User Manual 3.3 Celestial bodies

De Wiki
Aller à : navigation, rechercher
User Manual 3.3 Celestial bodies

Sommaire

Introduction

Scope

The celestial bodies are described by their main features : position and gemoetry. The positions are ephemeris that must be loaded from models, the geometries are created as one axis ellipsoids. The package provides a factory able to create any celestial body of the solar system.

Javadoc

The classes for bodies desciption are available in the package `bodies` of OREKIT and Patrius.

Library	Javadoc
Orekit	Package org.orekit.bodies
Patrius	Package fr.cnes.sirius.patrius.bodies

Links

Orekit Bodies architecture description, Orekit site.

Useful Documents

None as of now.

Package Overview

This package overview is from the OREKIT website, under apache license.

Features Description

Celestial bodies

The moon, the sun and planets of the solar system are all represented by the CelestialBody interface. This class associates a name (eg Sun) to :

- a gravitational coefficient GM
- a body centered **inertially oriented** frame
- a body centered **body oriented** frame.

The user can use the static methods of the CelestialBodyFactory to create instances of the most common celestial bodies. See CelestialBodyFactory.

Body shapes

The one-axis ellipsoid is a good approximate model for most planet-size and larger natural bodies. It is the equilibrium shape reached by a fluid body under its own gravity field when it rotates. The symmetry axis is the rotation or polar axis.

OneAxisEllipsoid

This type is used to represent the shape of a planet. One very useful implementation represents an ellipsoid (OneAxisEllipsoid). It is constructed from an equatorial radius, a flattening coefficient, and a reference frame that will be used to localize Geodetic points on the shape.

It could be interesting to obtained the geodetic coordinates of the nadir point of the satellite. To that purpose, one can use the method getIntersectionPoint(Line, Vector3D, Frame, AbsoluteDate) of the object OneAxisEllipsoid.

```
AbsoluteDate date = new AbsoluteDate(new DateComponents(2008, 03, 21),
                                             TimeComponents.H12,
                                             TimeScalesFactory.getUTC());
```

```
Frame frame = FramesFactory.getITRF();
// Body shape model
OneAxisEllipsoid earth = new OneAxisEllipsoid(6378136.460, 1 / 298.257222101,
frame);

// Satellite on any position

circ = new CircularOrbit(7178000.0, 0.5e-4, 0., FastMath.toRadians(50.),
FastMath.toRadians(0.),
                                   FastMath.toRadians(90.),
PositionAngle.MEAN,
                                   FramesFactory.getEME2000(), date, mu);

// Transform satellite position to position/velocity parameters in EME2000
and ITRF200B
PVCoordinates pvSatEME2000 = circ.getPVCoordinates();
PVCoordinates pvSatItrf  = frame.getTransformTo(FramesFactory.getEME2000(),
date).transformPVCoordinates(pvSatEME2000);
Vector3D pSatItrf  = pvSatItrf.getPosition();

// Nadir point of the satellite
Vector3D pointItrf     = new Vector3D.ZERO;
Vector3D direction = Vector3D(1., pSatItrf,-1., pointItrf);
Line line = new Line(pSatItrf, direction);
// intersection point between the ellipsoid and the line that joins the
satellite and the center of the body
GeodeticPoint nadir = earth.getIntersectionPoint(line, pSatItrf, frame,
date);
```

GeometricBodyShape and ExtendedOneAxisEllipsoid

The GeometricBodyShape interface extends BodyShape for more complex computations. ExtendedOneAxisEllipsoid implements it. See the specific page for more details.

GeodeticPoint

The geodetic point is defined by a latitude, a longitude and an altitude in the frame associated to the body. It could be interesting to know the position of a satellite in terms of geodetic coordinates rather than Cartesian ones and vice versa (the corresponding methods in OneAxisEllipsoid).

```
// equatorial radius of the celestial body
double ae = 6378137.0;
// flatness of the celestial body
double f = 1.0 / 298.257222101;
// date
AbsoluteDate date = AbsoluteDate.J2000_EPOCH;
// reference frame attached to the body
Frame frame = FramesFactory.getITRF();
// body shape model (ellipsoid)
OneAxisEllipsoid model = new OneAxisEllipsoid(ae, f, frame);
```

```
// transformation with jacobian matrix : cartesian to geodetic

// initial cartesian point that will be transformed
Vector3D cp = new Vector3D(4637885.347, 121344.608, 4362452.869);
// coresponding geodetic point
GeodeticPoint gp = model.transform(cp, frame, date);

// transformation with jacobian matrix : geodetic to cartesian

// initial geodetic point that will be transformed
GeodeticPoint gp2 = new GeodeticPoint(0.852479154923577, 0.0423149994747243,
111.6);
// coresponding Cartesian point
Vector3D cp2 = model.transform(gp2);
```

