HWR 582: Python & FloPy

So far in this class, we have been working with Groundwater Vistas.
Groundwater Vistas is a **GUI** (Graphical User Interface) that allows you to interact through pointing and clicking with MODFLOW, which is a **compiled executable program** created by the USGS using the **programming language** FORTRAN. This means that you cannot edit MODFLOW itself, only feed it various different inputs. The user guide for MODFLOW can be found here.
MODFLOW takes text files as inputs, executes them, and returns text files and binary files as outputs. It can be run directly through the **command line interface** (a way of interacting with a computer by entering successive text commands from the keyboard, rather than pointing and clicking with a mouse like a graphical interface). GUIs and command-line interfaces are both ways of "talking" to MODFLOW, but GUIs tend to be more human-friendly, since understanding long text files is not as intuitive as visually navigating windows and menus. Groundwater Vistas works by creating input text files based on the options you select in the drop-down menus and tabs (this is called **pre-processing**), and saving them in the folder GWV7/Models. It then feeds those files into MODFLOW, MODFLOW generates output text files, and Groundwater Vistas interprets the output files (this is called **post-processing**) visually for you. You can open, view, and edit these input and output files using a text editor like Notepad (sometimes this requires changing the file extension to '.txt' first).

But, Groundwater Vistas is only one of many MODFLOW interfaces. Other examples include MODFLOW-GUI, ModelMuse, Visual MODFLOW, and FloPy. A more comprehensive list of pre-processors, post-processors, and GUIs can be found here.

We will now be introducing a different MODFLOW interface: FloPy.
FloPy is an editable **package** (a bundle of scripts, functions, and definitions) written in the programming language Python. You will need to install Python on your computer, and then install a series of several additional packages that will enable you to "talk" to MODFLOW using Python. There are two versions of Python: Python 2 and Python 3. We will be using Python 3. In order to keep all these packages and versions straight, it's helpful to use a **package manager** - in this case, we will use the package manager Anaconda.
Finally, because Python is an **interpreted** language (not a compiled language like FORTRAN), to use it you write text files (scripts) and then run them from the command line, similar to running MODFLOW without a GUI, except that there is no .exe file and you can edit your script as you go. It can be clunky to constantly switch back and forth between editing the text files and running them, so a number of **development environments** are available to make it easier to modify and run Python scripts quickly - a development environment combines a text editor (used to create the input files) with an interpreter (used to actually tell Python to run/interpret the file). A more detailed explanation of development environments and a list of common Python-specific ones can be found here. We will use primarily either Jupyter or Spyder. If you are coming from MATLAB (yet another programming language, but a proprietary, i.e. not free, one that can only be used in the MATLAB development environment, unlike Python, which can be used without any development environment at all, or with any number of custom development environments), Spyder looks very similar to the development environment of MATLAB.

You can build exactly the same models you made in Groundwater Vistas, using FloPy. Initially, FloPy is less intuitive and visual than Groundwater Vistas, but for more complex models, it is far more flexible and is easier to modify & run. It is also free and open-source, unlike Groundwater Vistas, which is proprietary.

This guide will walk you through installing Python using the Anaconda package manager, installing the Spyder and Jupyter development environments, and adding several packages to Python: numpy, matplotlib, and flopy.

Note: Python will work on Macs, but because the FloPy package uses Python to talk to MODFLOW, which doesn't work on Macs, you will still need a PC to run FloPy.

Installation guide:

1. Download Anaconda here. Choose the Windows Python 3.6 version. Run the .exe file and follow the instructions.
For troubleshooting these steps in general, refer to the conda documentation here.

*Note: the command-line instructions in this documentation should be run from Anaconda Prompt (NOT the regular command prompt window), which you can find using the Windows search bar on your computer.

*Note: these instructions assume you don't already have a version of Python somewhere on your computer. If you do, it will be easiest if you get rid of it first. If you need to keep multiple versions of Python, you'll need to set up multiple environments to keep them separate (see here and step 7 below).

2. Check that the install was correctly performed: open Anaconda Prompt, and type: `conda --version`
You should get the version number.

3. Find out what packages are already installed. In Anaconda Prompt, type: `conda list`
You will get a list of packages installed. Look for numpy, matplotlib, jupyter, and flopy. If they are all present, you're good. If not, you will need to install them.

4. Install the extra packages. To find packages, go to anaconda.org and use the search bar at the top of the page to find the package you want - in this case, start with numpy. If there are multiple search results with the same name, click on the one with the most downloads. Copy the line of text on the package's page under the install section into Anaconda Prompt. In this case, it will look like this:

```
conda install -c conda-forge numpy
```

*Note: the general format for installs is: `conda install -c channel_name package_name`
Do this again for matplotlib and flopy.

To install jupyter, just type:

```
conda install jupyter
```

5. Check that you can open the Spyder development environment. Spyder should have been installed with Anaconda, so you can search for it in your computer's Finder window ("Search this PC") or using the search bar at the lower left corner of your screen. If you can't find it, follow the install instructions here.

6. Check that you can open Jupyter. In Anaconda Prompt, type:

```
jupyter notebook
```

Jupyter should open in a browser window. The Anaconda Prompt window is now dedicated to this Jupyter session, and needs to remain open the entire time you are using Jupyter. In order to perform other tasks from the command line, open a separate Anaconda Prompt window (you can have multiple prompt windows open at the same time).

7. Set up a Python environment for this project: open Anaconda Prompt, then see what environments exist:

```
conda info --env
```

The active environment will be marked with an asterisk.

To create a new environment called HWR582, type:

```
conda create --name HWR582
```

To activate the new environment, type:

```
activate HWR582
```

To check what packages are available in the new environment, type:

```
conda list
```

If numpy, matplotlib, jupyter, and flopy are not there, install them by following the instructions in step 4 above, in this environment.

8. Now that your HWR582 environment is active, open a new Jupyter window, and in the file directory it displays, navigate to the folder where you plan to store your Python scripts. A copy of your mf2005.exe file will need to be in the same folder as your Python scripts if you plan to use FloPy. You can create your first Jupyter notebook by clicking the “New” tab on the right side of the page and selecting “Python 3” under “Notebook”.

9. To update packages in an environment, activate that environment, then type:

```
conda update package_name
```

Useful links:

MODFLOW documentation:
https://water.usgs.gov/nrp/gwsoftware/modflow2000/MFDOC/index.html?dis.htm

FloPy documentation:
https://modflowpy.github.io/flopydoc/code.html

Python 3 documentation:
https://docs.python.org/3/

NumPy module documentation:
https://docs.scipy.org/doc/numpy-dev/user/index.html

Python cheatsheet for MATLAB users:
https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html