Project Reference

Author

April 2022

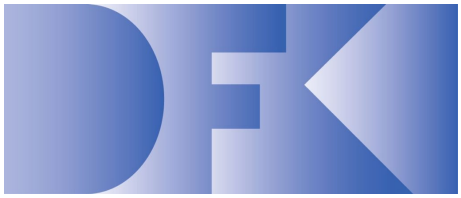


TABLE OF CONTENTS

INTRODUCTION

The following is a demonstration of what you can achieve by using Sphinx and reStructuredText to document your projects.

The capabilities of reStructuredText and Sphinx will be discussed and demonstrated, as well as best practices for writing and generating documentation with Sphinx.

Sphinx allows you to document your project with the docstrings in your code. You will find here what you must know to produce technical documents from your code. Lastly, you will see the documentation generated from the example source code in this repository.

GUIDANCE:

Primer

EXECUTIVE SUMMARY

Sphinx reStructuredText supports LaTeX math, citations, cross-refernces, figures, lists, tables, file inclusion, text highlight markup and more.

Read on to get started!

CHAPTER

ONE

PRIMER

References

For a more in-depth and possibly up-to-date description of the capabilities of reStructuredText and Sphinx, refer to the Sphinx reStructuredText primer.

Other references:

- reStructuredText reference: figures
- reStructuredText reference: directives

reStructuredText allows you flexibility close to that of LaTeX with terser syntax. Both LaTeX and HTML are rendered from it, abstracting content from typesetting. Sphinx reStructuredText supports LaTeX math, citations, cross-refernces, lists, figures, tables, text highlight markup, file inclusion and more.

Benefits

- Write a **singular**, **pleasant to read** documentation source for your project.
- Customize LaTeX and HTML output with **content agnostic** LaTeX style files and HTML+CSS templates (starting with Sphinx's) **usable across all your projects**.

And lastly,

- Document your project directly from the **docstrings** of your code.

What follows is a demonstration of the fundamental capabilities of Sphinx reStructuredText for technical writing, as well as best practices for writing and generating documentation with Sphinx.

Much of the same will be demonstrated again in *Your Project*, with a twist: *Your Project* is generated directly from the source code in this proejct.

MIND THE BACKTICKS

reStructuredText makes use of **backticks** (U+0060 *GRAVE ACCENT*) for its commands.

Not to be confused for apostrophes (')!

Robotics Innovation Center
Robert-Hooke-Str. 1
28359 Bremen
Deutschland
robotik@dfki.de

1.1 Math

```
# Inline equations
:math:`<inline equation>`

# Unnumbered block equations
.. math::
    :label: <equation label>

    <block equation>

# Numbered block equations
.. math::
    :label: <equation label>

    <block equation>

# Block equations in environment of choice
.. math::
    :label: <equation label>
    :nowrap:

    \begin{environment of choice}
        <block equation>
    \end{environment of choice}
```

Inline equation: $e^{\pi i} = -1$.

Block equations. Unlabeled, and thus unnumbered,

$$e^{\pi i} = -1$$

and labeled and thus numbered (1.1).

$$e^{\pi i} = -1 \tag{1.1}$$

By default Sphinx will place your LaTeX equations inside of an `equation` LaTeX environment. You can override this with the `:nowrap:` attribute of the `.. math::` directive. As HTML math consists of your LaTeX equations rendered and exported in SVG format, `:nowrap:` equations will display normally in HTML.

The equation below lies within a user-specified LaTeX environment.

$$\begin{aligned} \max_{\rho,\lambda} \quad & \rho \\ & (x^T x)^d (V(x) - \rho) + \lambda(x)\dot{V}(x) \quad is \quad SOS \end{aligned}$$

BEWARE

User-specified equation environments will break equation labeling in LaTeX. In particular, it seems that Sphinx will internally replace the reference to the new equation by a reference to the previous equation. This happens both for numbered and unnumbered LaTeX equation environments.

If you are considering user-specified equation environments, make absolutely sure you cannot obtain what you desire within the `equation` environment.