It could be also interesting to obtain the jacobian matrix of the transformation.

```
// transformation : cartesian to geodetic

double[][] jacobianGeodesicWrtCartesian = new double[3][3];
gp = model.transformAndComputeJacobian(cp, frame, date,
jacobianGeodesicWrtCartesian);

// transformation : geodetic to cartesian

double[][] jacobianCartesianWrtGeodesic = new double[3][3];
Vector3D cp2 = model.transformAndComputeJacobian(gp2,
jacobianCartesianWrtGeodesic );
```

Ephemeris Loader

For any celestial body of the Solar System, the actual computation of its position and velocity relies on the JPL planetary ephemerides files. These files are binary files and loaded thanks to the JPLEphemeridesLoader object.

For the moment, this object is only able to load the DE 405 or the DE 406 files which are the most commonly used data files. The former covers the years 1600 to 2200 at maximum precision, the latter covers the years-3000 to +3000 at only slightly reduced precision. The DE 421 file which is the latest JPL ephemeris with fully consistent treatment of planetary and lunar laser ranging data (Folkner *et al* 2009) is not yet readable by JPLEphemeridesLoader. However the DE 405 file is the basis for the Astronomical Almanac and leads to sufficiently accurate results and, for most purposes, even the accuracy of DE 406 is sufficient.

JPLEphemeridesLoader

In order to generate the ephemerides of one celestial body of the Solar System, one has to use the JPLEphemeridesLoader as follows :

```
final JPLEphemeridesLoader loaderEMB = new JPLEphemeridesLoader(fileName,
                     JPLEphemeridesLoader.EphemerisType.EARTH_MOON);
final JPLEphemeridesLoader loaderSSB = new JPLEphemeridesLoader(fileName,

JPLEphemeridesLoader.EphemerisType.SOLAR_SYSTEM_BARYCENTER);
CelestialBodyFactory.addCelestialBodyLoader(CelestialBodyFactory.EARTH_MOON,
loaderEMB);
CelestialBodyFactory.addCelestialBodyLoader(CelestialBodyFactory.
SOLAR_SYSTEM_BARYCENTER, loaderSSB);

// Reference frame
Frame icrf = FramesFactory.getICRF();

// OREKIT ephemeris of the Sun
JPLEphemeridesLoader loader = new JPLEphemeridesLoader("unxp2000.405",
                JPLEphemeridesLoader.EphemerisType.SUN);

// Creation of the Sun
CelestialBody sun = loader.loadCelestialBody(CelestialBodyFactory.SUN);

// Coordinates of the Sun given a date and a reference frame
PVCoordinates pvOREKIT = sun.getPVCoordinates(date, icrf);
```

When the user wants to create a JPLEphemeridesLoader, first of all, he must supply the folder where the DE 405 and DE 406 files are stored. Then he has to give the name of the file which contains the data (DE 405 or DE 406), the type of celestial body and a date (desired central date) as entries of the JPLEphemeridesLoader.

The first argument (name of the data file) may be `null`. In this case, OREKIT takes the first compatible file found. If one wants only files from the DE 405 ephemerides, for instance, he has to give the String `"^unx[mp](\\d\\d\\d\\d)\\.(?:405)$"`. The last argument can also be `null`. If the central date is not mentionned an arbitrary 100 days range will be loaded whereas if it is mentionned all data within a +/-50 days range around this date will be loaded.

