An introduction to scientific programming with

Session 3.3:
Packaging your code

Using your own modules

- Easiest: run python from folder(s) containing your modules
  - Typical approach for simple single-purpose scripts
  - But, code cannot refer to the module it is within
  - What about using the code from another project?
- Next easiest: edit PYTHONPATH
  - But, what about sharing your code?
- Best: package your code
  - approach has changed over time (setup.py / setup.cfg)
  - current preferred method (pyproject.toml):
    - Use hatch, flit or poetry
  - See https://hatch.pypa.io/ https://packaging.python.org/

Creating a package

- Basically:
  - Put your code in a folder with an __init__.py file
  - In folder below, create a pyproject.toml file
    - This specifies the build system and the project metadata
  - You can then install your module with pip:

```
$ pip install -e folder_containing_pyproject
```

  - Now importable anywhere just like the standard library
  - The -e means the installation is ‘editable’
    - No need to update if you change the code

Distributing a package

- Once you have created a package it is easy to upload it to PyPI
  - with build and twine (standard)
  - or with hatch, etc:

- And then convert this for Anaconda
  - using conda skeleton pypi