If you wish to do so anyway, and **wish to label your equations** for cross-referencing in your **site documentation** at the expense of that in LaTeX form, make sure to **use an unnumbered equation environment** such as `equation*`:

```
.. math::
    :label: equation2
    :nowrap:

    \begin{equation*}
        \begin{aligned}
        \max_{\rho,\lambda} \quad & \rho \\
                                  & (x^Tx)^d(V(x) - \rho) + \lambda(x)\dot{V}
↪(x) \quad is \quad SOS
        \end{aligned}
    \end{equation*}
```

Otherwise, the HTML rendering of your equations will display two numberings: that assigned to it by the LaTeX renderer used to generate the site's SVG equations, and that later assigned to it by Sphinx.

1.2 Figures

Figures must be included in `source/figures` and referenced by their path **relative to the current reStructuredText file**. Sphinx has two figure directives: `image` and `figure`.

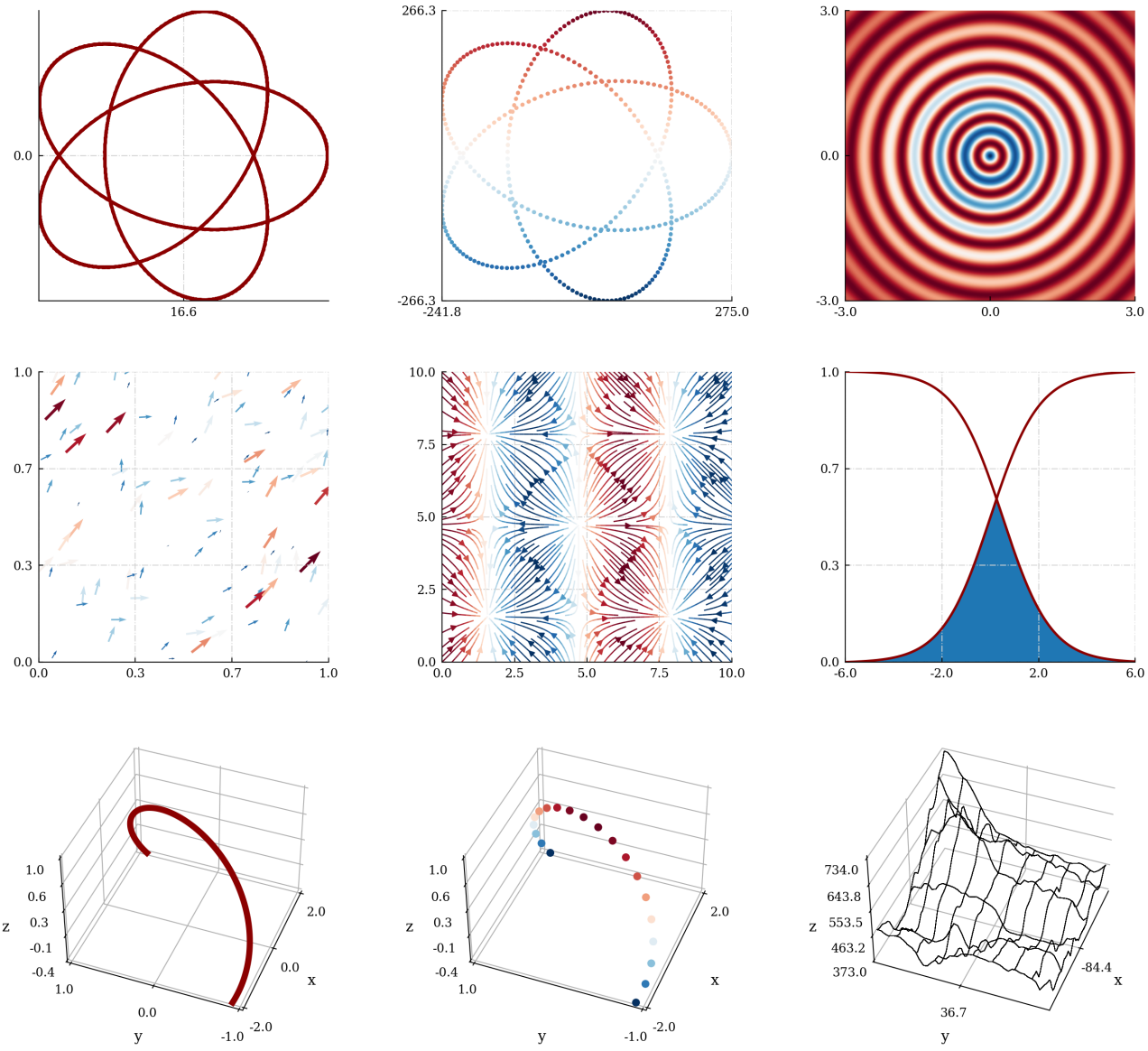