Once the data is loaded, thanks to the JPLEphemeridesLoader, the user can create the celestial body with the method loadCelestialBody() of the JPLEphemeridesLoader class. To do so, the user has to be sure that all required data is loaded. Indeed, apart from the Moon, the Earth and the Earth-Moon barycenter, the creation of a celestial body requires some data on the Earth Moon barycenter and the Solar System barycenter in order to instanciate the frame in which the ephemerides of the celestial body will be defined. Therefore, the user has to create a JPLEphemeridesLoader for the Earth-Moon barycenter and the Solar System barycenter prior to creating the celestial body, otherwise OREKIT will arbitrarily load a DE file to generate the corresponding ephemerides that are used afterwards for the generation of the frame. Then the user has to complete the list of the loaders of the CelestialBodyFactory with these two loaders before calling the method loadCelestialBody().

NB : The user has to clean the CelestialBodyFactory memory if he does not want to work with the previously defined celestial bodies.

JPL ephemerides

The JPL ephemerides data is available on the JPL FTP server [R3].

Available ephemerides :

Development Ephemerides	Created in...	Description
DE102	September 1981	includes nutations but not librations. Referred to the dynamical equator and equinox of 1950. Covers JED 1206160.5 (-1410 APR 16) to JED 2817872.5 (3002 DEC 22)
DE200	September 1981	includes nutations but not librations. Referred to the dynamical equator and equinox of 2000. Covers JED 2305424.5 (1599 DEC 09) to JED 2513360.5 (2169 MAR 31). This ephemeris was used for the Astronomical Almanac from 1984 to 2003. (See Standish, 1982 and Standish, 1990).
DE202	October 1987	includes nutations and librations. Referred to the dynamical equator and equinox of 2000. Covers JED 2414992.5 (1899 DEC 04) to JED 2469808.5 (2050 JAN 02).
DE403	May 1993	includes nutations and librations. Referred to the International Celestial Reference Frame. Covers JED 2305200.5 (1599 APR 29) to JED 2524400.5 (2199 JUN 22). Fit to planetary and lunar laser ranging data.(See Folkner et al. 1994).
DE405, DE406	May 1997	For the DE405: includes both nutations and librations. Referred to the International Celestial Reference Frame. Covers JED 2305424.50 (1599 DEC 09) to JED 2525008.50 (2201 FEB 20) For the DE406 : the same as the DE405 except the time span : Spans JED 0624976.50 (-3001 FEB 04) to 2816912.50 (+3000 MAY 06) This is the same integration as DE405, with the accuracy of the interpolating polynomials has been lessened to reduce file size for the longer time span covered by the file.
DE410	April 2003	includes nutations and librations. Referred to the International Celestial Reference Frame. Covers JED 2415056.5 (1900 FEB 06) to JED 2458832.5 (2019 DEC 15). Ephemeris used for Mars Exploration Rover navigation.
DE413	November 2004	includes nutations and librations. Referred to the International Celestial Reference Frame. Covers JED 2414992.5, (1899 DEC 04) to JED 2469872.5 (2050 MAR 07). Created to update the orbit of Pluto to aid in planning for an occultation of a relatively bright star by Charon on 11 July 2005.
DE414	May 2005	includes nutations and librations. Covers JED 2414992.5, (1899 DEC 04) to JED 2469872.5 (2050 MAR 07). Fit to ranging data from MGS and Odyssey through 2003. (See Konopliv et al., 2006.)

DE418	August 2007	includes nutations and librations. Covers JED 2414864.5 (1899 JUL 29) to JED 2470192.5 (2051 JAN 21)
DE421	February 2008	includes nutations and librations. Referred to the International Celestial Reference Frame. Covers JED 2414864.5 (1899 JUL 29) to JED 2471184.5 (2053 OCT 09) Fit to planetary and lunar laser ranging data. (See Folkner et al., 2009)
DE422	September 2009	includes nutations and librations. Referred to the International Celestial Reference Frame. Covers JED 625648.5, (-3000 DEC 07) to JED 2816816.5, (3000 JAN 30). Intended for the MESSENGER mission to Mercury. Extended integration time to serve as successor to DE406. Fit to ranging data from MGS and Odyssey through 2003. (See Konopliv et al., 2010.)
DE423	February 2010	includes nutations and librations. Referred to the International Celestial Reference Frame version 2.0. Covers JED 2378480.5, (1799 DEC 16) to JED 2524624.5, (2200 FEB 02). Intended for the MESSENGER mission to Mercury.

