TIPS & INSTRUCTIONS
"hands-on" session for N-body simulations, Wojciech Hellwing
"2nd Cosmology School, July 2016, Kielce, Poland"

We will be working with GADGET2 – vanilla version. It is a very popular package, also a "easy" one to compile and set-up.

Basic requirements: Linux system. It should be possible to compile the codes and libraries also on MAC, but we will not provide support for MacOS.
The source code package can be downloaded from the official web-page:
http://wwwmpa.mpa-garching.mpg.de/gadget/

For generating the initial condition files, we will use N-GenIC code that use Zeldovitch Approximation. Download it from here:
http://www.h-its.org/tap-software-en/ngenic-code/

For some quick analysis of the simulation outputs, some software packages will be provided before the course.

Dependences:
Both Gadget and NgenIC requires fftw library in version 2.1.5+ (for fast Fourier transforms) and gsl library 1.14+. In addition they need to be compiled with a MPI package (mpi-wrapper on c compiler), since they are parallel codes.
Some Linux systems provide these libraries as a installable packages. For example for Ubuntu and Mint you can try to install the following packages (sudo apt-get install):
fftw2, fftw-dev, libgsl0-dev, mpi-default-bin, mpi-default-dev

Should the packages be not avaiable for your system it is easy to compile all required libraries from sources. We will do this as well during "hands-on" session.
Links to library sources:
GSL: http://www.gnu.org/software/gsl/
fftw2: http://fftw.org/

MPICH:https://www.mpich.org/downloads/
or alternatively:
OpenMPI: https://www.open-mpi.org/

In addition we might use a nice programy by John Helly for visualisation of gadget snapshots, it's called gadgetviewer and the sources can be obtained here:
http://astro.dur.ac.uk/~jch/

Compilation of libraries:
Both MPI and GSL should work with gadget in their default configuration (./configure). However for fftw 2.1.5 we need to assure that the libraries are installed in both versions (double and single precision) and with special prefixes indicating the precision type.
To do this we need to compile and install them twice. Example for any linux system would be (assuming that a directory $HOME/local/ exists, where $HOME is your home):
Single precision:
./configure --prefix=$HOME/local/ --enable-type-prefix –enable-float --enable-mpi --enable-threads
make

make install
Double precision:
./configure --prefix=$HOME/local/ --enable-type-prefix --enable-mpi --enable-threads
make
make install

Since both MPI and GSL should be installed in their defaults, for them it should be enough to configure and compile the sources as:
./configure --prefix=$HOME/local/
make
make install

Compiling NgenIC and GADGET.
You need to edit the apprioprate “Makefile”. Create a new SYSTYPE=”My_sys_name” (and #-out all the others in the Makefile). Now copy one of the existing blocks
ifeq ($(SYSTYPE),”Some name”)
…
endif
and edit it to suite your system. This will most likely depend on setting the paths for libraries and linclude files for GSL, FFTW, MPI if you have compiled and install them locally.
Once properly set-up a normal `make` should work and produce working executable.
The meaning of various flags for the code workflow will be covered during session.