

# AWESOME documentation

## metal enrichment setup 2011ff

Harald Höller, Markus Haider \*

May 21, 2012

**FOR INTERNAL AWESOME USE**

## 1 Initial Conditions

### 1.1 Using Cosmics 1.04 (Standard)

The (constrained) initial conditions generator `Cosmics 1.04` can be downloaded from <http://web.mit.edu/edbert/cosmics-1.04.tar.gz>. The actually used version is to be gotten as git repository `git@github.com:harre/cosmics-initial-conditions.git`.

**Step 1: Make** For making, one has to specify the system and adapt the corresponding Makefile accordingly. In the folder `Make_files` one has to adapt `Make.LINUX` to

```
F77 = ifort
F77FLAGS = -O2 -parallel -par-report1 -openmp
FFT_OBJ = fft3r.o
CC = icc
CFLAGS = -O2 -parallel -par-report1 -openmp
```

and load the intel compiler module `load intel/64/12.1`. The intel compiler is used since for  $512^3$  particle initial conditions there is a problem with memory allocation with `gcc` (can be dealt with flags however). When one uses the version from the git repo, changing the Makefile should not be necessary anyhow.

**Step 2: Linear Extrapolation: linger** In the folder `linger_syn` the program with the same name is to be executed. This program generates a file called `linger.dat` which is then used by the actual IC generator `grafic`.

When one executes the program from github, the message

---

\*Department of Astro- and Particle Physics, University of Innsbruck, [harald.hoeller@uibk.ac.at](mailto:harald.hoeller@uibk.ac.at)

## 2 N-body Simulation

```
>>>> MAKE SURE YOU START THE PROGRAM WITH
>>>> $./linger_syn | tee lingersynIO.out
```

motivates the recording of IO in order to ensure reproducibility of simulations. Ideally one would also indicate some physical parameters in the IO file, such as `lingersynIO_h70.out`. This file is consequently copied from further simulation setup scripts (see later items and sections).

**Note:** There are files already in the repo which can be used, namely `linger_h100n216.dat` and `linger_h70n216.dat` for Hubble constants 70.3 and 100. They were generated using matter transfer functions (choice 0)

```
Enter 1 for full Boltzmann equation for CMB (lmax<=10000, linear k)
or 0 for matter transfer functions only (lmax=100, log k)
```

and the parameters

```
Enter kmin (1/Mpc), kmax (1/Mpc), nk, zend
```

are set to 1E-5, 50, 216, 0. These numbers are chosen partially arbitrary but have proven robust. The upper limit for the wave number certainly will be too small for larger volumes than 100Mpc (rem.: Nyquist frequency). The number of  $k$  to be calculated are chosen in a way that the calculation does not last too long.

The last input is then which kind of IC one wants to generate

Initial conditions cases:

- 1 for isentropic (adiabatic) fluctuations,
- 2 for cdm entropy/isocurvature fluctuations, or
- 3 for baryon entropy/isocurvature fluctuations, or
- 4 for seed/isocurvature fluctuations

```
Enter 1, 2, 3, or 4 now
```

where we chose 2.

**Note:** If one changes the cosmological parameters for `linger_syn` one usually has to delete the files `linger.dat` and `lingerg.dat` first.

### 1.2 Using NGenIC

## 2 N-body Simulation

### 2.1 Compiling Gadget

Gadget has dependencies on MPI, the FFTW and GSL (HDF5 is possible but not required, if you specify so in the Makefile). To Compile on leo 2 we have to load the

## 2 N-body Simulation

modules

```
module load intel/11.1 openmpi/1.3.3 fftw/2.1.5 gsl/1.12
```

However, the modules environment on leo2 changed the environment variables. The easiest way to compile is to change in line 89 the systype to

```
SYSTYPE="leo"
```

and later define

```
ifeq ($(SYSTYPE),"leo")
CC      = mpicc
OPTIMIZE = -Wall
GSL_INCL = -I$(UIBK_GSL_INC)
GSL_LIBS = -L$(UIBK_GSL_LIB)
FFTW_INCL= -I$(UIBK_FFTW_INC)
FFTW_LIBS= -L$(UIBK_FFTW_LIB)
MPICHLIB =
HDF5INCL = -I/opt/hdf5/include
HDF5LIB   = -L/opt/hdf5/lib -lhdf5 -lz -Wl,"-R /opt/hdf5/lib"
endif
```

after that just type “make” and Gadget should compile.

