Hands-on session using RAMSES
Exercises

Evangelia Ntormousi
`entorm@ia.forth.gr`

Before starting

Check that you have installed and compiled RAMSES and the post-processing software!

Exercise 1: 1d Advection test

Here we will solve the scalar advection equation in one dimension:

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = 0$$

using different CFL criteria and different Riemann solvers.

1. Compile RAMSES with `NDIM=1` and `SOLVER=hydro`.
2. Create a new directory on your computer in which you will run the code, say `mkdir ./1dadvectiontest`
3. Copy your new executable into the run directory:
   ```
   cp /yourpathto/ramses/bin/ramses1d ./1dadvectiontest
   ```
4. Then also copy a namelist into the directory:
   ```
   cp /yourpathto/ramses/namelist/advect1d.nml ./1dadvectiontest
   ```
5. Go into the run directory and run the code:
   ```
   ./ramses1d advect1d.nml > advect1d.log
   ```
6. Now the code's output is stored in the `advect1d.log` file. You can use the provided python script: `plot_1d_data.py` to plot the output at different times.
7. Now repeat the process, changing the Courant condition in the namelist. Try `courant_factor` equal to 0.1, 0.3, 0.8. Discuss.
8. Once you decide on a good Courant condition, change the Riemann solver. Try `riemann` equal to 'exact', 'hll', 'llf'. Discuss.

Exercise 2: 1d Sedov explosion

Consider the release of a large amount of energy, E in a small volume, V. This situation is applicable, for example, to a supernova explosion.

1. Similarly to Exercise 1, compile RAMSES with `NDIM=1` and `SOLVER=hydro` (If the code is already compiled for one dimension, no need to repeat this step).
2. As before, create a new directory on your computer to run the code, say `mkdir ./sedov1dtest`
3. Copy the executable into the run directory.
4. Then also copy a namelist into the directory:
   `cp /yourpathto/ramses/namelist/sedov1d.nml ./sedov1dtest`
5. Go into the run directory and run the code:
   `./ramses1d sedov1d.nml > sedov1d.log`
6. Now like in Exercise 1, you can use the provided script to plot the output at different times. Feel free to modify it, and maybe overplot the analytical solution.

Exercise 3: 3d Kelvin-Helmholtz instability

The Kelvin-Helmholtz instability is a classic example of an unstable flow. The equilibrium configuration is that of two superimposed fluids moving in opposite directions. Perturbations either of the interface between the two flows or of the vertical velocity of the two fluids, are unstable for all wavenumbers (see Chandrasekhar, "Hydrodynamic and Hydromagnetic stability").

This exercise is a bit more complicated, because we will use three dimensions, and it will require you to modify one of the code's routines. For that reason, we will create what is called a `PATCH` for the code.

A `PATCH` is a directory where you replace some of the code's routines with your own. When compiling, RAMSES will look for the routines there first, and omit the corresponding files its native directories. Here, our "patch" will only consist of the initial condition routine, `condinit.f90`.

Creating a `PATCH` directory

To keep things simple, our `PATCH` directory will be the same as the run directory. Go ahead and create a new directory:

```
mkdir kh_instability
```

and copy `condinit.f90` in it:

```
cp /yourpathto/ramses/hydro/condinit.f90 kh_instability
```

Then you can change into the patch directory and start modifying `condinit.f90`.

Modifying `condinit.f90`

Open the file `condinit.f90` you just copied into the run directory. You will find the following comment:

```
!  Add here, if you wish, some user-defined initial conditions
```

after which you can add a loop over the `nn` cells. In this loop, you need to set values for the primitive variable vector, `q(1:nn,nvar)`, where `nvar` is the variable number. Here, 1 is the density, 2-4 the velocities, and 5 the pressure. Your setup will be two flows, separated in the middle of the box, one moving in the positive x-direction and the other to the negative x-direction with velocities $u_x = \pm 1$. The flows can have the same, or different densities. In this routine, the positions of the grid cell centers are contained in `x(1:nn,1:3)`, in units of `[0,boxlen]`, where 1-3 are the x,y,and z directions.

In order for the instability to develop, you need to set a perturbation, either in the y-velocity u_y, or at the interface y itself. Choose a sinusoidal form for your perturbation, $\delta u_y = A \sin(k_x x)$, or $\delta y = A \sin(k_x x)$, with an amplitude $A < 0.1$ and a wavenumber $k_x = 4$.

Compiling and running the code

1. For this 3D run we will need to `make clean`, change `NDIM=3` and `PATCH=/path/to/our/patch/directory` in the `Makefile`, and type `make` again. Now we have an executable called `ramses3d`.

2. Now we need a namelist for our run. An example namelist (kh.nml) is given below. Create that into the run directory.

3. Now, like before, you can run:
   `./ramses3d kh.nml > kh.log`

4. This time, the output is only stored in ennumerated output directories: `output_00001`, etc. In order to plot the results, you must use one of the recommended plotting programs (pymses, pynbody, VisIt) described in the previous handout (installing requirements).

kh.nml

```
&RUN_PARAMS
nrestart=0
ncontrol=10
nremap=10
nsubcycle=10*1,
/
&AMR_PARAMS
levelmin=4
levelmax=9
ngridmax=100000
nexpand=1
boxlen=1.0
/
&BOUNDARY_PARAMS
nboundary = 2
ibound_min=-1,+1
ibound_max=-1,+1
bound_type= 2, 2
/
&INIT_PARAMS
nregion=2
d_region=2.0,1.0
v_region=+1.0,-1.0
x_center=0.5
/
&OUTPUT_PARAMS
foutput=100
noutput=1
tout=1.0
/
&HYDRO_PARAMS
gamma=1.4
courant_factor=0.8
slope_type=2
scheme='muscl'
/
&REFINE_PARAMS
err_grad_d=0.01
interpol_type=1
/
```