AWESOME documentation
metal enrichment setup 2011ff

Harald Höller, Markus Haider [*]

May 23, 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 acually 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

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

motivates the recording of IO in order to ensure reproducability of simulations. Ideally one would also indicate some physical parameters in the IO file, such as `lingersynIO_h70.out`.

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`. 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 hast to delete the files `linger.dat` and lingerg.dat first.

1.2 Using MUSIC

MUSIC allows to generate zoomed initial conditions. Here is an example configuration used for testing zoomed initial conditions

```
 [setup]
boxlength = 100
zstart = 50
levelmin = 7
levelmin_TF = 7
levelmax = 8
padding = 8
overlap = 4
```

```
ref_offset = 0.4, 0.4, 0.4
ref_extent = 0.3, 0.3, 0.3
align_top = yes
baryons = yes
use_2LPT = yes
use_LLA = no
periodic_TF = yes

[cosmology]
Omega_m = 0.276
Omega_L = 0.724
Omega_b = 0.045
H0 = 70.3
sigma_8 = 0.811
nspec = 0.961
transfer = eisenstein

[random]
seed[7] = 12345
seed[8] = 23456

[output]
##generic FROLIC data format (used for testing)
#format = generic
#filename = debug.hdf5

##ENZO - also outputs the settings for the parameter file
#format = enzo
#filename = ic.enzo

##Gadget-2 (type=1: high-res particles, type=5: rest)
##no gas possible at the moment
format = gadget2
filename = ics_zoom.dat
shift_back      = yes

##Grafic2 compatible format for use with RAMSES
##option 'ramses_nml'=yes writes out a startup nml file
#format = grafic2
#filename = ics_ramses
```

```
#ramses_nml     = yes

[poisson]
fft_fine = yes
accuracy = 1e-5
pre_smooth = 3
post_smooth = 3
smoother = gs
laplace_order = 6
grad_order = 6
```

