

# User Manual 3.4.1 Geometry

De Wiki

Aller à : [navigation](#), [rechercher](#)

[User Manual 3.4.1 Geometry](#)

## Sommaire

- [1 Introduction](#)
  - [1.1 Scope](#)
  - [1.2 Javadoc](#)
  - [1.3 Links](#)
  - [1.4 Useful Documents](#)
  - [1.5 Package Overview](#)
- [2 Features Description](#)
  - [2.1 2D Objects](#)
  - [2.2 3D Objects](#)
  - [2.3 Interactions between objects](#)
- [3 Getting Started](#)
  - [3.1 Interactions](#)
- [4 Contents](#)
  - [4.1 Interfaces](#)
  - [4.2 Classes](#)

## Introduction

### Scope

The geometry objects implemented in the PATRIUS library are both infinite and finite shapes.

The finite shapes shall be used for example to represent parts of a spacecraft or celestial bodies. The infinite ones can represent more mathematical surfaces such as instruments' characteristics.

All the objects provide methods to compute interactions with lines (intersections, distances, etc...), points and other objects.

### Javadoc

The geometry objects are available in the package `org.common.math3.geometry.euclidean.threed` in the Commons Math Library (and Commons Math add-ons).

#### Library

Commons Math

#### Javadoc

[Package org.common.math3.geometry.euclidean.threed](#)  
[Package org.common.math3.geometry.euclidean.threed](#)

## Links

Commons Math contains several other packages related to geometry (1D, 2D, 3D) and provides objects for Binary Space Partitioning (BSP) Tree. See javadoc for more information. Some other "basic" objects are contained in the Commons Math threed package and documented in other parts of this manual (eg : Rotations, Matrix3D).

## Useful Documents

Useful reference documents for this theme can be found here :

- Nürnberg, R.; *Distance from a Point to an Ellipse*, Imperial College London, 2006, <http://www2.imperial.ac.uk/~rn/distance2ellipse.pdf>.

## Package Overview

The aim of the following diagram is to present the architecture implemented for the Geometry package. This does not mean that all the classes and/or interfaces are in the Geometry package and it doesn't mean that all the classes and interfaces of the package are presented.



## Features Description

### 2D Objects

The two dimensional objects (in three dimensional space) implemented in the PATRIUS library are given hereunder :

- Lines
- Planes
- Plates
- Disks
- Ellipses

Please refer to the [Classes section](#) for more information.

### 3D Objects

The three dimensional objects (in three dimensional space) implemented in the PATRIUS library are given hereunder :

- Infinite and Finite Cylinders (right circular, elliptic and rectangular)
- Infinite and Finite Cones (right circular, oblique circular and rectangular)
- Spheres, Spheroids and Spherical Caps
- Parallelepipeds

Please refer to the [Classes section](#) for more information.

## Interactions between objects

[Interfaces](#) have been implemented for geometrical objects in order to set a standard as to what

method each and any geometrical object should implement.

As of now, (some of the) geometrical objects must provide methods to :

- Check if a line intersects with said objects,
- Compute and return the intersection points,
- Compute and return the closest point on the object surface to a given user point,
- Compute and return the distance between these two points,
- Compute and return the closest point on the object surface to a given user line and return the closest points on the line and on the object,
- Compute and return the distance between an object and a given user line.

Please refer to the [Interactions section](#) for more details.

## Getting Started

### Interactions

Given an object implementing the [Shape](#) interface, the following methods are available :

**intersects(Line) : boolean** Test the intersection of the object with a user given line.

Code snippet :

```
Line line = new Line(lineOrigin, lineDirection);
boolean result = myObject.intersects(line);
```

**getIntersectionPoints(Line) : Vector3D[]** Compute the intersection point of the object with a user given line.

Code snippet :

```
Line line = new Line(lineOrigin, lineDirection);
Vector3D[] intersectionPoints = myObject.getIntersectionPoints(line);
```

**closestPointTo(Vector3D) : Vector3D** Compute the point on the surface of the object that is the closest to a user given point.

Code snippet :

```
Vector3D point = new Vector3D(1.0, 2.0, 3.0);
Vector3D result = myObject.closestPointTo(point);
```

**distanceTo(Vector3D) : double** Compute the shortest distance from the surface of the object and user given point.

Code snippet :

```
Line line = new Line(lineOrigin, lineDirection);
double distance = myObject.distanceTo(line);
```

**closestPointTo(Line) : Vector3D[]** Compute the points realizing the shortest distance to a line (the first one is from the line, the second one from the object ). If the line intersects the object , the two points are identical, and equal to the first one returned by the "getIntersectionPoints" method. If the line is parallel to the object 's side, the closest point to the origin is used.

