

---

# **numba-scipy**

***Release 0.3.1+0.g19c39ca.dirty***

**Anaconda, Inc.**

**Jun 30, 2022**



**CONTENTS:**

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>User Manual</b>  | <b>3</b>  |
| <b>2</b> | <b>Reference Manual</b>                                       | <b>5</b>  |
| 2.1      | Supported functions from <code>scipy.special</code> . . . . . | 5         |
| <b>3</b> | <b>Contributing to numba-scipy</b>                            | <b>11</b> |
| 3.1      | Communication . . . . .                                       | 11        |
| 3.2      | Getting set up . . . . .                                      | 11        |
| 3.3      | Development rules . . . . .                                   | 13        |
| <b>4</b> | <b>Release Notes</b>  | <b>15</b> |
| 4.1      | Version 0.3.1 . . . . .                                       | 15        |
| 4.2      | Version 0.3 . . . . .   | 15        |
| 4.3      | Version 0.2 . . . . .   | 15        |
| 4.4      | Version 0.1 . . . . .   | 16        |
|          | <b>Index</b>  | <b>17</b> |



This is the numba-scipy documentation. Unless you are already acquainted with numba-scipy perhaps start with the *User manual*.



## USER MANUAL

The numba-scipy user manual.





## REFERENCE MANUAL

The numba-scipy reference manual.

### 2.1 Supported functions from `scipy.special`

The following functions are supported:

- `scipy.special.agm` Supported signature(s): `float64(float64, float64)`
- `scipy.special.bdtr` Supported signature(s): `float64(float64, float64, float64)`, `float64(long_, long_, float64)`
- `scipy.special.bdtrc` Supported signature(s): `float64(float64, float64, float64)`, `float64(long_, long_, float64)`
- `scipy.special.bdtri` Supported signature(s): `float64(float64, float64, float64)`, `float64(long_, long_, float64)`
- `scipy.special.bdtrik` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.bdtrin` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.bei` Supported signature(s): `float64(float64)`
- `scipy.special.beip` Supported signature(s): `float64(float64)`
- `scipy.special.ber` Supported signature(s): `float64(float64)`
- `scipy.special.berp` Supported signature(s): `float64(float64)`
- `scipy.special.besselpoly` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.beta` Supported signature(s): `float64(float64, float64)`
- `scipy.special.betainc` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.betaincinv` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.betaln` Supported signature(s): `float64(float64, float64)`
- `scipy.special.binom` Supported signature(s): `float64(float64, float64)`
- `scipy.special.boxcox` Supported signature(s): `float64(float64, float64)`
- `scipy.special.boxcox1p` Supported signature(s): `float64(float64, float64)`
- `scipy.special.btdtr` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.btdtri` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.btdtria` Supported signature(s): `float64(float64, float64, float64)`

- `scipy.special.btdtrib` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.cbirt` Supported signature(s): `float64(float64)`
- `scipy.special.chdtr` Supported signature(s): `float64(float64, float64)`
- `scipy.special.chdtrc` Supported signature(s): `float64(float64, float64)`
- `scipy.special.chdtri` Supported signature(s): `float64(float64, float64)`
- `scipy.special.chdtriv` Supported signature(s): `float64(float64, float64)`
- `scipy.special.chndtr` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.chndtridf` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.chndtrinc` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.chndtrix` Supported signature(s): `float64(float64, float64, float64)`
- `scipy.special.cosdg` Supported signature(s): `float64(float64)`
- `scipy.special.cosm1` Supported signature(s): `float64(float64)`
- `scipy.special.cotdg` Supported signature(s): `float64(float64)`
- `scipy.special.dawsn` Supported signature(s): `float64(float64)`
- `scipy.special.ellipse` Supported signature(s): `float64(float64)`
- `scipy.special.ellipseinc` Supported signature(s): `float64(float64, float64)`
- `scipy.special.ellipkinc` Supported signature(s): `float64(float64, float64)`
- `scipy.special.ellipkm1` Supported signature(s): `float64(float64)`
- `scipy.special.entr` Supported signature(s): `float64(float64)`
- `scipy.special.erf` Supported signature(s): `float64(float64)`
- `scipy.special.erfc` Supported signature(s): `float64(float64)`
- `scipy.special.erfcx` Supported signature(s): `float64(float64)`
- `scipy.special.erfi` Supported signature(s): `float64(float64)`
- `scipy.special.eval_chebyc` Supported signature(s): `float64(float64, float64), float64(long_, float64)`
- `scipy.special.eval_chebys` Supported signature(s): `float64(float64, float64), float64(long_, float64)`
- `scipy.special.eval_chebyt` Supported signature(s): `float64(float64, float64), float64(long_, float64)`
- `scipy.special.eval_chebyu` Supported signature(s): `float64(float64, float64), float64(long_, float64)`
- `scipy.special.eval_gegenbauer` Supported signature(s): `float64(float64, float64, float64), float64(long_, float64, float64)`
- `scipy.special.eval_genlaguerre` Supported signature(s): `float64(float64, float64, float64), float64(long_, float64, float64)`
- `scipy.special.eval_hermite` Supported signature(s): `float64(long_, float64)`
- `scipy.special.eval_hermitenorm` Supported signature(s): `float64(long_, float64)`