1.3 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 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. Here an example makefile used for zoomed initial conditions with 128^3 particles on the coarsest grid:

```
#--------------------------------------------------------------------------
# From the list below, please activate/deactivate the options that
# apply to your run. If you modify any of these options, make sure
# that you recompile the whole code by typing "make clean; make".
#
# Look at end of file for a brief guide to the compile-time options.
#--------------------------------------------------------------------------


#---------------------------------------- SFR & cooling

OPT += -DCOOLING
OPT += -DSFR
OPT += -DSTELLARAGE
OPT += -DMETALS
#OPT += -DMETAL_COOLING
OPT += -DMOREPARAMS

OPT += -DWINDS
OPT += -DISOTROPICWINDS

OPT += -DOUTPUTCOOLRATE

#---------------------------------------- Basic operation mode of code
OPT   +=  -DPERIODIC
OPT   +=  -DUNEQUALSOFTENINGS


#---------------------------------------- Things that are always recommended
OPT   +=  -DPEANOHILBERT
OPT   +=  -DWALLCLOCK


#---------------------------------------- TreePM Options
OPT   +=  -DPMGRID=128
OPT   +=  -DPLACEHIGHRESREGION=3
OPT   +=  -DENLARGEREGION=1.2
#OPT   +=  -DASMTH=1.25
#OPT   +=  -DRCUT=4.5


#-------------------------------------- Single/Double Precision
OPT   +=  -DDOUBLEPRECISION
OPT   +=  -DDOUBLEPRECISION_FFTW


#-------------------------------------- Time integration options
OPT   +=  -DSYNCHRONIZATION
#OPT   +=  -DFLEXSTEPS
#OPT   +=  -DPSEUDOSYMMETRIC
#OPT   +=  -DNOSTOP_WHEN_BELOW_MINTIMESTEP
#OPT   +=  -DNOPMSTEPADJUSTMENT


#-------------------------------------- Output
#OPT   +=  -DHAVE_HDF5
#OPT   +=  -DOUTPUTPOTENTIAL
#OPT   +=  -DOUTPUTACCELERATION
#OPT   +=  -DOUTPUTCHANGEOFENTROPY
#OPT   +=  -DOUTPUTTIMESTEP


#-------------------------------------- Things for special behaviour
#OPT   +=  -DNOGRAVITY
#OPT   +=  -DNOTREERND
#OPT   +=  -DNOTYPEPREFIX_FFTW
#OPT   +=  -DLONG_X=60
#OPT   +=  -DLONG_Y=5
#OPT   +=  -DLONG_Z=0.2
#OPT   +=  -DTWODIMS
#OPT   +=  -DSPH_BND_PARTICLES
#OPT   +=  -DNOVISCOSITYLIMITER
#OPT   +=  -DCOMPUTE_POTENTIAL_ENERGY
#OPT   +=  -DLONGIDS
#OPT   +=  -DISOTHERM_EQS
```

```
#OPT   +=  -DADAPTIVE_GRAVSOFT_FORGAS
#OPT   +=  -DSELECTIVE_NO_GRAVITY=2+4+8+16

#--------------------------------------- Testing and Debugging options
#OPT   +=  -DFORCETEST=0.1


#--------------------------------------- Glass making
#OPT   +=  -DMAKEGLASS=262144


#--------------------------------------------------------------------
# Here, select compile environment for the target machine. This may need
# adjustment, depending on your local system. Follow the examples to add
# additional target platforms, and to get things properly compiled.
#--------------------------------------------------------------------

#--------------------------------------- Select some defaults

#CC       =  mpicc               # sets the C-compiler
#OPTIMIZE =  -O2 -Wall -g        # sets optimization and warning flags
#MPICHLIB =  -lmpich


#--------------------------------------- Select target computer

SYSTYPE="leo2"
#SYSTYPE ="home"

#--------------------------------------- Adjust settings for target computer


ifeq ($(SYSTYPE),"leo2")
CC       =  mpicc
OPTIMIZE =   -Wall
GSL_INCL =  -I$(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


ifeq ($(SYSTYPE),"home")
CC = mpicc
OPTIMIZE = -O0 -g -Wall -fsignaling-nans -ftrapping-math
FFTW_INCL = -I/home/dominik/Downloads/fftw-2.1.5/mpi -I/home/dominik/Downloads/fftw-2.1.5
FFTW_LIBS = -L/home/dominik/Downloads/fftw-2.1.5/mpi/.libs -L/home/dominik/Downloads/fftw
endif

ifeq ($(SYSTYPE),"astro-cluster")
CC       =  mpicc
#OPTIMIZE =  -O3 -Wall
#OPTIMIZE = -march=core2 -msse4 -mcx16 -msahf -O2 -pipe
OPTIMIZE = -O2 -g
#MPICHLIB =  -lmpich
#FFTW_INCL = -I/site/fftw/2.1.5/gcc-4.3.0/64/include/
#FFTW_LIBS = -L/site/fftw/2.1.5/gcc-4.3.0/64/lib/

#GSL_INCL =
#GSL_LIBS =
#FFTW_INCL=  -I/usr/local/include/
#FFTW_LIBS=  -L/usr/local/lib/
#MPICHLIB =
#HDF5INCL =
#HDF5LIB  =  -lhdf5 -lz
endif


ifneq (HAVE_HDF5,$(findstring HAVE_HDF5,$(OPT)))
HDF5INCL =
HDF5LIB  =
endif


OPTIONS =  $(OPTIMIZE) $(OPT)

EXEC   = Gadget2
```

```
OBJS   = main.o  run.o  predict.o begrun.o endrun.o global.o  \
 timestep.o  init.o restart.o  io.o    \
 accel.o   read_ic.o  ngb.o  \
 system.o  allocate.o  density.o  \