1.2.1 `image`

Uncaptioned figures. Allow side-by-side arrangements.

```
# Single image
.. image:: ../../figures/demo.png

# Side by side arrangement
|pic1| |pic2|

.. |pic1| image:: ../../figures/demo.png
:width: 45%

.. |pic2| image:: ../../figures/demo.png
:width: 45%
```

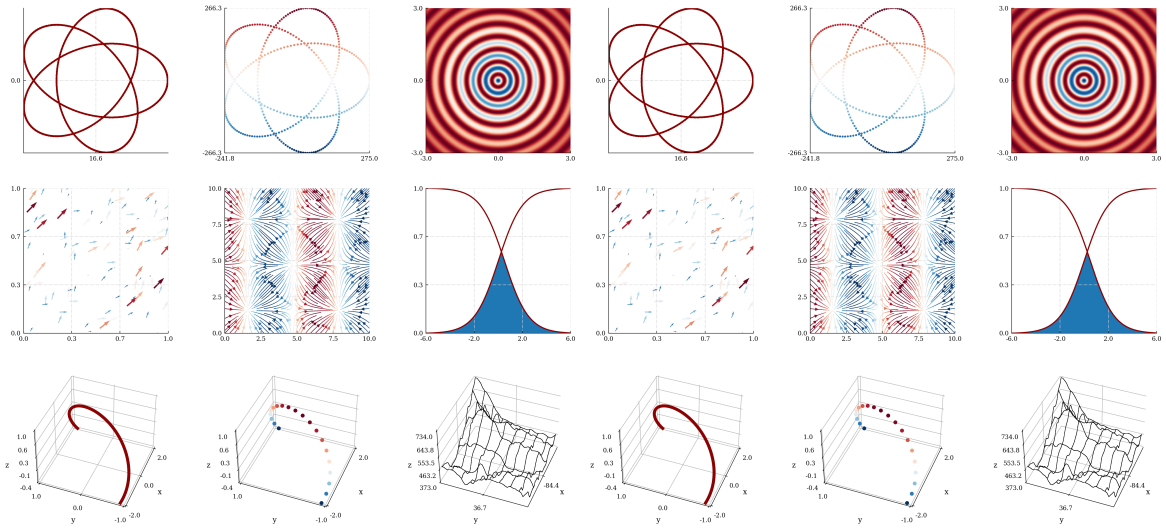
Single image:

Robotics Innovation Center
Robert-Hooke-Str. 1
28359 Bremen
Deutschland
robotik@dfki.de



Side by side:

Robotics Innovation Center
Robert-Hooke-Str. 1
28359 Bremen
Deutschland
robotik@dfki.de



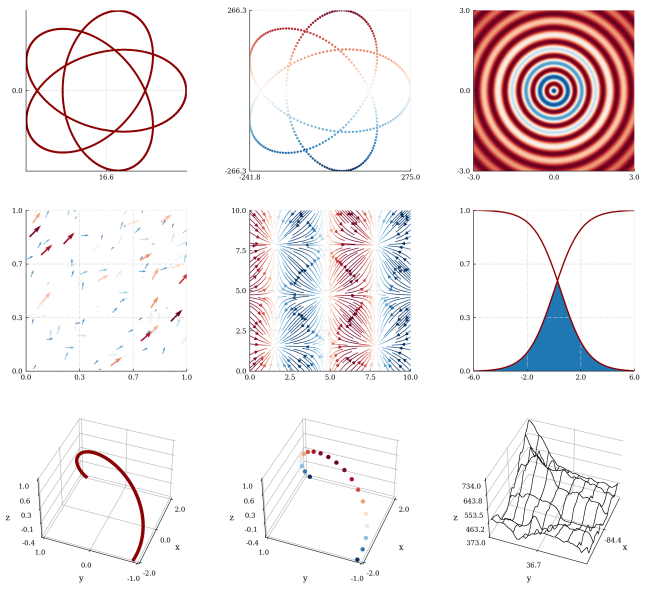
1.2.2 `figure`

Captioned figures. Do not allow side-by-side arrangements.

```
# Figure without caption
.. figure:: ../../figures/demo.png
    :width: 50%
    :align: center

# Figure with caption, labeled "figure1"
.. _figure1:
.. figure:: ../../figures/demo.png
    :width: 50%
    :align: center

    Caption
```

Figure without caption or label.

