Resources: conda, sage-math, and jupyter

George McNinch

2024-02-14

Using `SageMath` for computer algebra calculations

Blurb:

> `SageMath` is a free open-source mathematics software system licensed under the GPL. It builds on top of many existing open-source packages: NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R and many more. Access their combined power through a common, Python-based language or directly via interfaces or wrappers.

Moreover, `SageMath` can be run inside `jupyter` notebooks; since it combines `code` and exposition, this format can be useful e.g. for classroom demonstrations.

`CoCalc`

You can interact with a `SageMath` kernel "in the cloud" at `CoCalc`. For example, you can open `jupyter notebooks` whose kernel is `SageMath` via `CoCalc` (I'm going to provide links that should enable you to do this).

Installation on "your" computer

There are a fair number of parts required to get a `SageMath` installation working, so I recommend installing it using `conda` (short for *anaconda*; together with `python`, this amounts to a bunch of snakes …).

`conda` describes itself as follows:

> Initially started as a multi-platform package management tool, the term "conda" has since evolved to encompass an entire open-source packaging ecosystem and philosophy. This ecosystem is supported by many organizations who all share the common goal of providing easier access to programming tools and libraries.

install `conda` via `miniforge`

Specifically, I recommend installing `conda` using something called the `miniforge` – here is the `miniforge` github repository.

Here is the list of *installers*. The installer must be chose to match your computer's architecture and operating system.

- If you have a recent Apple MacBook ("M1" or "M2") your architecture is likely `arm64` and you need to use the `OS X arm64 (Apple Silicon)` installer.
- somewhat older Apple MacBooks need to use the `OS X x86_64` installer.
- Windows users need to use the `Windows x86_64` installer.
- There are installers for `linux` users as well.

You should now read and follow the installation instructions for your operating system. (I've provided some annotation below, which you might read first).

installing in Mac OS or Linux

The Mac OS & Linux instructions tell you to execute the following commands in a terminal window (i.e. a "shell"):

```
~$ curl -L -O "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-$(uname)-$

~$ bash Miniforge3-$(uname)-$(uname -m).sh
```

Note that these should be executed in a shell– in particular, you don't type the prompts ("~$"). The first command uses the utility curl to download the relevant installer (the expressions $(uname) and $(uname- m) should select the correct installer for your computers architecture + operating system, without you having to make the choice).

The second command (bash...) *executes* the installer that you just downloaded. When it is running, you'll need to interact a bit with it (pressing "ENTER" when requested, accepting the licence terms, confirming the install location, etc.)

Once the installer is done downloading packages from the conda-forge, you will be asked:

```
Do you wish the installer to initialize Miniforge3
by running conda init? [yes|no]
```

In order to use conda, you should answer yes. It will add some code to your config file ~/.bashrc.

After closing and reopening your shell, you should have a prompt that looks something like the following:

```
(base) george@valhalla:~$
```

The (base) prefix indicates that the base environment of conda has been activated.

Use conda to install sage in a new environment

After installing conda, run the following command in a shell:

```
$ mamba create -n sage sage python=3.11.6
...
```

Now activate the new sage environment you just created:

$ mamba activate sage …

Your terminal prompt should now look something like

```
(sage) george@valhalla:~$
```

At minimum you need to install jupyterlab

```
(sage) george@valhalla:~$  mamba install jupyterlab
...
```

Now if you *run* jupyterlab, a server will be started – running on your computer – and a tab in your browser will open with a jupyter session where you can interact with jupyter notebooks.

```
(sage) george@valhalla:~$  jupyter-lab
...
```

(You can exit this server by typing C-c ("control C") in the terminal in which jupyter-lab is running…)

If you save ipynb files from class, you can *open* them in this jupyterlab browser tab.

Bibliography