

# User Manual 4.8 Interpolation Methods

De Wiki

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

[User Manual 4.8 Interpolation Methods](#)

## Introduction

### Scope

In this section, a focus is realised on the following interpolation methods: spline, bicubic, tricubic, Lagrange and Newton, covariance matrix and linear in 1D, 2D or 3D interpolation.

### Javadoc

The interpolation objects are available in the package `fr.cnes.sirius.patrius.math.analysis.interpolation` and in the package `fr.cnes.sirius.patrius.propagation.analytical.covariance`.

#### Library

#### Javadoc

Patrius [Package fr.cnes.sirius.patrius.math.analysis.interpolation](#)

Patrius [Package fr.cnes.sirius.patrius.math.analysis.interpolation](#)

Patrius [Package fr.cnes.sirius.patrius.propagation.analytical.covariance](#)

### Links

None as of now.

### Useful Documents

None as of now.

### Package Overview

The package `fr.cnes.sirius.patrius.math.analysis.interpolation` contains all the interpolation classes described in this section.



## Features Description

### Spline interpolation

The **spline interpolator** generates an interpolating function  $f(x): \mathbb{R} \rightarrow \mathbb{R}$ . The user gives as entries 2 sets of values, the values of  $x$ ,  $y$ . The interpolator gives the function  $f$  such as  $y=f(x)$ .

For the linear equation  $y=2x+1$

```
double x[] = { 0.0, 1.0, 2.0 };
```

```
double y[] = { 1.0, 3.0, 5.0 };
```

```
UnivariateInterpolator interpolator = new SplineInterpolator();  
UnivariateFunction function = interpolator.interpolate(x, y);  
double value = function .value(0.5);
```

## Bicubic interpolation

The ***bicubic interpolator*** generates an interpolating function  $f(x,y): \mathbb{R}^2 \rightarrow \mathbb{R}$ . The interpolator computes internally the coefficients of the bicubic function that is the interpolating function. The user gives as entries 3 sets of values, the values of x, y and z. The interpolator gives the function f such as  $z=f(x,y)$ .

For the equation of the plane  $z=2x-3y + 5$

```
double x[] = { 3, 4, 5, 6.5 };  
double y[] = {-4, -3, -1, 2, 2.5 };  
double z[][] = {{ 23, 20, 14, 5, 3.5 },  
                { 25, 22, 16, 7, 5.5 },  
                { 27, 24, 18, 9, 7.5 },  
                { 30, 27, 21, 12, 10.5 }};
```

```
BivariateGridInterpolator interpolator = new BicubicSplineInterpolator();  
BivariateFunction function = interpolator.interpolate(x, y, z);
```

## Tricubic interpolation

The ***tricubic interpolator*** generates an interpolating function  $f(x,y,z): \mathbb{R}^3 \rightarrow \mathbb{R}$ . The interpolator computes internally the coefficients of the tricubic function that is the interpolating function. The user gives as entries 4 sets of values, the values of x, y, z and w. The interpolator gives the function f such as  $w=f(x,y,z)$ .

For the equation of the plane  $w=2x- 3y - z + 5$

```
double x[] = { 3.0, 4.0, 5.0, 6.5 };  
double y[] = {-4.0, -3.0, -1.0, 2.0, 2.5 };  
double z[] = {-12.0, -8.0, -5.5, -3.0, 0.0, 2.5 };  
double w[][][] = {{{ 35, 31, 28.5, 26, 23, 20.5 },  
                   { 32, 28, 25.5, 23, 20, 17.5 },  
                   { 26, 22, 19.5, 17, 14, 11.5 },  
                   { 17, 13, 10.5, 8, 5, 2.5 },  
                   { 15.5, 11.5, 9, 6.5, 3.5, 1 }},  
                {{ 37, 33, 30.5, 28, 25, 22.5 },  
                 { 34, 30, 27.5, 25, 22, 19.5 },  
                 { 28, 24, 21.5, 19, 16, 13.5 },  
                 { 19, 15, 12.5, 10, 7, 4.8 },  
                 { 17.5, 13.5, 11, 8.5, 5.5, 3 }},  
                {{ 39, 35, 32.5, 30, 27, 24.8 },  
                 { 36, 32, 29.5, 27, 24, 21.5 },
```

```
{ 30, 26, 23.5, 21, 18, 15.5 },
{ 21, 17, 14.8, 12, 9, 6.5 },
{ 19.5, 15.5, 13, 10.5, 7.5, 5 }},
{{ 42, 38, 35.5, 33, 30, 27.5 },
{ 39, 35, 32.5, 30, 27, 24.8 },
{ 33, 29, 26.5, 24, 21, 18.5 },
{ 24, 20, 17.5, 15, 12, 9.5 },
{ 22.5, 18.5, 16, 13.5, 10.5, 8 }}}
```

```
TrivariateGridInterpolator interpolator = new TricubicSplineInterpolator();
TrivariateFunction function = interpolator.interpolate(x, y, z, w);
```

## Lagrange interpolation

The **Lagrange interpolator** generates an interpolating function  $f(x): \mathbb{R} \rightarrow \mathbb{R}$ . The user gives as entries 2 sets of values, the values of  $x, y$ . The interpolator gives the function  $f$  such as  $y=f(x)$ .

