature	Class for wall gas temperature following Cook model adapted to Gins.	...
GlobalAeroModel	Aerodynamic model with generic user defined drag coefficient.	...
AeroProperty	Aerodynamic properties for generic user defined drag coefficient.	...
DragCoefficient	Drag coefficient container (absorption, specular and diffuse parts).	...
AlphaConstant	Constant alpha coefficient.	...
AlphaCookModel	Alpha coefficient following Cook law.	...

Tutorials

Tutorial 1

Modèle:SpecialInclusion prefix=$theme sub section="Tuto1"/

Tutorial 2

Modèle:SpecialInclusion prefix=$theme sub section="Tuto2"/

Tips & Tricks

Modèle:SpecialInclusion prefix=$theme sub section="Tips"/

Récupérée de
«
http://patrius.cnes.fr/index.php?title=User_Manual_3.3_Properties_and_models:_Mass_and_Forces&oldid=1308 »

Catégorie :

- User Manual 3.3 Spacecraft

Menu de navigation

Outils personnels

- 3.139.235.177
- Discussion avec cette adresse IP
- Créer un compte
- Se connecter

Espaces de noms

- Page
- Discussion

Variantes

Affichages

- Lire
- Voir le texte source
- Historique
- Exporter en PDF

Plus

Rechercher

Rechercher Lire

PATRIUS

- Welcome

Evolutions

- Main differences between V4.15 and V4.14
- Main differences between V4.14 and V4.13

- Main differences between V4.13 and V4.12
- Main differences between V4.12 and V4.11
- Main differences between V4.11 and V4.10
- Main differences between V4.10 and V4.9
- Main differences between V4.9 and V4.8
- Main differences between V4.8 and V4.7
- Main differences between V4.7 and V4.6.1
- Main differences between V4.6.1 and V4.5.1
- Main differences between V4.5.1 and V4.4
- Main differences between V4.4 and V4.3
- Main differences between V4.3 and V4.2
- Main differences between V4.2 and V4.1.1
- Main differences between V4.1.1 and V4.1
- Main differences between V4.1 and V4.0
- Main differences between V4.0 and V3.4.1

User Manual

- User Manual 4.15
- User Manual 4.14
- User Manual 4.13
- User Manual 4.12
- User Manual 4.11
- User Manual 4.10
- User Manual 4.9
- User Manual 4.8
- User Manual 4.7
- User Manual 4.6.1
- User Manual 4.5.1
- User Manual 4.4
- User Manual 4.3
- User Manual 4.2
- User Manual 4.1
- User Manual 4.0
- User Manual 3.4.1
- User Manual 3.3

Tutorials

- Tutorials 4.15
- Tutorials 4.14
- Tutorials 4.13.5
- Tutorials 4.12.1
- Tutorials 4.8.1
- Tutorials 4.5.1
- Tutorials 4.4
- Tutorials 4.1
- Tutorials 4.0

Links

- CNES freeware server

Navigation

- Accueil
- Modifications récentes
- Page au hasard
- Aide

Outils

- Pages liées
- Suivi des pages liées
- Pages spéciales
- Adresse de cette version
- Information sur la page
- Citer cette page

- Dernière modification de cette page le 1 mars 2018 à 12:54.

- Politique de confidentialité
- À propos de Wiki
- Avertissements