- `scipy.special.eval_jacobi` Supported signature(s): float64(float64, float64, float64, float64), float64(long\_, float64, float64, float64)
- `scipy.special.eval_laguerre` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.eval_legendre` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.eval_sh_chebyt` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.eval_sh_chebyu` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.eval_sh_jacobi` Supported signature(s): float64(float64, float64, float64, float64), float64(long\_, float64, float64, float64)
- `scipy.special.eval_sh_legendre` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.exp1` Supported signature(s): float64(float64)
- `scipy.special.exp10` Supported signature(s): float64(float64)
- `scipy.special.exp2` Supported signature(s): float64(float64)
- `scipy.special.exp1` Supported signature(s): float64(float64)
- `scipy.special.expit` Supported signature(s): float64(float64), float32(float32)
- `scipy.special.expm1` Supported signature(s): float64(float64)
- `scipy.special.expn` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.exprel` Supported signature(s): float64(float64)
- `scipy.special.fdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.fdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.fdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.fdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.gamma` Supported signature(s): float64(float64)
- `scipy.special.gammamainc` Supported signature(s): float64(float64, float64)
- `scipy.special.gammamaincc` Supported signature(s): float64(float64, float64)
- `scipy.special.gammamainccinv` Supported signature(s): float64(float64, float64)
- `scipy.special.gammamaincinv` Supported signature(s): float64(float64, float64)
- `scipy.special.gammaln` Supported signature(s): float64(float64)
- `scipy.special.gammaln` Supported signature(s): float64(float64)
- `scipy.special.gdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.gdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.gdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.gdt` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.gdt` Supported signature(s): float64(float64, float64, float64)

- `scipy.special.huber` Supported signature(s): float64(float64, float64)
- `scipy.special.hyp0f1` Supported signature(s): float64(float64, float64)
- `scipy.special.hyp1f1` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.hyp2f1` Supported signature(s): float64(float64, float64, float64, float64)
- `scipy.special.hyperu` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.i0` Supported signature(s): float64(float64)
- `scipy.special.i0e` Supported signature(s): float64(float64)
- `scipy.special.i1` Supported signature(s): float64(float64)
- `scipy.special.i1e` Supported signature(s): float64(float64)
- `scipy.special.inv_boxcox` Supported signature(s): float64(float64, float64)
- `scipy.special.inv_boxcox1p` Supported signature(s): float64(float64, float64)
- `scipy.special.it2struve0` Supported signature(s): float64(float64)
- `scipy.special.itmodstruve0` Supported signature(s): float64(float64)
- `scipy.special.itstruve0` Supported signature(s): float64(float64)
- `scipy.special.iv` Supported signature(s): float64(float64, float64)
- `scipy.special.ive` Supported signature(s): float64(float64, float64)
- `scipy.special.j0` Supported signature(s): float64(float64)
- `scipy.special.j1` Supported signature(s): float64(float64)
- `scipy.special.jv` Supported signature(s): float64(float64, float64)
- `scipy.special.jve` Supported signature(s): float64(float64, float64)
- `scipy.special.k0` Supported signature(s): float64(float64)
- `scipy.special.k0e` Supported signature(s): float64(float64)
- `scipy.special.k1` Supported signature(s): float64(float64)
- `scipy.special.k1e` Supported signature(s): float64(float64)
- `scipy.special.kei` Supported signature(s): float64(float64)
- `scipy.special.keip` Supported signature(s): float64(float64)
- `scipy.special.ker` Supported signature(s): float64(float64)
- `scipy.special.kerp` Supported signature(s): float64(float64)
- `scipy.special.kl_div` Supported signature(s): float64(float64, float64)
- `scipy.special.kn` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.kolmogi` Supported signature(s): float64(float64)
- `scipy.special.kolmogorov` Supported signature(s): float64(float64)
- `scipy.special.kv` Supported signature(s): float64(float64, float64)
- `scipy.special.kve` Supported signature(s): float64(float64, float64)
- `scipy.special.log1p` Supported signature(s): float64(float64)
- `scipy.special.log_ndtr` Supported signature(s): float64(float64)

