

2021 Sagan Summer Workshop

Circumstellar Disks and Young Planets

Python Getting Started Guide

For the workshop hands-on sessions, participants can choose to run the exercises from their own installation of Python or use Google Colabory notebooks. These instructions are for the Python notebooks.

Please note that *you should set up your Python distribution in advance of the workshop*; just follow the instructions below. In this document, we use the `Courier New` font to indicate a command you type into the command prompt.

The Python Distribution

In order to run the Jupyter notebooks used in the hands-on session, you first need a working installation of Python3. Even if your operating system comes with a system level Python distribution installed (e.g. most flavors of Linux include this), we recommend you follow these instructions to install the standalone Anaconda Python distribution. System level distributions are often far out of date, and are difficult to upgrade without breaking the entire operating system.

You can download the free Anaconda here: <https://www.anaconda.com/products/individual>

Versions exist for Windows, Mac, and Linux. *Choose the Python 3.8 download appropriate for your operating system.* Note that the Individual Edition is free, while the Team and Enterprise distributions require license agreements. For the Sagan Workshop, and most data analysis you would do for independent research, the Individual Edition is sufficient.

These downloads are between 300 and 600 MB, depending on operating system. If your internet is limited in bandwidth, you can utilize the Miniconda distribution, which is much smaller (~50 MB) at <https://docs.conda.io/en/latest/miniconda.html>. This gives you the same core packages as Anaconda, but strips out many extra packages that may not be needed. If you do need missing components later on, it is straightforward to add them individually (see page 5 for a list of the main packages you would have to install).

Note: If you already have Python installed on your system, but would like to install the new packages in their own virtual environment, you can choose to install Python in a custom working environment (see the instructions on page 5).

Windows  MacOS 

The Windows and Mac versions of Anaconda and Miniconda are installed in the normal way by double-clicking the downloaded installer. Follow the prompts to install the Python distribution. Be patient as the installation can take as much as 15-20 min to complete.

You will interact with Python from a terminal prompt. On macOS you can reach the terminal from your Applications/Utilities directory. On Windows, use the search feature next to the start menu to search for Anaconda Navigator. From there open the Powershell to access the command line.

Linux

The Linux version is installed by running the 'sh' script from the command line in a terminal:
`bash Anaconda3-2021.05-Linux-x86_64.sh`

After accepting the license agreement, you will be asked for a location into which to install the distribution (e.g. the default `/home/username/anaconda3`). All of the Python packages we will be using will be installed under the directory you specified. This will take quite some time to complete. When you are asked "Do you wish the installer to initialize Anaconda3 by running `conda init`?", answer "yes". This will add code to your `.bashrc` file which initializes Anaconda Python whenever you open a new terminal window.

Now close the terminal and open another (so that the new `.bashrc` takes effect).

Update to the Latest Version

Next update the core conda package to the latest version by typing:
`conda update conda`

Answer "y" when asked if you wish to update, and wait for the update to finish.

If you would like to update *all* of the packages that come with Anaconda to their latest versions, type:

```
conda update --all
```

Note that this will take some time. This step is optional, since packages can also be updated individually as needed.

You now have a working Python distribution installed in your account.

Install Packages and Download Files for the Workshop

We will use a few additional Python packages that are not included in the Anaconda distribution. You should install them by typing:

```
conda install bs4
```

If not already installed:

```
conda install astropy
```

Disk Models Hands-on Session:

(These installation instructions are adapted from the ones posted at https://www.ita.uni-heidelberg.de/~dullemond/software/radmc-3d/manual_rmcpy/index.html; you can check out that website for more details if needed.)

Download radmc3d-2.0 from:

<https://github.com/dullemond/radmc3d-2.0>

On that page, click on the green “Code” button, and select “Download ZIP”. This will download the code in a zip file. Extract the file by double-clicking on it.

You should now have a directory called “radmc3d-2.0-master”, which contains several subdirectories. We will only use radmc3dPy. So, change to the directory `radmc3d-2.0-master/python/radmc3dPy` and install radmc3dPy with

```
python setup.py install
```

You may have to instead use

```
sudo python setup.py install (if you need to install as administrator)
```

or

```
python setup.py install --user (if you can only install software for one user)
```

This will install radmc3dPy to Python’s site-packages directory. (See the note at the end of this section if this step does not complete successfully.)

Next you will download the files for the disks that will be studied; download the contents of each of these four folders:

- HH 30: <https://caltech.box.com/s/v3kq0tnkthu7ame1faxx2apye2ya9r1v>
- HL Tau: <https://caltech.box.com/s/ts5gp2ytgel2t8xzt2qkwhwcfllcvlel>
- HR 4796: <https://caltech.box.com/s/ro3r19pbbkowitzm31ur5xl44z2kkep6>
- PDS 70: <https://caltech.box.com/s/z4eva5dhog70ucx65st52gagw152ipcz>

Each of the folders contains various sub-folders and also one Jupyter notebook (and its backup copy). You will use one Jupyter notebook for each disk that will be studied during this hands-on session.

Note: If running the ‘setup.py’ script in the ‘radmc3dPy’ directory does not work, you can simply copy the `radmc3dPy` directory to the same directory where the Jupyter notebook you are working on is located; in this way the notebook will find the radmc3dPy scripts used for the plots.

