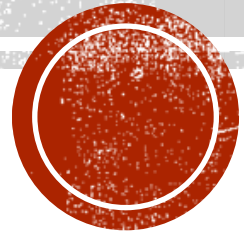
OPENMC INSTALLATION



*note: some of the characters are rendering wrong in PowerPoint (ie. em dashes instead of en dashes, '~' character not working, etc.) so beware of that when copying from this presentation

IF USING A MAC

- SKIP STEPS 1-4
  - You will still want to have python installed, and I recommend making an env for openmc
- Go to this link and follow the install steps
  - https://docs.openmc.org/en/stable/quickinstall.html#building-source-on-linux-or-macos
- NOTE: You cannot complete that install using the default apple compiler (Clang). You will need to install the GCC compiler and use that to run make and cmake. However, even after specifying GCC, sometimes mac redirects a GCC call to Clang for whatever reason. Use Chat and figure out how to make it not do that. (also feel free to email willm412@byu.edu if you are really stuck)
- N

1. DOWNLOAD WSL (WINDOWS 11)

Find the settings “Turn Windows features on or off”

Make Sure “Windows Subsystem for Linux” is enabled

Open windows Powershell

wsl --install

→ should by default install wsl 2 on your computer with ubuntu (default)

2. DOWNLOAD MINICONDA IN WSL

HTTPS://WWW.ANACONDA.COM/DOCS/GETTING-STARTED/MINICONDA/INSTALL#LINUX-2

mkdir -p ~/miniconda3

→ Makes directory in the home dir called miniconda3

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda3/miniconda.sh

→ Goes to the minconda website and downloads the installer (bash file)

bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3

→ Runs the bash script to install conda

rm ~/miniconda3/miniconda.sh

→ Removes the used install bash script

source ~/miniconda3/bin/activate

→ makes it so that your conda is active

conda init

→ Make it active in all instances (may need to include the flag –all)

3. CONDA-FORGE

conda config --add channels conda-forge

conda config --set channel_priority strict

→ changes conflicting package version resolution to favor conda-forge packMakes it so that you can download packages from non-default locations (conda-forge, and ages)

4. CREATE OPENMC ENVIRONMENT WITH OPENMC PACKAGE INSTALLED IN IT

conda create --name openmc-env openmc

→ Creates virtual env called openmc-env that has the package openmc installed in it

conda activate openmc-env

→ Switches you to the openmc-env environment

This will take a second because it will create the env, and then install openmc package and all dependent packages

5. DOWNLOAD FUNDAMENTAL NUCLEAR DATA

mkdir -p ~/openmc/fundamental_nuclear_data/depletion_chains

mkdir –p ~/openmc/fundamental_nuclear_data/x_section_data

→ make directories to put the data in

https://openmc.org/data/

→ download official release of ENDF VII.1 and the Fast spectrum depletion chains for VII.1

→ put the x-sections in x_section_data directory, and depletion chains in depletion_chains directory

cd ~/openmc/fundamental_nuclear_data/x_section_data

→ moves you to the directory where your data is

tar –xf endfb71.tar.xz

→ unpacks the data from the tar.xz files

. . . CONTINUED

rm endfb71.tar.xz

→ delete the used compressed file

Add the x-sections and chains to your .bashrc file (set as environment variables)

nano ~/.bashrc

→ you're now editing the .bashrc file located in the home directory (using the nano editor)

Scroll to the very bottom and add these lines:

export OPENMC_CROSS_SECTIONS="~/openmc/fundamental_nuclear_data/x_section_data/endfb-vii.1-hdf5/cross_sections.xml"

export OPENMC_CHAIN_FILE="~/openmc/fundamental_nuclear_data/depletion_chains/chain_endfb71_sfr.xml"

→ These make environment variables so that when you run openmc it can find the data

Exit the nano editor:

[ctrl-x] [y] [enter]

... CONTINUED

source ~/.bashrc

→ rerun the .bashrc file to set the environment variables

echo $OPENMC_CROSS_SECTIONS

echo $OPENMC_CHAIN_FILE

→ check to see if the variables were set correctly (it should display the path that we set in the .bashrc file)

cd ~/openmc

→ navigate to openmc dir (if not already there)

mkdir test

→ create a directory called test (a surprise tool for later)

6. OPTIONAL (I RECOMMEND) : INSTALL JUPYTER NOTEBOOK

Ensure you are in openmc-env (not base)

conda install jupyter ipykernel

→ Install jupyter inside your Conda environment

Ensure you are in openmc-env (not base)

cd ~/openmc

→ navigate to the openmc directory

jupyter-notebook

→ run jupyter

Open jupyter in a browser

If you have a jupyter instance running on your windows side it sometimes gets mixed up. If needed add the flag --port 9999 (or some other port so it doesn't conflict)

7. TEST THAT IT ALL WORKS

Open the test director

Create a notebook called “test1.ipynb”

Write and run these lines (if you get the same result, you have succeeded)

```
import openmc
```

```
uo2 = openmc.Material(1, "uo2")
print(uo2)
```

```
Material
        ID             =       1
        Name           =       uo2
        Temperature    =       None
        Density        =       None [sum]
        Volume         =       None [cm^3]
        Depletable     =       False
        S(a,b) Tables
        Nuclides
```

A LAST COUPLE NOTES:

- You don't need a notebook to run openmc. I prefer to code it as a .py file and to run periodically to debug, but notebooks are really helpful for learning how it all works
- The openmc documentation website is a great resource, refer to that often