Robotics Innovation Center
Robert-Hooke-Str. 1
28359 Bremen
Deutschland
robotik@dfki.de



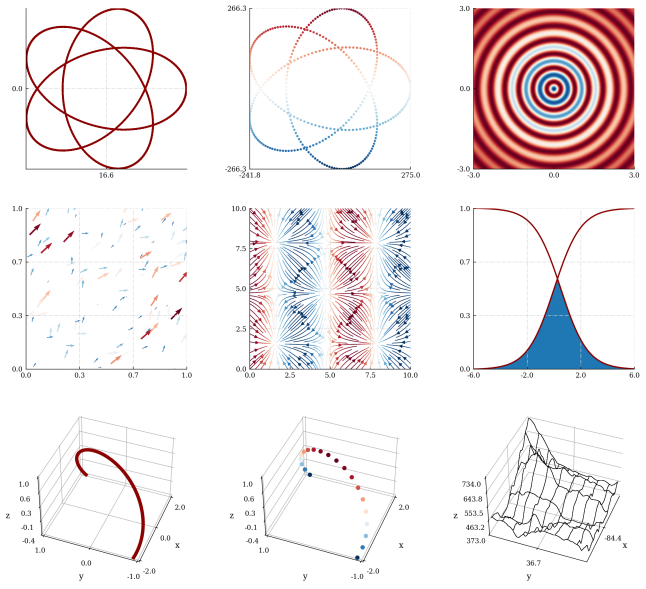


Figure 1.1: Captioned and labelled figure.

1.3 Tables

Tables may be input in list or grid format. With conventional editors the list format is most convenient. Certain editors may greatly ease generating grid tables, making them a viable option (ie: Emacs Org Mode).

```
# Centered table
.. _table1:
.. list-table:: Example table.
    :widths: auto
    :header-rows: 1
    :align: center

    * - **Header 1**
      - **Header 2**
      - **Header 3**
      - **Header 4**
    * - A
      - B
      - C
      - D
```

Table 1.1: Example table.

Header 1	Header 2	Header 3	Header 4
A	B	C	D

1.4 References

Sphinx supports citations, cross-references and hyperlinks.

1.4.1 Citations

```
# In-text
:cite:t:`Tedrake`

# Parenthetic
:cite:p:`Tedrake`
```

You can cite Tedrake [2] in text, or parenthetic fashion [2].

1.4.2 Cross-references

Note: If you wish to create cross-references to paragraphs, in-text targets, etc. consult the *Sphinx cross-referencing guide <https://docs.readthedocs.io/en/stable/guides/cross-referencing-with-sphinx.html>*.

```
# Figures
:numref:`<figure label>`

# Equations
:eq:`<equation label>`

# Sections
:ref:`<Section Name>`
```

- Table 1.1
- Figure 1.1
- Equation (1.1)
- *Your Project*

Warning: Only the **top level section labels** are automatically created by `sphinx.autosectionlabel`. If you wish to reference a subsection, create a label to it as follows.

```
.. _Subsection1:

Subsection 1
------------
```

1.4.3 Hyperlinks

```
`Link text <url>`_
```

1.5 Markup

Note: For a more in-depth discussion of the markup supported by Sphinx reStructuredText, refer to the Sphinx reStructuredText primer.

reStructuredText supports different forms of text markup.

1.5.1 Inline

```
# Italic
*italic*

# Bold
**bold**

# Math
`math`

# Code
`code`
```

1.5.2 Blocks

Code

Note: All supported languages are listed in the pygments languages index.

```
.. code-block:: <language>

    <code>
```

1.6 Admonitions

Important: You can use admonitions to bring attention to important information.

1.6.1 Default

Note: For the full list of reStructuredText admonitions, visit the docutils reStructuredText admonitions reference.

