11.4 Activity 1

In Activity 1, you set up, compiled and ran a 1D (1-dimensional) test problem with the PLUTO MPI code. In this activity you will test the Rayleigh-Taylor instability in 2D. The initial condition consists of an interface separating two fluids with different densities in hydrostatic equilibrium. As the fluids fall into each other an instability ensues.

At this point, it may be useful to browse the documentation for the PLUTO Test Problems:
http://plutocode.ph.unito.it/Doxygen/Test_Problems/files.html
http://plutocode.ph.unito.it/Doxygen/Test_Problems/_m_h_d_2_rayleigh___taylor_2init_8c.html

```
username@h2ologin2: cd $PLUTO_DIR/Test_Problems/MHD/Rayleigh_Taylor
username@h2ologin2: cp pluto_01.ini pluto.ini
username@h2ologin2: cp definitions_01.h definitions.h
username@h2ologin2: vi pluto.ini
```

```
[Static Grid Output]

uservar    0
dbl       -1.0  -1   single_file
dbl.h5    -1.0  -1
flt       -1.0  -1   single_file
flt.h5    -1.0  -1
vtk        1.0  -1   single_file
tab       -1.0  -1
ppm       -1.0  -1
png       -1.0  -1
log        10
analysis  -1.0  -1
```

```
username@h2ologin2: python $PLUTO_DIR/setup.py
```

Select `Linux.mpicc.defs` and `Auto-update`. Make pluto, remove unneeded object files, cp your old qsub file, and run job.

```
username@h2ologin2: make
username@h2ologin2: make clean
username@h2ologin2: cp $PLUTO_DIR/Test_Problems/HD/Sod/pluto.pbs .
username@h2ologin2: qsub pluto.pbs
username@h2ologin2: qstat <job>.bw
```

When your job completes, MPI should produce a log file for each processor (32) and 15 vtk files obtained at 1-second intervals during the simulation. We will visualize this output next time.

For now, check one of the .log files, and one of the pluto.o<job> files.

1. How long did the simulation take?
2. What was the resolution of the grid X-Y grid?
3. The simulation stopped after tstop=15.0 (these are code time units). How long (walltime) would the computation have taken for tstop=20?
4. Imagine you were to double the number of grid points in X and Y. How long (walltime) would the computation have taken for tstop=20?

We will now modify our problem, and rerun the simulation. Before we do that. Let's save our results from before in a directory called run1.

```
username@h2ologin2: mkdir run1
username@h2ologin2: mv *.out *.log *.vtk pluto.o* pluto.e* run1
username@h2ologin2: cp pluto.ini definitions.h run1
```

Now edit the pluto.ini file, and repeat the prior steps. You will only need to modify the Grid block and the Time block:

```
username@h2ologin2: vi pluto.ini
```

```
[Grid]

X1-grid  1    -0.5    512    u     0.5
X2-grid  1    -1.0    1024   u     1.0
X3-grid  1    -0.5     1     u     0.5

[Time]

CFL              0.8
CFL_max_var      1.1
tstop            20.0
first_dt         1.e-3
```

```
username@h2ologin2: python $PLUTO_DIR/setup.py
```

Select `Auto-update.`

```
username@h2ologin2: make
username@h2ologin2: make clean
```

How long (walltime) do you think this computation will take? Make sure that your walltime parameter in `pluto.pbs` is longer than you estimate.

```
username@h2ologin2: qsub pluto.pbs
username@h2ologin2: qstat <job>.bw
```

Now, check one of the .log files, and one of the pluto.o<job> files. How long did the simulation take? How good was your estimate?

Optional Exercise

For the curious, you can try different methods for solving this problem. See: http://plutocode.ph.unito.it/Doxygen/Test_Problems/_m_h_d_2_rayleigh___taylor_2init_8c.html

For example, try initialization 03:

```
username@h2ologin2: cp pluto_03.ini pluto.ini
username@h2ologin2: cp definitions.h
```

Now edit the pluto.ini file, and repeat the prior steps. You will only need to modify the Grid block and the Time block:

```
username@h2ologin2: vi pluto.ini
[Grid]

X1-grid  1    -0.5    512    u     0.5
X2-grid  1    -1.0   1024    u     1.0
X3-grid  1    -0.5     1     u     0.5

[Time]

CFL              0.45
CFL_max_var      1.1
tstop            20.0
first_dt         1.e-3

username@h2ologin2: python $PLUTO_DIR/setup.py
username@h2ologin2: make
username@h2ologin2: make clean
```

Make sure that your walltime parameter in `pluto.pbs` is at least 30 minutes. Try using 2 nodes, 64 cores, 1 task per core.

```
username@h2ologin2: vi pluto.pbs
#!/bin/bash
#PBS -l nodes=2:ppn=32:xe
#PBS -l walltime=00:30:00
#PBS -N pluto

cd $PBS_O_WORKDIR
```

```
aprun -n 64 ./pluto
username@h2ologin2: qsub pluto.pbs
username@h2ologin2: qstat <job>.bw
```