Interactive Python Exercise
Modifying the Python Scripts

SU2 Workshop Feb 3rd 2017
Heather Kline
Python Scripts

- Source code location: SU2/SU2_PY/
- Installed location: SU2/bin/
- Dependencies: SU2/bin/SU2/ (source: SU2/SU2_PY/SU2/)

To run a local version:
$ ./python_script.py

To run version installed in the bin/ directory:
$ python_script.py
#!/usr/bin/env python

# file Compute_Mpolar.py
# brief Python script for performing polar sweep of Mach number.
# author H. Kline (E. Arad compute_polar script)
# version 5.0.0 "Raven"

# SU2 Lead Developers: Dr. Francisco Palacios (Francisco.D.Palacios@boeing.com).
# Dr. Thomas D. Economou (economou@stanford.edu).
# Prof. Piero Colonna's group at Delft University of Technology.
# Prof. Nicolas R. Gauger's group at Kaiserslautern University of Technology.
# Prof. Alberto Guardone's group at Polytechnic University of Milan.
# Prof. Rafael Palacios' group at Imperial College London.

# Copyright (C) 2012-2017 SU2, the open-source CFD code.

# SU2 is free software; you can redistribute it and/or
# modify it under the terms of the GNU Lesser General Public
# License as published by the Free Software Foundation; either
# version 2.1 of the License, or (at your option) any later version.

# SU2 is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# Lesser General Public License for more details.

# You should have received a copy of the GNU Lesser General Public
# License along with SU2. If not, see <http://www.gnu.org/licenses/>.

# imports
import numpy as np
import matplotlib.pyplot as plt
from optparse import OptionParser
import os, sys, shutil, copy, os.path
sys.path.append(os.environ['SU2_RUN'])
import SU2

# def main():
if __name__ == "__main__":
    main()
wget su2.stanford.edu/documents/WorkshopFeb2017/compute_Mpolar.py

Starts python environment

Import python packages and functions defined in other files

'import SU2' loads numerous functions defined in SU2_PY/SU2/

Execute 'main()' when this script is called from terminal
wget su2.stanford.edu/documents/WorkshopFeb2017/compute_Mpolar.py

```python
#!/usr/bin/env python

# file Compute_Mpolar.py
# brief Python script for performing polar sweep of Mach number.
# author H. Kline (E. Arad compute_polar script)
# version 5.0.0 "Raven"

# SU2 Lead Developers: Dr. Francisco Palacios (Francisco.D.Palacios@boeing.com).
# Dr. Thomas D. Economon (economon@stanford.edu).

# Copyright (C) 2012-2017 SU2, the open-source CFD code.

# SU2 is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

# SU2 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

# You should have received a copy of the GNU Lesser General Public License along with SU2. If not, see <http://www.gnu.org/licenses/>.

# imports
import numpy as np
import matplotlib.pyplot as plt
from optparse import OptionParser
import os, sys, shutil, copy, os.path
sys.path.append(os.environ['SU2_RUN'])
import SU2

def main():
    if __name__ == "__main__":
        main()
```
```python
import os
import numpy as np
import matplotlib.pyplot as plt
from SU2 import SU2

def main():
    parser = OptionParser()
    parser.add_option("-f", "--file", dest="filename",
                    help="read config from FILE", metavar="FILE")
    parser.add_option("-p", "--partitions", default=2,
                    help="number of PARTITIONS", metavar="PARTITIONS")
    parser.add_option("-i", "--iterations", dest="iterations", default=99999,
                    help="number of ITERATIONS", metavar="ITERATIONS")
    (options, args) = parser.parse_args()
    options.partitions = int(options.partitions)
    options.iterations = int(options.iterations)

    # load config, start state
    config = SU2.io.Config(options.filename)
    state = SU2.io.State()

    # find solution files if they exist
    state.find_files(config)

    # prepare config
    config.NUMBER_PART = options.partitions
    config.MAX_ITER = options.iterations

    # Initialize results arrays
    nMach = 5
    MachList = np.linspace(0.5, 0.6, nMach)
    LiftList = []
    DragList = []

    # Output file
    outFile = 'Polar M' + str(MachList[0]) + '.dat'
    f = open(outFile, 'w')
    f.write('% Mach, C_L, C_D, C_D0n')

    # Iterate on Mach number
    for MachNumber in MachList:
        # local config and state
        config = copy.deepcopy(config)
        state = copy.deepcopy(state)

        # set config options
        config.MACH_NUMBER = MachNumber
        cosName = 'DIRECT_M' + str(MachNumber)

        # run su2
        drag = SU2.eval.funct('DRAG', config, state)
        lift = SU2.eval.funct('LIFT', config, state)
        LiftList.append(lift)
        DragList.append(drag)

        output = str(MachNumber) + '. ' + str(lift) + ' ' + str(drag) + ' \n'
        f.write(output)

    # Store result in a subdirectory
    if os.path.isdir(cosName): 
        os.system('rm -R ' + cosName)
        command = 'mkdir ' + cosName
        os.system(command)

    # Close open file
    f.close()

# plotting
plt.figure()
plt.plot(MachList, LiftList)
plt.xlabel('Mach')
plt.ylabel('Lift Coefficient')
plt.show()

if __name__ == '__main__':
    main()
```

Define and parse command line options (-f filename, etc)
Initialize python objects that run SU2, modify options.

To modify config options from the python script:

```python
config.EXT_ITER = options.iterations
```
Set up range of Mach numbers, open the output file, and start a for loop.
Copy the config and state objects using `copy.deepcopy` and set the Mach number.
Evaluate lift and drag using `SU2.eval.func('DRAG', konfig, ztate)`

Note that only the first `eval` runs `SU2_CFD`, after that the output values are stored in the state object.
Store results, close output file, and plot results
Run, and modify compute_Mpolar.py

- What do we need to change to plot an additional output (ie, Cl/Cd, Cmz)?
Mini-project: N minutes to modify a python script

- Goal: execute a problem that requires several CFD simulations (10-20).
- Use the Quickstart input files
- Suggested mini-projects:
  - Sweep through angle of attack and output the lift coefficients.
  - Estimate the standard deviation of lift with respect to an input distribution of Mach number.
- Resources:
  Plotting with python: [http://matplotlib.org/faq/howto_faq.html](http://matplotlib.org/faq/howto_faq.html)
  Numpy (arrays, random numbers, etc): [https://docs.scipy.org/doc/numpy/reference/index.html](https://docs.scipy.org/doc/numpy/reference/index.html)
  File I/O with Python: [https://docs.python.org/2/tutorial/inputoutput.html#reading-and-writing-files](https://docs.python.org/2/tutorial/inputoutput.html#reading-and-writing-files)
  Output function names: SU2_PY/SU2/io/tools.py → get_headerMap()