Quoting from the Sphinx directives reference, *Most themes style only "note" and "warning" specially*.

---

reStructuredText provides a set of default admonitions. The syntax is the following:

```
.. important::

    Of import
```

(continues on next page)

(continued from previous page)

```
.. note::

    Note

.. warning::

    Warning

.. danger::

    Danger
```

1.6.2 Custom

Custom Admonition

A dangerous admonition.

You can use the base `.. admonition::` directive to create custom admonitions based on the predetermined admonition CSS classes, or custom ones provided by yourself.

```
.. admonition:: Custom Admonition
    :class: danger

    A dangerous admonition.
```

1.7 Extensions

It is possible to extend Sphinx reStructuredText to suit unsatisfied needs.

This can be done by creating new

- `classes`,
- `:roles:`, and
- `.. directives::`

It is easiest to extend Sphinx's HTML rendering capabilities by creating new CSS classes.

BEWARE

Extensions using CSS classes will have no effect on LaTeX output. New roles or directives are necessary to such an effect.

Creating new roles and directives is more involved. For creating new roles, refer to the blog post by Doug Hellmann. For creating new directives, heed Sphinx forums and the online guidance you may find.

The following are the extensions present in this Sphinx configuration.

1.7.1 Classes

Center

Center HTML elements. Works with text as well as side-by-side `images`, section headings, and perhaps any other HTML element.

```
.. rst-class:: center

    Centered elements.
```

CHAPTER

TWO

REQUIREMENTS

Important: Sphinx needs to complete two tasks to generate a documentation site and report for your project:

- Sphinx needs to successfully **scan your source code** for documentation in the form of docstrings, and
- Sphinx needs to successfully **render your documentation**.

Care for the structure of your source code and the syntax of your documentation sources is fundamental.

2.1 Source code structure

Say that you want to create documentation from the docstrings in your code. **Two conditions**:

- Your project must be a Python **package**
- Your **package** must have a `__version__` global variable

Global variables

You can declare the `__version__` in the `__init__.py` of your project's Python package.

If your project consists of a single script, you can declare it anywhere in your script.

2.2 Writing docstrings successfully

- Use raw strings.
- Write reStructuredText (*Primer*).

CHAPTER

THREE

CONFIGURATION

Sphinx must be configured to:

1. Find your project's source code
2. Correctly display your project's information

3.1 `project.py`

This file includes the information that Sphinx needs to build a documentation site and LaTeX report for your project, as well as the fundamental details of your project that will be rendered across the site and report.

CODE GENERATION

These variables must be correctly set, or Sphinx will fail to run.

- **`codename`** Name of your project's Python package.
- **`codedir`** Code directory relative to your projects source directory, where the source directory is understood to be the **parent of the docs directory**.

PROJECT INFORMATION

- `project`
- `author`
- `report_title`
- `report_author`
- **`logo`** Logo filename relative to the `source/figures` directory.

Robotics Innovation Center
Robert-Hooke-Str. 1
28359 Bremen
Deutschland
robotik@dfki.de

3.2 `project.sty`

This file contains information about your project as it will be rendered in LaTeX.

PROJECT INFORMATION

- `project`
- `institution`
- `logo`
- `colorHeader`

CHAPTER

FOUR

YOUR PROJECT

It's usually convenient to write a short introduction in your `__init__.py`.

NOTE

This project is:

- A Python package
- Its version is specifed in its `__init__.py`

4.1 Main Module

`lonely_function``()`

As promised, what follows is a demonstration of docstring documentation.

You can add **bold** and *italic* text, text with *math typesetting* and `source code`. You can cite your sources either in text, such as Tedrake [2], or in parenthetic fashion [1],

- Bullet points work as well

and so does LaTeX math, unlabeled and labeled, as you can see in (4.1)!

$$f(n) = \begin{cases} n/2, & \text{if } n \text{ is even} \\ 3n+1, & \text{if } n \text{ is odd} \end{cases}$$

$$f(n) = \begin{cases} n/2, & \text{if } n \text{ is even} \\ 3n+1, & \text{if } n \text{ is odd} \end{cases} \tag{4.1}$$

Table syntax is rather weird but manageable.

Table 4.1: Example table.

