

Complexity: Theory and Computational Models - Installation guideline for software

Claudius Gräbner¹ and Torsten Heinrich²

¹Johannes Kepler University

²University of Oxford

August 4-11 2017, Neudietendorf
Version 1.1. of April 10, 2017

For the course, the following software is required:

1. A python3 engine
2. The python modules numpy, scipy and matplotlib, and
3. An editor or IDE
4. The Jupyter package

These packages can be installed separately or as an integrated distribution. We detail now which options are available for the most common operating systems; all the software packages we recommend are free and open source. There are also proprietary options.

It is absolutely essential that the software is running the first day of the course. If you have problems with the installation, please contact use before the workshop!

We will also provide you with a test script. Please follow the instructions below to test whether your installation works properly and let us know if not. Claudius will offer a troubleshooting session during the evening before the first session, please get there if you have still problems with your installation.

Contents

1	Install Python and the required packages	2
1.1	For Mac OS X and macOS	2
1.1.1	Install Anaconda	2
1.1.2	Install packages individually	2
1.2	For Windows	2
1.2.1	Install Anaconda	2
1.2.2	Install packages individually	3
1.3	For Linux	3
1.3.1	Individual Packages (python engine and libraries)	3
1.3.2	As a Distribution	4
2	Install a suitable editor or IDE	4
2.1	Windows and macOS	4
2.2	Linux	4
2.2.1	Debian/Ubuntu	4
2.2.2	Fedora	4

2.2.3	Archlinux	5
2.2.4	Gentoo	5
2.2.5	Other	5
3	Install Jupyter	5
3.1	Windows and macOS	5
3.2	Linux	6
4	Run the test script	6

1 Install Python and the required packages

1.1 For Mac OS X and macOS

Mac typically already includes a Python engine and interpreter (cPython). However, this will be a version 2.X, not 3.X. As the syntax is different, we strongly recommend to install a Python 3.X version alongside Python 2.X.

We suggest installing the Anaconda package which already includes all the libraries you will need for scientific computing. But it is also possible to install all packages individually.

1.1.1 Install Anaconda

Visit the following website and follow the [installation instructions](#). Please make sure that you have Python3, and not Python2!

1.1.2 Install packages individually

First, make sure you have the package manager MacPorts installed. If not, just visit this website and follow the instructions: <https://www.macports.org/install.php>

Then follow these steps to install Python3:

1. Open a terminal
2. Run

```
sudo port install py35-numpy py35-scipy py35-matplotlib
```

Note that 35 indicates version number 3.5 (i.e. Python 3.5). You may have to change this number (3.6, 3.7 would be more recent) but it should be consistent between all packages.

1.2 For Windows

1.2.1 Install Anaconda

Visit the following website and follow the [installation instructions](#). Please make sure that you have Python3, and not Python2! You may also use another distribution such as [WinPython](#).

1.2.2 Install packages individually

Windows unfortunately does not have a package manager, this makes it difficult to install packages individually so that they match and work well together. Installing a distribution might be the preferred option.

If you want to install individual packages anyway, you need to download the python engine from the [python website](#).

The other packages are offered as binary for instance on [Christoph Grohlke's website](#).

Please be very careful which websites you trust and which websites you download software from.

1.3 For Linux

Many Linux distributions come with the python engine/interpreter (cPython) installed. It varies, however, which version (2.X or 3.X) it is. All packages are preferably installed using the distribution's proper package manager since it will then be included in future system updates. All common package managers offer the packages we require.

