

# Оглавление

1	Интерфейс библиотеки . . . . .	2
1.1	Функция <code>interpolate</code> . . . . .	2

# 1 Интерфейс библиотеки

Разработанная библиотека предоставляет 3 функции:

`(interpolate points type & options)`

`(interpolate-parametric points type & options)`

`(interpolate-grid grid type & options)`

Все вышеперечисленные функции имеют похожую сигнатуру. Они принимают следующие параметры:

- `points` или `grid` - набор точек или сетка, которые требуется интерполировать;
- `type` - тип интерполяции;
- `options` - дополнительные опции, специфичные для каждой функции и типа интерполяции;

Все функции возвращают новую функцию - интерполирующую функцию, с помощью которой находятся значения в интересующих точках.

## 1.1 Функция `interpolate`

Данная функция позволяет построить интерполирующую функцию  $f(x) = y$  по заданному набору точек  $(x_i, y_i)$ . Точки могут задаваться в любом порядке, перед использованием они будут отсортированы по координате  $x$ . Данная функция поддерживает следующие типы интерполяции: линейная, полиномиальная, кубический сплайн, кубический Эрмитов сплайн, среднеквадратичное приближения. Соответствующие аргументы для параметра `type`: `:linear`, `:polynomial`, `:cubic`, `:cubic-hermite`, `:linear-least-squares`.

Дополнительные опции:

- `:boundaries` - граничные условия для кубического сплайна. Поддерживаются 2 вида условий: естественные (`:natural`) и замкнутые (`:closed`);
- `:derivatives` - производные для кубического Эрмитова сплайна;
- `:basis`, `:n`, `:degree` - опции настройки среднеквадратичного приближения. Позволяют задать базис, произвольный или один из 2 встроенных (полиномиальный и В-сплайны); число функций в базисе, если выбран встроенный; степень В-сплайнов;

Пример:

Построение кубического сплайна с замкнутыми граничными условиями по точкам  $(0, 0)$ ,  $(1, 3)$ ,  $(2, 0)$ ,  $(4, 2)$ :

```
(def points [[0 0] [1 3] [2 0] [4 2]])

(def cubic (interpolate points :cubic :boundaries :closed))

(cubic 0) ; 0.0
(cubic 1) ; 3.0
(cubic 3) ; -1.2380952380952381
```