

# Package ‘opensimplex2’

March 29, 2026

**Title** Generate Multi-Dimensional Open Simplex Noise

**Version** 0.0.3

**Description** Generate 2, 3 or 4-dimensional gradient noise.  
The noise function is comparable to classic Perlin noise,  
but with less directional artefacts and lower computational overhead.  
It can have applications in procedural generation or (flow fields)  
simulations.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**LinkingTo** cpp11

**Depends** R (>= 3.5.0)

**Suggests** gifski, knitr, ragg, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://pepijn-devries.github.io/opensimplex2/>,  
<https://github.com/pepijn-devries/opensimplex2/>

**BugReports** <https://github.com/pepijn-devries/opensimplex2/issues>

**Language** en-GB

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Pepijn de Vries [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-7961-6646>>),  
Marco Ciaramella [aut, cph],  
KdotJPG [aut, cph]

**Maintainer** Pepijn de Vries <[pepijn.devries@outlook.com](mailto:pepijn.devries@outlook.com)>

**Repository** CRAN

**Date/Publication** 2026-03-29 15:50:03 UTC

## Contents

opensimplex_noise . . . . .	2
opensimplex_space . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

opensimplex_noise	<i>Get a 2, 3, or 4 Dimensional Array of Simplex Noise</i>
-------------------	--

---

### Description

Create a regular n-dimensional grid with an OpenSimplex2 noise gradient. You can control the noisiness to some degree with the frequency argument. If you want more control, you should use [opensimplex\\_space\(\)](#), which allows you to create a continuous OpenSimplex noise gradient space, that can be sampled at any arbitrary coordinate.

### Usage

```
opensimplex_noise(type = "S", width, height, depth, slice, frequency = 1)
```

### Arguments

type	Type of OpenSimplex2 you wish to use. Should be either "F" for fast or "S" for smooth.
width, height, depth, slice	Positive integer size of each of the desired dimensions. width and height dimensions are required. Other dimensions are optional. However, depth is required when slice dimension is specified.
frequency	The frequency (numeric) with which the noise gradient fluctuates with respect to each respective dimension. Low values (<1) will generate smooth gradients, whereas large values (>1) will result in very noisy gradients. Default value is 1.

### Details

The exact state of the noise gradient space depends on R's internal, random generator. So each time you call [opensimplex\\_space](#), you will get a space in a different state. If you want to obtain a reproducible state, you simply set the random seed with [set.seed\(\)](#).

### Value

A matrix (in case of two dimensions) or array (in case of more dimensions) of numeric values between -1 and +1. OpenSimplex2 uses gradient tables that ensure that the distribution of values is centred at zero, meaning the "peaks" and "valleys" are statistically balanced.

### Examples

```
mat <- opensimplex_noise("S", 100, 100)
image(mat)
```

---

opensimplex\_space      *Get a 2, 3, or 4 Dimensional OpenSimplex Noise Gradient Field*

---

## Description

Create a continuous OpenSimplex noise gradient space, that can be sampled at any arbitrary coordinate. The gradient has numeric values between -1 and +1. OpenSimplex2 uses gradient tables that ensure that the distribution of values is centred at zero, meaning the "peaks" and "valleys" are statistically balanced.

The exact state of the noise gradient space depends on R's internal, random generator. So each time you call `opensimplex_space`, you will get a space in a different state. If you want to obtain a reproducible state, you simply set the random seed with `set.seed()`.

## Usage

```
opensimplex_space(type = "S", dimensions = 2L)
```

## Arguments

type	Type of OpenSimplex2 you wish to use. Should be either "F" for fast or "S" for smooth.
dimensions	An integer value of number of dimensions to be used in your gradient space. Should be 2, 3 or 4.

## Value

Returns a named list. It has the elements:

- `sample`: which is a function that will return the simplex noise value (between -1 and +1) at a given coordinate. The function takes the same number of arguments as the value of `dimensions`. Each argument is a coordinate in the  $n^{th}$  dimension. You can provide multiple coordinates, as long as you provide the same number of coordinates in each dimension.
- `dimensions`: an integer value indicating the number of dimensions available in this space object.
- `type`: A character string indicating whether this object uses the fast ("F") or smooth ("S") variant of OpenSimplex2.
- `close`: which is a function that closes the simplex noise gradient space. In other words, memory used to specify the gradient space is freed, and can no longer be accessed. After closing the space, you can no longer sample it.

## Examples

```
## By setting a random generator seed, the example below becomes reproducible
set.seed(0)
```

```
## Open a extra smooth ("S") simplex noise gradient with 3 dimensions:
```

```
space <- opensimplex_space("S", 3L)

## Sample it at some random coordinates
space$sample(i = 5*runif(10), j = 10*runif(10), k = 15*runif(10))

## Close it when you are done
space$close()
```

# Index

`opensimplex_noise`, [2](#)  
`opensimplex_space`, [3](#)  
`opensimplex_space()`, [2](#)  
`set.seed()`, [2](#), [3](#)