Code snippet :

```
Line line = new Line(lineOrigin, lineDirection);
Vector3D[] points = myObject.closestPointTo(line);
```

**distanceTo(Line) : double** Compute the shortest distance from a line to the surface of the object.

Code snippet :

```
Line line = new Line(lineOrigin, lineDirection);
double distance = myObject.distanceTo(line);
```

## Contents

### Interfaces

| Interface                            | Summary   | Javadoc |
|--------------------------------------|---|---------|
| <a href="#">Cone</a>                 | This interface extends the SolidShape interface for the particular case of cones.   | ...     |
| <a href="#">CrossSectionProvider</a> | Interface for all geometric objects that can provide their cross section from a direction defined by a Vector3D.  | ...     |
| <a href="#">Cylinder</a>             | This interface extends the SolidShape interface for the particular case of cylinders.   | ...     |
| <a href="#">IEllipsoid</a>           | This interface extends the InfiniteShape interface for the particular cases of ellipsoid objects.   | ...     |
| <a href="#">InfiniteCone</a>         | This interface extends the InfiniteShape interface for the particular cases of infinite cones.  | ...     |
| <a href="#">InfiniteCylinder</a>     | This interface extends the InfiniteShape interface for the particular cases of infinite cylinders.  | ...     |
| <b>InfiniteShape</b>                 | This interface extends the Shape interface for the particular cases of infinite shapes.   | ...     |
| <b>Shape</b>                         | This interface is the <b>main interface for all shapes</b> , including infinite and solid shapes. It defines the general methods required by each geometrical object implementing any of the above interfaces : All of them must be able to compute their intersection and distance to a line. See [MAT_GEO_Home#HInteractionsbetweenobjects Interactions]. | ...     |
| <b>SolidShape</b>                    | This interface extends the Shape interface for the particular cases of solids.  | ...     |

## Classes

### Basic Shapes

| Class                          | Summary  | Javadoc |
|--------------------------------|--|---------|
| <a href="#">Disk</a>           | The class represent disks in a three dimensional space.  | ...     |
| <a href="#">Ellipse</a>        | The class represent ellipses in a three dimensional space.   | ...     |
| <a href="#">Line</a>           | The class represent lines in a three dimensional space.  | ...     |
| <a href="#">Plane</a>          | The class represent planes in a three dimensional space.   | ...     |
| <a href="#">Plate</a>          | This is a describing class for a 3D rectangle plate shape, with some algorithm to compute intersections and distances to some other objects.       | ...     |
| <a href="#">Parallelepiped</a> | This is a describing class for a rectangle parallelepiped shape, with some algorithm to compute intersections and distances to some other objects. | ...     |

### Cylinders

| Class   | Summary   | Javadoc |
|---|---|---------|
| <a href="#">InfiniteEllipticCylinder</a>      | This class is the Infinite Elliptic Cylinder class.   | ...     |
| <a href="#">InfiniteRightCircularCylinder</a> | This is a describing class for a 3D infinite right circular cylinder, with some algorithm to compute intersections and distances to some other objects.                                 | ...     |
| <a href="#">InfiniteRectangleCylinder</a>     | This is a describing class for a 3D infinite rectangle cylinder, with some algorithm to compute intersections and distances to some other objects.                                      | ...     |
| <a href="#">EllipticCylinder</a>              | This is a describing class for a 3D oblique circular cylinder ended by two planes normal to its axis, with some algorithm to compute intersections and distances to some other objects. | ...     |
| <a href="#">RightCircularCylinder</a>         | This is a describing class for a 3D right circular cylinder ended by two planes normal to its axis, with some algorithm to compute intersections and distances to some other objects.   | ...     |

NB : For the rectangular cylinder, please refer to the parallelepiped class mentioned above.

### Cones

| Class                                     | Summary  | Javadoc |
|---|--|---------|
| <a href="#">InfiniteRightCircularCone</a> | This is a describing class for a 3D infinite cone, with some algorithm to compute intersections and distances to some other objects.   | ...     |
| <a href="#">InfiniteEllipticCone</a>      | This class is the Infinite Elliptic Cone class.  | ...     |
| <a href="#">InfiniteRectangleCone</a>     | This is a describing class for a 3D infinite cone, with some algorithm to compute intersections and distances to some other objects.   | ...     |
| <a href="#">RightCircularCone</a>         | This is a describing class for a 3D right circular cone ended by a plane normal to its axis, with some algorithm to compute intersections and distances to some other objects. | ...     |

[EllipticCone](#) This is a describing class for a 3D elliptic cone ended by a plane normal to its axis, with some algorithm to compute intersections and distances to some other objects. ...

[RectangleCone](#) This is a describing class for a 3D rectangle cone ended by a plane normal to its axis (pyramid), with some algorithm to compute intersections and distances to some other objects. ...

## Ellipsoids

| Class                     | Summary  | Javadoc |
|---------------------------|--|---------|
| <a href="#">Ellipsoid</a> | This is a describing class for an ellipsoid, with some algorithm to compute intersections and distances to some other objects.         | ...     |
| <a href="#">Sphere</a>    | This is a describing class for a 3D spherical shape, with some algorithm to compute intersections and distances to some other objects. | ...     |
| <a href="#">Spheroid</a>  | This is the Spheroid (also called Revolved Ellipsoid) class.   | ...     |

## Other objects

| Class                        | Summary   | Javadoc |
|------------------------------|---|---------|
| <a href="#">Facet</a>        | Implements a representation of a facet.               | ...     |
| <a href="#">SphericalCap</a> | Implements a representation of a spherical cap solid. | ...     |

Récupérée de « [http://patrius.cnes.fr/index.php?title=User\\_Manual\\_3.4.1\\_Geometry&oldid=1680](http://patrius.cnes.fr/index.php?title=User_Manual_3.4.1_Geometry&oldid=1680) »  
Catégorie :

- [User Manual 3.4.1 Mathematics](#)

## Menu de navigation

### Outils personnels

- [3.136.19.124](#)
- [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

## 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 4 avril 2018 à 15:19.

- [Politique de confidentialité](#)
- [À propos de Wiki](#)
- [Avertissements](#)