For the linear equation  $y=2x+1$

```
double x[] = { 0.0, 1.0, 2.0 };
double y[] = { 1.0, 3.0, 5.0 };
```

```
UnivariateFunction interpolator = new PolynomialFunctionLagrangeForm(x,y);
double value = interpolator.value(0.5);
```

## Newton interpolation

The **Newton interpolator** generates an interpolating function  $f(x): \mathbb{R} \rightarrow \mathbb{R}$ . The user gives as entries 2 sets of values, the coefficients  $c_i$  and the centers  $x_i$  such as the polynomial function  $P(x)=c_0 + c_1 (x - x_0) + \dots + c_n (x - x_n)$ . The interpolator gives the function  $f$  such as  $y=P(x)$ .

For the linear equation  $y=2x+1$

```
double c_i[] = { 3.0, 2.0 };
double x_i[] = { 1.0 };
```

```
UnivariateFunction interpolator = new PolynomialFunctionNewtonForm(c_i,x_i);
double value = interpolator.value(0.5);
```

## Covariance matrix interpolation

The purpose of this interpolation algorithm is to compute the covariance matrix at a given date through a simplified model of the transition matrix. When a covariance in PV coordinates is searched for an object orbiting around an celestial body, a simple dynamical model can be used, meaning limited to the newtonian attraction, plus a constant acceleration. The value of this constant acceleration will not change the transition matrix.

The transition matrix between a date  $t_1$  and a date  $t$  can be approximated :

- at order 0 : by  $\phi_1(t_1, t) = I_{\{3 \times 3\}}$
- at order 1 : by  $\phi_1(t_1, t) = I_{\{3 \times 3\}} + J_{\{PV\}} (t - t_1)$
- at order 2 : by  $\phi_1(t_1, t) = I_{\{3 \times 3\}} + J_{\{PV\}} (t - t_1) + 0.5 * J_{\{PV\}}^2 (t - t_1)^2$

where  $J_{\{PV\}} = \begin{pmatrix} 0_{\{3 \times 3\}} & I_{\{3 \times 3\}} \\ A & 0_{\{3 \times 3\}} \end{pmatrix}$ ,  $J_{\{PV\}}^2 = \begin{pmatrix} A & 0_{\{3 \times 3\}} \\ A & 0_{\{3 \times 3\}} \end{pmatrix}$  and  $A = -\frac{GM}{r^3} \left( I_{\{3 \times 3\}} - 3 \frac{PP^T}{r^2} \right)$ , where  $A$  is considered as a constant on the interval  $[t_1, t]$  and  $P$  is the satellite position vector.

We denote by  $M(t)$  the covariance matrix at instant  $t$ . Let  $t \in [t_1, t_2]$ . The transition matrices  $\phi_1(t_1, t)$  and  $\phi_2(t_2, t)$  are given by the above formula, and since matrix  $A$  is constant on  $[t_1, t_2]$ , we have that the covariance matrix at instant  $t$  is given by  $M(t) = (1 - \alpha) \phi_1(t_1, t) M(t_1) \phi_1^T(t_1, t) + \alpha \phi_2(t_2, t) M(t_2) \phi_2^T(t_2, t)$  with  $\alpha = \frac{t - t_1}{t_2 - t_1}$ .

## Linear interpolation

These classes allow linear piecewise interpolations in dimensions 1, 2 or 3.

### 1D interpolation

Let  $f$  be a real function  $\mathbb{R} \rightarrow \mathbb{R}$  and  $[x_1, x_2]$  the interpolation interval, where  $f(x_1), f(x_2)$  are known. For all  $x \in [x_1, x_2]$ , the interpolated value  $f(x)$  is given by  $f(x) = f(x_1) + (x - x_1) \frac{f(x_2) - f(x_1)}{x_2 - x_1}$ .

### 2D interpolation

The two dimensional interpolation will be two successive 1D interpolations. Let  $f$  be a real function  $\mathbb{R}^2 \rightarrow \mathbb{R}$  and  $[x_1, x_2] \times [y_1, y_2]$  the interpolation interval. First, a 1D interpolation in the  $y$  direction is made, leading to  $f(x, y_1) = f(x_1, y_1) + (y - y_1) \frac{f(x_2, y_1) - f(x_1, y_1)}{y_2 - y_1}$ ,

$$f(x, y_2) = f(x_1, y_2) + (y - y_1) \frac{f(x_2, y_2) - f(x_1, y_2)}{y_2 - y_1}.$$

Then a second 1D interpolation is made in the  $x$  direction with the previous two interpolated values :  $f(x, y) = f(x, y_1) + (x - x_1) \frac{f(x, y_2) - f(x, y_1)}{x_2 - x_1}$ .

