

# Hands-on session using RAMSES

## Installing the necessary requirements

Evangelia Ntormousi  
entorm@ia.forth.gr

### Downloading and compiling RAMSES

The first step is to make sure that you have access to a computer (either your laptop or a remote server) with a fortran compiler for compiling RAMSES and a python installation for post-processing the simulation data.

Then you can download the code from the public RAMSES repository. The easiest way is to run:

```
git clone https://bitbucket.org/rteyssie/ramses
```

from your command line (Linux, Mac). For other systems, you can just download and unpack the code bundle on your computer.

Now you have a directory called `ramses`, with various subdirectories related to the different modules of the code. In order to compile the code, change into the `bin` directory and start editing the `Makefile`. Table 1 shows the options to use for testing that everything works ok, and for a 1-D run. You can leave all the other options unchanged.

| Option   | Value | Comment   |
|----------|-------|---|
| DEBUG    | 1     | This option for testing. Recompile with "0" for faster results. |
| COMPILER | GNU   | Prefer INTEL if you have access to an intel compiler.           |
| NDIM     | 1     | This parameter defines the dimensionality of our problem.       |
| SOLVER   | hydro | You can also test with mhd, but then choose NDIM=3.             |
| PATCH    | blank | We will come back to this parameter for some exercises.         |
| MPI      | 0     | You can switch to "1" if you have access to a server with MPI.  |

Table 1: Compile options for testing the RAMSES installation.

Then type `make`. If everything goes smoothly, then you will have a new executable file in the `bin` directory, called `ramses1d`. Congratulations, you can now move on to installing the next requirement!

If not, you can check out the code documentation, that you will find into the `doc/src` subdirectory, or at the code webpage. If that is of no help, contact me and we will try to figure it out.

## Post-processing packages

For plotting the 1-d results from the simpler problems, a python script will be provided before the lectures. If you want to challenge yourself with a multi-dimensional simulation, you will need one of the following packages to turn a RAMSES output into plotable arrays of the different variables. You can try one or all of them; they are equivalent for the provided exercise set.

### PymSes

If you have a Python 2.7 installation or similar, you can use the `pymes` package. Download the package from the linked webpage above and follow the installation instructions.

There is a version for Python 3.0, which is maintained by Sam Geen, but I have not used it yet. If you do, it will be great to see how it does!

### Pynbody

You can get Pynbody here. The installation is very simple from your python environment: just type `pip install pynbody` from your command line.

### FORTRAN+VisIt

If for some reason you cannot use python, or you like a more interactive environment, you can also try installing VisIt. VisIt works with many different datasets, but it can also be scripted in python.

For RAMSES outputs, you have to first turn the raw simulation output into a `.vtk` file. This can be done by using the `amr2cube.f90` routine, which you will find in your `ramses` repository, in the subdirectory `utils/f90`.

Once you compile `amr2cube.f90`, and say, produce the executable `amr2cube`, you just type

```
./amr2cube inp </path/to/your/output/file> -lmax [max resolution]
-fil 'vtk' -out </path/to/your/output/file.vtk> -typ [number of
variable 1-8]
```

This `.vtk` file can be imported in VisIt and you can make plots interactively.