 gravtree.o hydra.o  driftfac.o  \
 domain.o  allvars.o potential.o  \
         forcetree.o   peano.o gravtree_forcetest.o \
 pm_periodic.o pm_nonperiodic.o longrange.o \
 cooling.o sfr_eff.o

INCL   = allvars.h  proto.h  tags.h  Makefile cooling.h


CFLAGS = $(OPTIONS) $(GSL_INCL) $(FFTW_INCL) $(HDF5INCL)


ifeq (NOTYPEPREFIX_FFTW,$(findstring NOTYPEPREFIX_FFTW,$(OPT)))    # fftw installed with
  FFTW_LIB = $(FFTW_LIBS) -lrfftw_mpi -lfftw_mpi -lrfftw -lfftw
else
ifeq (DOUBLEPRECISION_FFTW,$(findstring DOUBLEPRECISION_FFTW,$(OPT)))
  FFTW_LIB = $(FFTW_LIBS) -ldrfftw_mpi -ldfftw_mpi -ldrfftw -ldfftw
else
  FFTW_LIB = $(FFTW_LIBS) -lsrfftw_mpi -lsfftw_mpi -lsrfftw -lsfftw
endif
endif


LIBS   =   $(HDF5LIB) -g  $(MPICHLIB)  $(GSL_LIBS) -lgsl -lgslcblas -lm $(FFTW_LIB)

$(EXEC): $(OBJS)
$(CC) $(OBJS) $(LIBS)   -o  $(EXEC)

$(OBJS): $(INCL)


clean:
rm -f $(OBJS) $(EXEC)
```

```
#--------------------------------------------------------------------------
#
#   Brief guide to compile-time options of the code. More information
#   can be found in the code documentation.
#
# - PERIODIC:
#     Set this if you want to have periodic boundary conditions.
#
# - UNEQUALSOFTENINGS:
#     Set this if you use particles with different gravitational
#     softening lengths.
#
# - PEANOHILBERT:
#     This is a tuning option. When set, the code will bring the
#     particles after each domain decomposition into Peano-Hilbert
#     order. This improves cache utilization and performance.
#
# - WALLCLOCK:
#     If set, a wallclock timer is used by the code to measure internal
#     time consumption (see cpu-log file).  Otherwise, a timer that
#     measures consumed processor ticks is used.
#
# - PMGRID:
#     This enables the TreePM method, i.e. the long-range force is
#     computed with a PM-algorithm, and the short range force with the
#     tree. The parameter has to be set to the size of the mesh that
#     should be used, (e.g. 64, 96, 128, etc). The mesh dimensions need
#     not necessarily be a power of two.  Note: If the simulation is
#     not in a periodic box, then a FFT method for vacuum boundaries is
#     employed, using an actual mesh with dimension twice(!) that
#     specified by PMGRID.
#
# - PLACEHIGHRESREGION:
#     If this option is set (will only work together with PMGRID), then
#     the long range force is computed in two stages: One Fourier-grid
#     is used to cover the whole simulation volume, allowing the
#     computation of the longe-range force.  A second Fourier mesh is
#     placed on the region occupied by "high-resolution" particles,
#     allowing the computation of an intermediate scale force. Finally,
#     the force on short scales is computed with the tree. This
#     procedure can be useful for "zoom-simulations", provided the
#     majority of particles (the high-res particles) are occupying only
#     a small fraction of the volume. To activate this option, the
#     parameter needs to be set to an integer bit mask that encodes the
#     particle types that make up the high-res particles.
#     For example, if types 0, 1, and 4 form the high-res
#     particles, set the parameter to PLACEHIGHRESREGION=19, because
#     2^0 + 2^1 + 2^4 = 19. The spatial region covered by the high-res
#     grid is determined automatically from the initial conditions.
#     Note: If a periodic box is used, the high-res zone may not intersect
#     the box boundaries.
#
# - ENLARGEREGION:
#     The spatial region covered by the high-res zone has a fixed size
#     during the simulation, which initially is set to the smallest
#     region that encompasses all high-res particles. Normally, the
#     simulation will be interrupted if high-res particles leave this
#     region in the course of the run. However, by setting this
#     parameter to a value larger than one, the size of the high-res
#     region can be expanded, providing a buffer region.  For example,
#     setting it to 1.4 will enlarge its side-length by 40% (it remains
#     centered on the high-res particles). Hence, with this setting, the
#     high-res region may expand or move by a limited amount.
#     Note: If SYNCHRONIZATION is activated, the code will be able to
#     continue even if high-res particles leave the initial high-res
#     grid. In this case, the code will update the size and position of
#     the grid that is placed onto the high-resolution region
#     automatically. To prevent that this potentially happens every
#     single PM step, one should nevertheless assign a value slightly
#     larger than 1 to ENLARGEREGION.
#
# - ASMTH:
#     This can be used to override the value assumed for the scale that
#     defines the long-range/short-range force-split in the TreePM
#     algorithm. The default value is 1.25, in mesh-cells.
#
# - RCUT:
#     This can be used to override the maximum radius in which the
#     short-range tree-force is evaluated (in case the TreePM algorithm
#     is used). The default value is 4.5, given in mesh-cells.
```