### 3D interpolation

Let  $f$  be a real function  $\mathbb{R}^3 \rightarrow \mathbb{R}$  and  $[x_1, x_2] \times [y_1, y_2] \times [z_1, z_2]$  the interpolation interval. There will be

$2^3 - 1$  successive 1D interpolations.

$f(x,y,z)$  is interpolated from  $f(x,y,z_1)$  and  $f(x,y,z_2)$ .

$f(x,y,z_1)$  is interpolated from  $f(x,y_1,z_1)$  and  $f(x,y_2,z_1)$ .

$f(x,y,z_2)$  is interpolated from  $f(x,y_1,z_2)$  and  $f(x,y_2,z_2)$ .

$f(x,y_1,z_1)$  is interpolated from  $f(x_1,y_1,z_1)$  and  $f(x_2,y_1,z_1)$ .

$f(x,y_2,z_1)$  is interpolated from  $f(x_1,y_2,z_1)$  and  $f(x_2,y_2,z_1)$ .

$f(x,y_1,z_2)$  is interpolated from  $f(x_1,y_1,z_2)$  and  $f(x_2,y_1,z_2)$ .

$f(x,y_2,z_2)$  is interpolated from  $f(x_1,y_2,z_2)$  and  $f(x_2,y_2,z_2)$ .

## Getting Started

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

## Contents

### Interfaces

The library defines the following interfaces related to interpolation :

Interface	Summary	Javadoc
<b>UnivariateInterpolator</b>	Interface for a univariate interpolating function.	...
<b>BivariateGridInterpolator</b>	Interface for a bivariate interpolating function where the sample points must be specified on a regular grid.	...
<b>TrivariateGridInterpolator</b>	Interface for a trivariate interpolating function where the sample points must be specified on a regular grid.	...
<b>UnivariateFunction</b>	Interface for a univariate function	...

### Classes

This section is about the following classes related to interpolation :

Class	Summary	Javadoc
<b>SplineInterpolator</b>	Spline interpolator for a univariate real function.	...
<b>BicubicSplineInterpolator</b>	Bicubic spline interpolator for a bivariate real function.	...
<b>TricubicSplineInterpolator</b>	Tricubic spline interpolator for a trivariate real function.	...

<b>PolynomialFunctionLagrangeForm</b>	Lagrange interpolator, directly usable as a univariate real function.	<a href="#">...</a>
<b>PolynomialFunctionNewtonForm</b>	Newton interpolator, directly usable as a univariate real function.	<a href="#">...</a>
<b>Class</b>	<b>Summary</b>	<b>Javadoc</b>
<b>CovarianceInterpolation</b>	Interpolator of a covariance matrix based on its two surrounding covariance matrices.	<a href="#">...</a>
<b>OrbitCovariance</b>	Class containing a covariance matrix and its associated AbsoluteDate. New class replacing older class CovarianceMatrix	<a href="#">...</a>
<b>Class</b>	<b>Summary</b>	<b>Javadoc</b>
<b>CovarianceInterpolation</b>	Interpolator of a covariance matrix based on its two surrounding covariance matrices.	<a href="#">...</a>
<b>OrbitCovariance</b>	Class containing a covariance matrix and its associated AbsoluteDate. New class replacing older class CovarianceMatrix	<a href="#">...</a>
<b>Class</b>	<b>Summary</b>	<b>Javadoc</b>
<b>AbstractLinearIntervalsFunction</b>	Abstract class for linear interpolations.	<a href="#">...</a>
<b>UniLinearIntervalsFunction</b>	Linear one-dimensional function.	<a href="#">...</a>
<b>BiLinearIntervalsFunction</b>	Linear two-dimensional function.	<a href="#">...</a>
<b>TriLinearIntervalsFunction</b>	Linear three-dimensional function.	<a href="#">...</a>
<b>UniLinearIntervalsInterpolator</b>	Interpolator of linear one-dimensional functions.	<a href="#">...</a>
<b>BiLinearIntervalsInterpolator</b>	Interpolator of linear two-dimensional functions.	<a href="#">...</a>
<b>TriLinearIntervalsInterpolator</b>	Interpolator of linear three-dimensional functions.	<a href="#">...</a>

Récupérée de

« [http://patrius.cnes.fr/index.php?title=User\\_Manual\\_4.8\\_Interpolation\\_Methods&oldid=3022](http://patrius.cnes.fr/index.php?title=User_Manual_4.8_Interpolation_Methods&oldid=3022) »

[Catégorie](#) :

- [User Manual 4.8 Mathematics](#)

## Menu de navigation

### Outils personnels

- [52.15.238.221](#)
- [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 20 octobre 2021 à 13:50.



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