### 2.2 Gadget2 Units

If we make cosmological simulations (which we do) the internal time unit of Gadget is actually the scale factor and not the physical time (at least I think that this is the case). Starting at page 25 of the Gadget user guid <http://www.mpa-garching.mpg.de/galform/gadget/users-guide.pdf> the unit system is described. From there, it seems that the internal units of Gadget are

**UnitVelocity\_in\_cm\_per\_s** is by default set to  $1e5$  cm/s which is 1 km/s. At the moment, I don't understand if this is the proper, physical or peculiar velocity.

**UnitLenght\_in\_cm** is by default set to  $3.085679e21$  cm/h, where the Hubble constant is  $H_0 = 100h$  km s $^{-1}$ Mpc $^{-1}$ . This sets the unitg length to 1.0 kpc/h. If **ComovingIntegrationOn** is set to true (which it should be four our purposes), our UnitLength is in comoving coordinates. To get the physical coordinates, we should multiply the comoving coordinates with the scale factor a.

**UnitMass\_in\_g** is by default set to  $1.989e43$  g/h which is just  $10^{10}M_{\odot}/h$ .

I therefore suppose that if we would like to simulate a “physical” universe with  $H_0 = 70$ km/s/Mpc, we would have to multiply the **UnitLength\_in\_cm** and the **UnitMass\_in\_g** by  $h = 0.7$ . Another possibility should be to just convert the numbers after the simulation has finished (multiplying Length and mass by h and velocity?).

### 3 Halo Finder + Merger Trees

**ATTENTION:** In the cosmological parameters of the Gadget start file there is the `HubbleParam` variable. This variable has no influence on the used units. It is only used, if we also do an SPH simulation for the gas, because for the cooling it is necessary to convert to physical units.

For me, i still don't understand how the velocity units work. Interesting in this respect is this post <http://www.mpa-garching.mpg.de/gadget/gadget-list/0113.html> on the Gadget mailing list. It states that the Initial Conditions file should contain the peculiar velocity divided by  $\sqrt{a}$ . To clarify the nomenclature (see the post), we let  $x$  denote the comoving coordinates and  $r = a \cdot x$  the physical coordinates. We then define

**Comoving velocity**  $\frac{dx}{dt}$

**Physical velocity**  $\frac{dr}{dt} = H(a) \cdot r + a \cdot \frac{dx}{dt}$

**Peculiar velocity**  $v = a \cdot \frac{dx}{dt}$

therefore the physical velocity is the peculiar velocity plus the Hubble flow. I think the post also states, that the velocity variable in the snapshot files is  $u = v/\sqrt{a}$ , the peculiar velocity divided by  $\sqrt{a}$ . **ATTENTION:** We should check again, whether this is what we assumed in the ICs conversion and in giving the values to Rockstar. The snapshot format is described on page 32 of the users guide. As supposed, the particle positions are (if we did not change the units) in comoving kpc/h. The particle velocities in the snapshot are in  $u$  in internal units, so peculiar velocities can be obtained by multiplying  $u$  with  $\sqrt{a}$ . Therefore, we have to pay attention, that the velocities in Gadget are neither physical nor comoving velocities!.

## 3 Halo Finder + Merger Trees

## 4 SAM

### 4.1 Galacticus

#### 4.1.1 Compilation

### 4.2 Galacticus v0.9.1

Checkout latest revision:

```
bzr checkout http://bazaar.launchpad.net/~abenson/galacticus/v0.9.1/
```

## 5 Hydro