Header 1	Header 2	Header 3	Header 4
A	B	C	D

You can add pictures if you want too. Importantly, when you are generating documentation from your code:

- Place your figures in the **docs/source/figures directory**.
- Include them with their path relative to the source. That is, **figures/<your figure>.<ext>**.

Unfortunately the figure doesn't nicely fit in this page. Fortunately, we can break page to try and have our discourse flow anyway.

Robotics Innovation Center
Robert-Hooke-Str. 1
28359 Bremen
Deutschland
robotik@dfki.de

Check it:

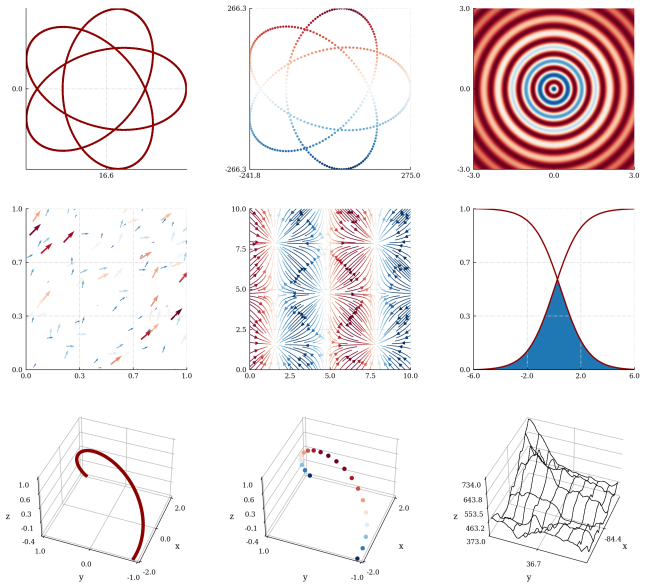


Figure 4.1: You're free to set alignment, figure width relative to the text's, etc. Don't trust Sphinx to go too far. If you need something complex use proper LaTeX code inside a `.. raw:: latex` block.

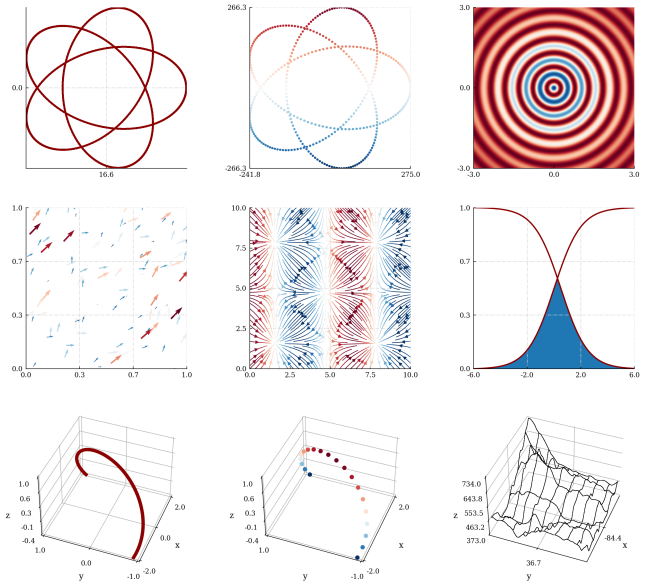


Figure 4.2: Adding a cross-reference to Table 1.1 or Figure 1.1 works much as it would in LaTeX.

4.2 A Subpackage

As before, this package's `__init__.py` is the home of this introduction.

4.2.1 A Module Within

h()
: A function which is part of one of the modules of a subpackage of your project.

g()
: A second function in the module.

f()
: The three functions in this module are stowed in this order:

  1. h
  2. g
  3. f

  As you can see, functions are rendered in the order in which they are written in their source. Nice.

4.3 `__init__.py`

some_API_function()
: You might wanna keep commonly used utilities or high level interfaces to your package here for ease of import. A way to achieve the same effect is by importing functions defined in your package from this file.

BIBLIOGRAPHY

[1] Jared Di Carlo. *Software and control design for the MIT Cheetah quadruped robots*. PhD thesis, Massachusetts Institute of Technology, 2020.

[2] Russ Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). 2021. Downloaded on 28.07.2021 from http://underactuated.mit.edu/.

PYTHON MODULE INDEX

INDEX