pyFWI installation Manual

- Open Anaconda Prompt
- conda create -n pyfwi_env python=3.9 -y
- conda activate pyfwi_env
- git clone https://github.com/AmirMardan/PyFWI
- cd pyFWI
- pip install -e .
- conda install jupyter notebook
- Restart Anaconda Prompt
- conda activate pyfwi_env
- cd pyFWI
- pip install -e .
- jupyter notebook (in the pyfwi_env and in the pyFWI directory)/(still we didn't get pyFWI kernel)
- pip install ipykernel
- python -m ipykernel install --user --name=pyfwi-env --display-name "Python (pyFWI)" -------(makes pyFWI kernel)
- again restart jupyter notebook after the installation of pyFWI kernel
- now when we open jupyter notebook select pyFWI kernel instead of python 3 kernel
- now open Anaconda prompt again
- pip list
- findstr pyFWI (not necessary) (if it doesn't show up)
- pip install -e .
- conda install -c conda-forge pyopencl (or, pip install pyopencl)
- conda install -c conda-forge segyio (or, pip install segyio)
- pip install charset_normalizer (or, pip install chardet)
- Open Anaconda prompt under pyfwi_env in the pyFWI directory
- pip uninstall -y numpy
- conda remove -y numpy
- conda install -c conda-forge numpy=1.24.4
- python -c "from numpy.lib.function_base import kaiser; print('kaiser OK ✅')"
- Open Anaconda prompt under pyfwi_env in the pyFWI directory and select the pyFWI kernel and open new notebook and run codes using pyFWI

Simple FWI example

- There will always show (ImportError: cannot import name 'MemoizeJac' from 'scipy.optimize.optimize') (this is because newer versions of scipy has no 'MemoizeJac')
- So we have to manually fix this problem (one approach is to downgrade the scipy but it will jeopardize the status of other libraries and then we have to downgrade the python version also which will affect our module so never downgrade scipy or python)
- so now to manually fix this import error open jupyter notebook
- Home--src--pyFWI----fwi.py (open)
- Remove or comment the line (from scipy.optimize.optimize import MemoizeJac)
- add the line
- class MemoizeJac:
- def __init__(self, func):
- self.func = func

- self.jac = None
- self.x = None
- self.cost = None
-
- def __call__(self, x, *args):
- # Must return a scalar cost
- if self.jac is None or not np.allclose(x, self.x):
- self.cost, self.jac = self.func(x, *args)
- self.x = np.copy(x)
- return self.cost
-
- @property
- def derivative(self):
- return self
- ctrl+s to save the change
- now restart the pyFWI kernel and run all the cells

py Torch installation

- If we don’t install this then it will show (ModuleNotFoundError: No module named 'torch')
- Open pyfwi_env in anaconda prompt and cd pyFWI
- conda install pytorch torchvision torchaudio cpuonly -c pytorch
- After installation restart pyFWI kernel and run the pytorch code