1.3.1 Individual Packages (python engine and libraries)

Debian/Ubuntu type distributions The standard python distribution is typically Python 2.X, hence you have to install Python 3.X additionally:

```
apt-get install python3 python3-numpy python3-scipy python3-matplotlib
```

Fedora like distributions Recent versions of Fedora have the python3 engine preinstalled as /usr/bin/python (If for some reason not, do "sudo yum install python") The modules are available by:

```
sudo yum install python3-numpy python3-scipy python3-matplotlib
```

Archlinux like distributions Python 3.X is the standard python engine of the distribution. The modules can be installed like so:

```
pacman -Sv python-numpy, python-scipy, python-matplotlib
```

Gentoo like distributions Make sure that the appropriate version of Python 3.X is in in your USE FLAGS or defined in your portage make.conf (see [here](#)). Then:

```
emerge -pv python numpy scipy matplotlib
```

```
emerge -v python numpy scipy matplotlib
```

Others We can, of course not list instructions for all distributions. Most package managers feature the required packages. Otherwise, it is still possible to install modules using python's own pip package manager. Note that e.g. on older versions of Redhat/Fedora/CentOS, python3 (and associated modules) are not part of the official repositories. You may need to compile and install them manually. There are manuals for this [manuals for this](#).

1.3.2 As a Distribution

You may install python as a distribution. This has the disadvantage that you will have several installations on your system for many packages, which reverses one of the advantages of fully open and integrated systems. It does, however, have the advantage that you can install it completely in userspace, i.e. without root privileges.

Anaconda is an example, [see the website](#).

2 Install a suitable editor or IDE

Just select and install a suitable editor or IDE. There are many options. You may find a list of possible choices [here \(IDEs\)](#) or [here \(editors\)](#).

Some options we could recommend include

- [Spyder](#)
- [Geany](#)
- [Emacs](#)

2.1 Windows and macOS

Installation packages can usually be downloaded directly from the editor/IDE websites. Note, however, that at least for Spyder the installation will become tricky if you do not install Anaconda (see [here](#)). And Emacs can be installed on macOS via the following terminal command (provided you use the macOS package manager MacPorts):

```
sudo port install emacs
```

2.2 Linux

Linux users may start with the following package manager commands, but keep in mind that some further system configuration may become necessary:

2.2.1 Debian/Ubuntu

```
apt-get install python3-spyder
```

```
apt-get install geany
```

```
apt-get install emacs
```

...

2.2.2 Fedora

```
yum install python3-spyder
```

```
yum install geany
```

```
yum install emacs
```

...

2.2.3 Archlinux

```
pacman -Sv spyder
```

```
pacman -Sv geany
```

```
pacman -Sv emacs
```

...

2.2.4 Gentoo

```
emerge -pv spyder
```

```
emerge -v spyder
```

```
emerge -pv geany
```

```
emerge -v geany
```

```
emerge -pv emacs
```

```
emerge -v emacs
```

You may need to set `ACCEPT_KEYWORDS` appropriately for portage to allow the use of packages in testing.

2.2.5 Other

The availability and maintenance of IDEs and editor packages differs between distributions. Check out what is available for your distributions. Some editors can also be installed with pip; jupyter is an example, see [here](#).

3 Install Jupyter

You need to make sure that jupyter is installed. In case you installed Python3 via Anaconda, Jupyter is already installed.

In case you did not installed Anaconda you need to install Jupyter via your package manager (Linux) or via the Python package manager pip (macOS and Windows). You may test whether everything is working fine by starting a notebook according to: [this manual](#).

Let us know if there are any error messages!

3.1 Windows and macOS

Jupyter must be installed via the python package manager pip. It comes with all new python versions. To install jupyter via pip just follow the instructions [here](#).

3.2 Linux

Linux users may install jupyter via their respective package managers, for Ubuntu/Debian-like distributions, for example, this would be:

```
apt-get install python3-jupyter
```

For Arch-like distributions it would be:

```
pacman -Sv jupyter
```

And for Gentoo-like distributions:

```
emerge -pv jupyter
```

```
emerge -v jupyter
```

Please check out tutorials in the WWW for your particular distribution.

4 Run the test script

Please download our test script [from the course page](#).

Save the file in a folder and execute it using Python. To do this, you use a terminal to get to the location of the file and type:

```
python test-complexity.py
```

If you get the message that everything is all right, great! You are well prepared for the course.

If not, please send the error message via email to [Claudius](#).

We are looking forward meeting you in Neudietendorf!

Claudius & Torsten