

# User Manual 3.4.1 Maneuvers

De Wiki

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

[User Manual 3.4.1 Maneuvers](#)

## 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 Constant Thrust Maneuver](#)
  - [2.2 Variable Thrust Maneuver](#)
  - [2.3 Impulse Maneuver](#)
  - [2.4 List of maneuvers](#)
- [3 Getting Started](#)
- [4 Contents](#)
  - [4.1 Interfaces](#)
  - [4.2 Classes](#)

## Introduction

### Scope

The scope of this section is to present the maneuver models available through the Orekit and Patrius libraries.

It is possible to define the thrust direction in a frame. If this frame is set to null, it means that the thrust direction is defined in spacecraft frame. The [SpacecraftFrame class](#) cannot be used.

### Javadoc

All the classes related to maneuvers are in the following packages:

<b>Library</b>	<b>Javadoc</b>
Orekit	<a href="#">Package org.orekit.forces.maneuvers</a>
Orekit addons	<a href="#">Package org.orekit.forces.maneuvers</a>
Patrius	<a href="#">Package fr.cnes.sirius.patrius.forces.maneuvers</a>

### Links

None as of now.

## Useful Documents

None as of now.

## Package Overview

The maneuvers conception is described hereafter :

[Fichier:Propulsion3.png](#)

## Features Description

### Constant Thrust Maneuver

The [ConstantThrustManeuver class](#) implements the [ForceModel interface](#). The thrust, the ISP and the acceleration direction are constant values.

#### The thrust direction can either be defined:

- In spacecraft frame:

```
final double duration = 1000;
final double thrust = 420;
final double isp = 318;
final Vector3D direction = Vector3D.PLUS_I;
final ConstantThrustManeuver maneuver = new ConstantThrustManeuver(date,
duration, thrust, isp, direction, massProvider, "thruster");
```

- In a frame:

```
final double duration = 1000;
final double thrust = 420;
final double isp = 318;
final Vector3D direction = Vector3D.PLUS_I;
final Frame frame = FramesFactory.getGCRF();
final ConstantThrustManeuver maneuver = new ConstantThrustManeuver(date,
duration, thrust, isp, direction, massProvider, "thruster", frame);
```

- In a local orbital frame frame:

```
final double duration = 1000;
final double thrust = 420;
final double isp = 318;
final Vector3D direction = Vector3D.PLUS_I;
final LOFType lof = LOFType.TNW;
final ConstantThrustManeuver maneuver = new ConstantThrustManeuver(date,
duration, thrust, isp, direction, massProvider, "thruster", lof);
```

#### Maneuver start and stop criterion can either be defined:

- With a start date and a duration (former method)
- With event detectors, first detector for starting the maneuver, second detector for stopping the maneuver. **Action.STOP** is required to trigger (start/stop) the thrust. Any other action will be discarded.

When using event detectors, pay attention to events that may occur several times. If you want the maneuver to perform only once, the event detectors have to be included in a nth occurrence detector or its `shouldBeRemoved()` method have to return `true`. Otherwise the maneuver may be performed more than once: if a maneuver starts at the perigee (using a perigee detector) and stops at the apogee (using an apogee detector), then a maneuver will start at every perigee and stop at every apogee.

**Parameter objects can also be used to define thrust and flow rate:**

```
final double duration = 1000;
final Parameter thrust = new Parameter("thrust", 420);
final Parameter flowRate = new Parameter("flow rate", 0.);
final Vector3D direction = Vector3D.PLUS_I;
final ConstantThrustManeuver maneuver = new ConstantThrustManeuver(date,
duration, thrust, flowRate, direction, massProvider, "thruster");
```

**At last, two methods are available in `ConstantThrustManeuver` to start/restart a propagation during a maneuver:**

- `isFiring()`
- `setFiring(final boolean isFiring)`

A user wanting to start a propagation during a maneuver will call `setFiring(true)` before launching the propagation.

This is useful since maneuver start and stop criteria are defined through event detectors. This mechanism is available for any non-impulsive maneuver i.e. `ConstantThrustManeuver`, `VariableThrustManeuver` and `ConstantThrustError`.

