**Initial setup: these instructions are for getting a version of the linux operating system working on your Windows machine. This allows you to use the same linux commands (e.g. for navigation and making directories) as you would on a remote linux machine. These also show you how to install the Anaconda distribution of python in that linux environment, and make plots appear.**

**Thanks to Amanda Manaster, Alexander Islas and other students from the 2020 Effective computing class for putting this together.**

**NOTE: You can also have a separate installation of Anaconda python that works directly in your Windows operating system. Just get the most recent graphical installer for Windows from here:** [**https://www.anaconda.com/**](https://www.anaconda.com/) **and follow their directions.**

1. **Activate WSL**
   1. Navigate to: Control Panel>Programs>Programs and Features>Turn Windows features on and off
   2. Check: Windows Subsystem for Linux
   3. Click OK and let system reboot
2. **Getting Linux**
   1. Go to Microsoft Store and search for Ubuntu
   2. Download Ubuntu for Windows
   3. Open Ubuntu and create a user account and password
      1. Note: from here, using bash in the windows Command Prompt gets you to the same Linux style prompt.
   4. In Ubuntu, enter: sudo apt-get update && sudo apt-get upgrade
3. **Install Python in Ubuntu:**
   1. Find the latest version of Anaconda for Linux at: <https://repo.anaconda.com/archive/> in any browser.
      1. Find the version with “Linux-x86\_64.sh” at the end.
      2. Find the version number as “Anaconda3-{version\_number}-Linux”
      3. Use that version number in step 3.b and 3.c without the curly brackets.
   2. Enter: wget [https://repo.anaconda.com/archive/Anaconda3-{version\_number}-Linux-x86\_64.sh](https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh)
   3. Enter: bash Anaconda3-{version\_number}-Linux-x86\_64.sh , and wait.
   4. When prompted, select ‘yes’ to add to PATH, selection defaults to ‘no’
   5. Enter: conda list, if this shows an error about ‘command not found’, conda was not added to path, continue, else skip to step 4.
   6. Enter: nano ~/.bashrc
   7. At the bottom of the file add: export PATH=~/anaconda3/bin:$PATH , save and close file. Using Ctrl + o and then ctrl + x
   8. Enter: source ~/.bashrc
   9. Enter: conda list, now you should end up with a list of available modules
4. **Update Python:**
   1. Enter: conda update --all
   2. Pip install
      1. Try: pip install netCDF4 seawater gsw
      2. For the modules that give an error (often gsw), continue to 4.c
   3. Conda install
      1. conda install [package\_name], without brackets.
      2. For the modules that give an error, continue to 4.d
   4. Conda install through channels
      1. conda install -c conda-forge [package\_name], without brackets.
5. **Verify that Python works in Ubuntu:**
   1. Enter: ipython --pylab
   2. Enter:
      * + - import matplotlib.pyplot as plt
          - import numpy as np
          - x = np.linspace(0,10,1000)
          - y = np.sin(x\*\*2)
          - plt.plot(x,y)
   3. If the plot doesn’t show, continue to step 6 or within the same ipython enter:

* plt.savefig(“[filepath]/[filename].png”)

1. **To display graphics from Ubuntu:**
   1. Download and install Xming X Server for Windows
      1. <https://sourceforge.net/projects/xming/>
      2. Keep the default settings
   2. Launch Xming
      1. It should show up in your system tray (the X icon being visible means it’s running)

A screen shot of a computer

Description automatically generated

* 1. Tell Ubuntu where to display the graphics
     1. export DISPLAY=localhost:0.0
  2. Add the previous command to ~/.bashrc to make it permanent
     1. nano ~/.bashrc
     2. Add export DISPLAY=localhost:0.0 to the end of the file and save it (Ctrl+O to write to the file, hit ‘enter’, then Ctrl-X to exit)
  3. Run source ~/.bashrc
  4. Try Step 5 again
  5. If step 5 still doesn’t show the plot, check your backend in ipython
     1. Enter: import matplotlib and then matplotlib.get\_backend()
     2. If the backend is agg, go to the next step.
  6. Install libraries to get the correct backend (for Ubuntu, we want our backend to be TkAgg).
     1. sudo apt-get install tcl-dev tk-dev python-tk python3-tk

1. **Try Step 5 again, but enter ipython --pylab=tk to set the backend.**