Simplified analytical models

Meeus Model

The Meeus Model is a simplified model which gives the position of the Sun and the Moon with respect to the time T expressed in centuries (TT time scale). This model is an analytical model less precise than the DE ephemerides given by JPL. It is adapted for onboard applications.
The class implementing the Meeus Model allows three different models computing the position of the Sun with appropriate equations : the standard model (provided by Jean Meeus), the Stela model and an onboard model. The main differences between these model is the computation of the obliquity of the ecliptic : indeed, its value is fixed to 0 for the standard model, given by an expression involving [math]T, T^{2}[/math]for Stela model and [math]T, T^{2}, T^{3}[/math] for the onboard model. Moreover, the onboard model computed the position in J2000 frame, whereas standard and Stela models use respectively EOD and MOD frame.

Regarding the precision, one has to expect a maximum difference of 25593km in position for the Sun and a maximum angular difference of 34.6 *(wrt DE405 ephemerides). For the Moon, one has to expect a maximum difference of 26 km in position and a maximum angular difference of 15.2* (wrt DE405 ephemerides). As for the performances (in terms of execution time), the Meeus model for the Sun is faster than the DE405 ephemerides. However, we did not come to the same conclusion for the Moon even by decreasing the degree of precision (number of terms taken into account to compute the latitude, longitude and distance which are needed to compute in fine the position of the Moon).

	Deviation in position wrt DE423	**Angular deviation wrt DE423**
Sun	12000 km	34.6
Moon 62x66x46	12.4 km	15.2
Moon 26x13x13	225 km	122
Moon 9x4x3	1591 km	889

	Deviation in position wrt DE405	Angular deviation wrt DE405
Sun	25593km	34.6
Moon 62x66x46	26 km	15.2

	Number of elementary operations	Number of trigonometric operations
Sun	89	10
Moon 62x66x46	2671	182
Moon 26x13x13	917	60
Moon 9x4x3	401	24

References for the tables :

- "Modèles d'éphémérides luni-solaires", CNES DCT/SB/MS, 03/14/2011
- "Modèle MEEUS pour éphémérides Lune-Soleil : Compléments sur le nombre d'opérations", CNES DCT/SB/MS

	execution time - DE405	execution time- MEEUS
Sun	59 s 548 ms	19 s 938 ms
Moon 62x66x46	11 s 625 ms	3 mn 22 s 323 ms
Moon 26x13x13	11 s 625 ms	1 mn 17 s 74 ms
Moon 9x4x3	11 s 625 ms	40 s 326 ms

In order to build such a Sun or Moon, one has to use the object MeeusSun or MeeusMoon which both extend AbstractCelestialBody. Note that it is not possible to build those celestial bodies from the CelestialBodyFactory, for the moment.

Basic board Sun model

The basic board Sun model is a simplified analytical model which gives the direction of the Sun (the normalized position) with respect to time.
This model is similar to the Meeus model (the constants of the model are different) and is adapted for onboard applications. The reference inertial frame is the CIRF.

Getting Started

TBD

Contents

Interfaces

Interface	Summary	Javadoc
BodyShape	Interface representing the rigid surface shape of a natural body.	...
CelestialBody	Interface for celestial bodies like Sun, Moon or solar system planets.	...

Classes

Class	Summary	Javadoc

CelestialBodyFactory	Factory class for bodies of the solar system.	...
GeodeticPoint	Point location relative to a 2D body surface.	...
JPLEphemeridesLoader	Loader for JPL ephemerides binary files (DE 405, DE 406, ...).	...
OneAxisEllipsoid	Modeling of a one-axis ellipsoid.	...
MeeusSun	Position of the Sun according to Meeus model. Three models with there appropriate equations are avaible : the standard model (former MeeusSun), Stela model (former MeeuSunStela) and an on-board model	...
MeeusMoon	Position of the Moon according to Meeus model.	...
BasicBoardSun	Direction of the Sun according to a basic board Sun model.	...

Tutorials

Tutorial 1

TBD

Tips & Tricks

Non yet !

- Dernière modification de cette page le 28 février 2018 à 15:41.

- Avertissements
-