PySimpleMAC Documentation

Release 1.0

Jon Komperda

December 12, 2012

CONTENTS

CHAPTER

ONE

INTRODUCTION

PySimpleMAC is an Open Source Computational Fluid Dynamics Solver. It employs Harlow's Marker and Cell discretization scheme to solve the non-dimensional form of the incompressible Navier-Stokes equations.

The original version of the code, known as SimpleMAC was written for a graduate Supercomputing course at University of Illinois at Chicago. It has since been expanded and ported to Python for this project, which is part of a graduate Python class.

PySimpleMAC includes a GUI which produces a real time solution to the Lid Driven Cavity problem as a benchmark.

CHAPTER

TWO

GETTING PYSIMPLEMAC

The source code may be downloaded as a tarball from the main page, however users are strongly encouraged to use Git and contribute to the project. To get the code using Git, you may simply enter the following in the terminal:

```
$ cd you_source_directory
$ git clone https://github.com/jonkomperda/PySimpleMAC.git
```

A great reference for using git may be found here

CHAPTER

THREE

USING PYSIMPLEMAC

In order to run the example included with PySimpleMAC you must have the following software/packages:

- Compilers
  - gfortran
  - OpenMPI
- Python Packages
  - Numpy
  - Matplotlib
  - MPI4PY
  - PyVTK

Once all the dependancies are installed, simply navigate to the PySimpleMAC directory and execute the 'Makefile' and run the example script:

```
$ cd PySimpleMAC
$ make
$ ./runGUIExample.py
```

CHAPTER

FOUR

PYSIMPLEMAC MODULES

This section describes the Python modules included in the project.

Note: The Fortran source is documented separately in the developers guide

4.1 includes module

This folder includes all wrapped source files.

4.2 macMethod module

class `includes.macMethod.`**`MACSolver`**(*fluid*, *domain*, *steps=5000*)
A generalized Marker and Cell method solver for the incompressible Navier-Stokes equations.

`reset`()
Resets the solver to the initial condition state.

`run`(*pinterval=1000*)
Runs the method as a 'black box' for a predetermined number of steps, set upon initialization of the solver. Prints using PyVTK at predetermined intervals.

`step`()
Takes a single computational step. Data may be accessed from the fluid class.

`visitPrint`(*k*)
Uses PyVTK to write out data in a VisIt Visualization Tool capable format.

class `includes.macMethod.`**`domain`**(*r=0.1*)
Defines an object that behaves as a computatational domain.

class `includes.macMethod.`**`fluid`**(*domain*, *re=100*)
Defines a fluid object. Contains fluid parameters.

`p`()
Accesses the Fortran persistent data and returns the Pressure in the form of a Numpy array.

`setRe`(*re*)
Sets the Reynolds number in real time for the Fortran program.

`u`()
Accesses the Fortran persistent data and returns U-Velocity in the form of a Numpy array.

v()
Accesses the Fortran persistent data and returns V-Velocity in the form of a Numpy array.

w()
Calls a Fortran function to calculate Vorticity and returns it's value in the form of a Numpy array.

4.3 sliderWindow module

class includes.sliderWindow.**App**(*master*)
The main GUI window which controls the plot window. Should be called from a separate MPI process as the plot window.

quitIt()
Sends a quit message to both threads. Proper way to exit the program and ensure both threads recieve an exit message.

reset()
Sends reset message to computational thread. Restores initial conditions to the solver.

saveit()
Uses a communicator to have computational thread save a jpeg

startRun()
Tells the compute thread to begin computation. Should be called by a button.

stopRun()
Tells the compute thread to end computation. Should be called by a button.

updateRe(*master*)
Function which communicates with the computational thread to change the Reynolds number in the solver. Recieves value from slider and uses non-blocking send to main thread.

viewP()
Sets plot to display Pressure

viewU()
Sets plot to display U-Velocity

viewV()
Sets plot to display V-Velocity

viewW()
Sets plot to display Vorticity

CHAPTER

FIVE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

INDEX