- `scipy.special.loggamma` Supported signature(s): float64(float64)
- `scipy.special.logit` Supported signature(s): float64(float64), float32(float32)
- `scipy.special.lpmv` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.mathieu_a` Supported signature(s): float64(float64, float64)
- `scipy.special.mathieu_b` Supported signature(s): float64(float64, float64)
- `scipy.special.modstruve` Supported signature(s): float64(float64, float64)
- `scipy.special.nbdtr` Supported signature(s): float64(float64, float64, float64), float64(long\_, long\_, float64)
- `scipy.special.nbdtrc` Supported signature(s): float64(float64, float64, float64), float64(long\_, long\_, float64)
- `scipy.special.nbdtri` Supported signature(s): float64(float64, float64, float64), float64(long\_, long\_, float64)
- `scipy.special.nbdtrik` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.nbdtrin` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.ncfdtr` Supported signature(s): float64(float64, float64, float64, float64)
- `scipy.special.ncfdtri` Supported signature(s): float64(float64, float64, float64, float64)
- `scipy.special.ncfdtridfd` Supported signature(s): float64(float64, float64, float64, float64)
- `scipy.special.ncfdtridfn` Supported signature(s): float64(float64, float64, float64, float64)
- `scipy.special.ncfdtrinc` Supported signature(s): float64(float64, float64, float64, float64)
- `scipy.special.nctdtr` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.nctdtridf` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.nctdtrinc` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.nctdtrit` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.ndtr` Supported signature(s): float64(float64)
- `scipy.special.ndtri` Supported signature(s): float64(float64)
- `scipy.special.nrdtrimn` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.nrdtrisd` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.obl_cv` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.owens_t` Supported signature(s): float64(float64, float64)
- `scipy.special.pdtr` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.pdtrc` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.pdtri` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.pdtrik` Supported signature(s): float64(float64, float64)
- `scipy.special.poch` Supported signature(s): float64(float64, float64)
- `scipy.special.pro_cv` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.pseudo_huber` Supported signature(s): float64(float64, float64)

- `scipy.special.psi` Supported signature(s): float64(float64)
- `scipy.special.radian` Supported signature(s): float64(float64, float64, float64)
- `scipy.special.rel ENTR` Supported signature(s): float64(float64, float64)
- `scipy.special.rgamma` Supported signature(s): float64(float64)
- `scipy.special.round` Supported signature(s): float64(float64)
- `scipy.special.sindg` Supported signature(s): float64(float64)
- `scipy.special.smirnov` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.smirnovi` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.spence` Supported signature(s): float64(float64)
- `scipy.special.stdtr` Supported signature(s): float64(float64, float64)
- `scipy.special.stdtridf` Supported signature(s): float64(float64, float64)
- `scipy.special.stdtrit` Supported signature(s): float64(float64, float64)
- `scipy.special.struve` Supported signature(s): float64(float64, float64)
- `scipy.special.tandg` Supported signature(s): float64(float64)
- `scipy.special.tklmbda` Supported signature(s): float64(float64, float64)
- `scipy.special.xlog1py` Supported signature(s): float64(float64, float64)
- `scipy.special.xlogy` Supported signature(s): float64(float64, float64)
- `scipy.special.y0` Supported signature(s): float64(float64)
- `scipy.special.y1` Supported signature(s): float64(float64)
- `scipy.special.yn` Supported signature(s): float64(float64, float64), float64(long\_, float64)
- `scipy.special.yv` Supported signature(s): float64(float64, float64)
- `scipy.special.yve` Supported signature(s): float64(float64, float64)
- `scipy.special.zetac` Supported signature(s): float64(float64)

## CONTRIBUTING TO NUMBA-SCIPY

Contributions to numba-scipy are always welcomed! Even simple documentation improvements are encouraged. If you have questions, don't hesitate to ask them (see below).

### 3.1 Communication

#### 3.1.1 Contact

The Numba community uses Discourse for asking questions and having discussions about numba-scipy. There are various categories available and it can be reached at: [numba.discourse.group](https://numba.discourse.group). There is a category for [numba-scipy](#).

#### 3.1.2 Real-time Chat

numba-scipy uses Gitter for public real-time chat. To help improve the signal-to-noise ratio, there are two channels:

- [numba/numba](#): General discussion, questions, and debugging help.
- [numba/numba-dev](#): Discussion of PRs, planning, release coordination, etc.

Both channels are public.

Note that the Github issue tracker is the best place to report bugs. Bug reports in chat are difficult to track and likely to be lost.