Young Planets Spectroscopy Hands-on Session:

Follow the instructions posted here: <https://natashabatalha.github.io/picaso/installation.html>

The instructions above include the installation of PICASO (you can either install with `pip` or with `git`), downloading a large file (6 GB) with opacities and another file (17 MB) with the Castelli & Kurucz (2004) stellar atlas.

If you install with `pip`, you will have to download the PICASO “reference” and “docs” folder.

To do so, go to

<https://github.com/natashabatalha/picaso>

and click on the green “Code” button, then select “Download ZIP”. This will download the code in a zip file. Extract the file by double-clicking on it; under the main directory, you will find the “reference” and “docs” folders.

Next, download the Sonora Grid of Models (there are four files, one of which is 1.2 GB in size):

<https://zenodo.org/record/1309035#.YMvJU5IAQ>

You should place the files in a folder (it does not matter which folder) and untar the `profile.tar` and `spectra.tar` files. In the Jupyter notebooks that will be used for this hands-on session, there will be a variable called “sonora_profile_db” which should be set to the path of the “profile” directory (which is the directory with the files from `profile.tar`).

After you have installed PICASO, you may find it helpful to look through the “Basics of Thermal Emission” tutorial, which walks you through computing a thermal emission spectrum (it has two parts, one on thermal emission from exoplanets and one on brown dwarfs):

<https://natashabatalha.github.io/picaso/tutorials.html#basics-of-thermal-emission>

This tutorial is optional, but may be helpful if you are not familiar with thermal emission spectra.

You can find the four Jupyter notebooks that will be used during the hands-on session in the following subdirectory of Picaso:

`picaso/docs/notebooks/workshops/SaganSchool2021`

The first two (`1_Spectroscopy.ipynb`, `2_HotVsCold.ipynb`) will be used on Wednesday and Thursday, respectively, while the other two (`3_Chemistry.ipynb`, `4_Clouds.ipynb`) are optional. Before working with the Jupyter notebooks, create a backup copy of each of them.

Custom Working Environment (*Optional*)

If you are developing code or running a program that requires you to install custom packages, it can be useful to create a virtual environment dedicated to that project. This environment allows you to create local installations of your Python environment for specific purposes; if you do something undesirable in a virtual environment, you can delete it and start over, without having to reinstall conda.

Note: This step is optional for the Sagan Workshop. It can be useful for users who are already using Python for other projects, and want to keep the workshop materials separated out.

To create a virtual Python3 environment for use with the Sagan Workshop, type the following:
`conda create --name sagan python=3.8`

This creates a directory `~/anaconda3/envs/sagan` (C:\Users\username\anaconda3\envs\sagan under Windows), which contains a local copy of your Python directory. You can modify the Python environment in here without risk of damaging the base install.

To ‘activate’ your custom environment, type:

```
conda activate sagan
```

The beginning of your terminal prompt should change to `[sagan]`, and any packages that you install now will end up in `~/anaconda3/envs/sagan`. Note that each new terminal you open needs to have the activate command run on it.

To ‘deactivate’ a custom environment, type:

```
conda deactivate or conda activate
```

(with no environment specified). That will return your terminal to the base environment.

By default, conda environments contain only a minimal set of packages. If you are working in an environment, you will need to install packages as necessary for the workshop; here are the main ones:

```
conda activate sagan
conda install numpy
conda install scipy
conda install astropy
conda install matplotlib
conda install jupyter
conda install pandas
conda install numba
conda install bs4
conda install bokeh
conda install PyMieScatt
conda install requests
```

Python Jupyter Notebooks

Jupyter notebooks are a handy, interactive way to work through programming problems in a GUI environment in your web browser. They allow you to easily document your work (just like a real lab notebook) and visualize the results of each step in a computer program. The activities for the Sagan Workshop are in the form of Jupyter notebooks (files ending in 'ipynb'), so if you are not familiar with them, read the following instructions.

Jupyter comes included with the full Anaconda distribution.

If you installed miniconda, run the following to add the Jupyter notebook package:

```
conda install jupyter
```

You can start running a Jupyter notebook by typing the following command in a terminal window (you should type this in the directory where the Jupyter notebook is located, or any directory above it):

```
jupyter notebook
```

 *Note: Windows users can use Anaconda Navigator to access the Jupyter notebook interface.*

This will launch the Jupyter kernel and load an interface in your local web browser where you can see the content of the directory where you started Jupyter. To open a Jupyter notebook, simply click on the file (it ends in 'ipynb'), and the notebook will open up in a new browser window. To execute code in a Jupyter notebook cell, type CTRL-Enter (or Shift-Enter). Note that Jupyter notebooks are automatically saved, so it is helpful to create a backup copy of any notebook before working with it.

When you are done, click on the Jupyter icon  in the upper left. This will return you to the directory interface, where you can load additional notebooks.

To close a Jupyter notebook, in the directory interface select the notebook and click on "Shutdown"; close the browser window. Then, if you launched the notebook from a terminal window, press Control-C twice in that window to shut down the notebook server. Another option to shut down the server is to click on "Quit" in the top right corner of the directory interface page and close the browser window.