**Warning:** `ConstantThrustManeuver` can be defined using class `PropulsiveProperty`. In that case, ISP and thrust should be defined as constant. If defined as variable, no error will be thrown but behaviour may be erratic (in particular, ISP and Thrust cannot be defined as parameters).

## Variable Thrust Maneuver

The following diagram shows the classes and interfaces used to compute a variable thrust maneuver:

[Fichier:VariableThrustManeuver.png](#)

The `VariableThrustManeuver` class implements the `ForceModel` interface as the `ConstantThrustManeuver` class. The difference is that thrust, ISP and acceleration direction are constant values in `ConstantThrustManeuver`, whereas in `VariableThrustManeuver` they are represented by the following interfaces:

- `IDependentVariable<SpacecraftState>` (thrust and ISP)
- `IDependentVectorVariable<SpacecraftState>` (acceleration direction)

In this way thrust, ISP and thrust direction can be customized. In particular they can depend on spacecraft state.

Variable thrust maneuver are used in the same way as constant thrust maneuvers (see section above). In particular maneuver start and stop criterion can either be defined using a date and a duration (former method) or two event detectors.

## Impulse Maneuver

This `ImpulseManeuver` class implements the `EventDetector` interface.

This maneuver is provided with an underlying event detector; when an underlying event is triggered, and only if its action is set to `STOP`, the maneuver is triggered and the current `SpacecraftState` is reset. An increment or a decrement is applied to its velocity and a ratio is applied to the mass. These changes depend on the direction of variation of the integration variable (time) during integration : propagation or retro-propagation case.

## List of maneuvers

The class `ManeuversSequence` handles the creation and the operations on a list of maneuvers; the maneuvers that can be added to the list are the following ones :

- Impulse maneuvers;
- Continue constant maneuvers;
- Continue maneuvers with variable thrust and ISP;

The user can do the following operations on the list:

- add a new maneuver; to add it, the following conditions must be met:
  - there must be no superposition between the new maneuver and any maneuver in the list. If the new maneuver is an impulse one and its triggering event detector is not a date detector, it always adds the maneuver (there is no way to know in advance its date). Otherwise if the new maneuver is an impulse one and its triggering event detector is a date detector, it gets the date of the maneuver from the triggering detector and uses this information to check the superposition;
  - the time between the end of the previous maneuver and the start of the new maneuver, as well as the time between the end of the new maneuver and the start of the next maneuver, must be bigger than a threshold value. The threshold value depends on the type of maneuver (impulse or continue) and can be chosen by the user.
- remove a maneuver already in the list;
- get the number of maneuvers in the list;
- add all the maneuvers of the list to the propagator;

The following lines show the usage of the class:

```
final ManeuversSequence sequence = new ManeuversSequence(0.0, 0.0);  
// add the maneuvers to the sequence:  
sequence.add(constantManeuver);  
sequence.add(impulseManeuver);
```

```
// apply the maneuvers to the propagator:  
sequence.applyTo(numericalPropagator);
```

## Getting Started

[Modèle:SpecialInclusion prefix=\\$theme sub section="GettingStarted"/](#)

## Contents

### Interfaces

### Classes

Class	Summary	Javadoc
<b>ConstantThrustManeuver</b>	This class implements a simple maneuver with constant thrust.	...
<b>VariableThrustManeuver</b>	This class implements a maneuver with variable thrust.	...
<b>ImpulseManeuver</b>	This class implements an impulse maneuver model.	...
<b>SmallManeuverAnalyticalModel</b>	This class implements an analytical model for small maneuvers.	...
<b>ManeuversSequence</b>	This class implements a sequence of maneuvers.	...

Récupérée de « [http://patrius.cnes.fr/index.php?title=User\\_Manual\\_3.4.1\\_Maneuvers&oldid=1428](http://patrius.cnes.fr/index.php?title=User_Manual_3.4.1_Maneuvers&oldid=1428) »  
[Catégories](#) :

- [Pages avec des liens de fichiers brisés](#)
- [User Manual 3.4.1 Mission](#)

## Menu de navigation

### Outils personnels

- [3.135.247.17](#)
- [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 5 mars 2018 à 13:05.

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