#### 3.1.3 Bug tracker

The [Github issue tracker](#) is used to track both bug reports and feature requests.

### 3.2 Getting set up

If you want to contribute, it's best to fork the [Github repository](#), then create a branch representing your work. When your work is ready, you should submit it as a pull request from the Github interface.

If you want, you can submit a pull request even when you haven't finished working. This can be useful to gather feedback, or to stress your changes against the [continuous integration](#) platform. In this case, please prepend [WIP] to your pull request's title.



### 3.2.1 Build environment

numba-scipy has a number of dependencies (mostly [Numba](#) and [SciPy](#)). Unless you want to build those dependencies yourself, it's recommended you use [conda](#) to create a dedicated development environment and install pre-compiled versions of those dependencies there.

First add the Anaconda Cloud numba channel so as to get development builds of the numba library:

```
$ conda config --add channels numba
```

Then create an environment with the right dependencies:

```
$ conda create -n numba-scipy python=3.7 scipy numba
```

---

**Note:** This installs an environment based on Python 3.7, but you can of course choose another version supported by Numba.

---

To activate the environment for the current shell session:

```
$ conda activate numba-scipy
```

---

**Note:** These instructions are for a standard Linux shell. You may need to adapt them for other platforms.

---

Once the environment is activated, you have a dedicated Python with the required dependencies.

### 3.2.2 Building numba-scipy

For a convenient development workflow, it's recommended that you build numba-scipy inside its source checkout:

```
$ git clone git://github.com/numba/numba-scipy.git
$ cd numba-scipy
$ python setup.py develop
```

### 3.2.3 Running tests

numba-scipy is validated using a test suite comprised of various kind of tests (unit tests, functional tests). The test suite is written using the standard [unittest](#) framework and rely on [pytest](#) for execution. The [pytest](#) package will need installing to run the tests, using [conda](#) this can be achieved by:

```
$ conda install pytest
```

The tests can then be executed via `python -m pytest`.



## 3.3 Development rules

### 3.3.1 Code reviews

Any non-trivial change should go through a code review by one or several of the core developers. The recommended process is to submit a pull request on github.

A code review should try to assess the following criteria:

- general design and correctness
- code structure and maintainability
- coding conventions
- docstrings, comments
- test coverage

### 3.3.2 Coding conventions

All Python code should follow [PEP 8](#). Code and documentation should generally fit within 80 columns, for maximum readability with all existing tools (such as code review UIs).

### 3.3.3 Stability

The repository's master branch is expected to be stable at all times. This translates into the fact that the test suite passes without errors on all supported platforms (see below). This also means that a pull request also needs to pass the test suite before it is merged in.

### 3.3.4 Platform support

Every commit to the master branch is automatically tested on a selection of platforms. [Azure](#) is used to provide public continuous integration information for as many combinations as can be supported by the service. If you see problems on platforms with which you are unfamiliar, feel free to ask for help in your pull request. The numba-scipy core developers can help diagnose cross-platform compatibility issues.

### 3.3.5 Documentation

This documentation is under the docs directory of the [numba-scipy repository](#). It is built with [Sphinx](#), which is available using conda or pip.

To build the documentation, you need the bootstrap theme:

```
$ pip install sphinx_bootstrap_theme
```

You can edit the source files under docs/source/, after which you can build and check the documentation:

```
$ make html
$ open _build/html/index.html
```



## RELEASE NOTES

### 4.1 Version 0.3.1

This is a maintenance release that:

- Updates the maximum SciPy dependency to 1.7.3.
- Updates Azure CI.

Contributors:

- Benda Xu
- Siu Kwan Lam
- Valentin Haenel

### 4.2 Version 0.3

This release adds:

- Dynamic generation of `scipy.special.signatures` (with thanks to Brandon T. Willard)
- Github Actions CI

Importantly, the dynamic generation of the signatures now makes numba-scipy compatible with multiple SciPy releases simultaneously.

Contributors:

- Brandon T. Willard
- Valentin Haenel

### 4.3 Version 0.2

This release adds the start of real functionality and builds out the project infrastructure.

- Support for a large number of `scipy.special` functions (with thanks to Josh Wilson).
- Entry point registration system in use to auto-register this as a Numba extension project.
- Documentation.
- CI builds.

- Conda package.

Contributors:

- Josh Wilson
- Stan Seibert (core dev)
- Stuart Archibald (core dev)

## 4.4 Version 0.1

This is the initial demo release.

- Initial Release.
- Demonstrative support for a couple of example `scipy.special` functions.
- Basic project infrastructure.

Contributors:

- Stuart Archibald (core dev)

## INDEX

### P

Python Enhancement Proposals

PEP 8, [13](#)