```
#
# - DOUBLEPRECISION:
#     This makes the code store and compute internal particle data in
#     double precision. Note that output files are nevertheless written
#     by converting the particle data to single precision.
#
# - DDOUBLEPRECISION_FFTW:
#     If this is set, the code will use the double-precision version of
#     FTTW, provided the latter has been explicitly installed with a
#     "d" prefix, and NOTYPEPREFIX_FFTW is not set. Otherwise the
#     single precision version ("s" prefix) is used.
#
# - SYNCHRONIZATION:
#     When this is set, particles are kept in a binary hierarchy of
#     timesteps and may only increase their timestep if the new
#     timestep will put them into synchronization with the higher time
#     level.
#
# - FLEXSTEPS:
#     This is an alternative to SYNCHRONIZATION. Particle timesteps are
#     here allowed to be integer multiples of the minimum timestep that
#     occurs among the particles, which in turn is rounded down to the
#     nearest power-of-two devision of the total simulated
#     timespan. This option distributes particles more evenly over
#     individual system timesteps, particularly once a simulation has
#     run for a while, and may then result in a reduction of work-load
#     imbalance losses.
#
# - PSEUDOSYMMETRIC:
#     When this option is set, the code will try to "anticipate"
#     timestep changes by extrapolating the change of the acceleration
#     into the future. This can in certain idealized cases improve the
#     long-term integration behaviour of periodic orbits, but should
#     make little or no difference in most real-world applications. May
#     only be used together with SYNCHRONIZATION.
#
# - NOSTOP_WHEN_BELOW_MINTIMESTEP:
#     If this is activated, the code will not terminate when the
#     timestep falls below the value of MinSizeTimestep specified in
#     the parameterfile. This is useful for runs where one wants to
#     enforce a constant timestep for all particles. This can be done
#     by activating this option, and by setting MinSizeTimestep and
#     MaxSizeTimestep to an equal value.
#
# - NOPMSTEPADJUSTMENT:
#     When this is set, the long-range timestep for the PM-force
#     computation (when the TreePM algorithm is used) is always
#     determined by MaxSizeTimeStep.  Otherwise, it is determined by
#     the MaxRMSDisplacement parameter, or MaxSizeTimeStep, whichever
#     gives the smaller step.
#
# - HAVE_HDF5:
#     If this is set, the code will be compiled with support for input
#     and output in the HDF5 format. You need to have the HDF5
#     libraries and headers installed on your computer for this option
#     to work. The HDF5 format can then be selected as format "3" in
#     Gadget's parameterfile.
#
# - OUTPUTPOTENTIAL:
#     This will make the code compute gravitational potentials for
#     all particles each time a snapshot file is generated. The values
#     are then included in the snapshot file. Note that the computation
#     of the values of the gravitational potential costs additional CPU.
#
# - OUTPUTACCELERATION:
#     This will include the physical acceleration of each particle in
#     snapshot files.
#
# - OUTPUTCHANGEOFENTROPY:
#     This will include the rate of change of entropy of gas particles
#     in snapshot files.
#
# - OUTPUTTIMESTEP:
#     This will include the current timesteps of all particles in the
#     snapshot files.
#
# - NOGRAVITY
#     This switches off gravity. Useful only for pure SPH simulations
#     in non-expanding space.
#
```

```
# - NOTREERND:
#     If this is not set, the tree construction will succeed even when
#     there are a few particles at identical locations. This is done by
#     ‘rerouting’ particles once the node-size has fallen below 1.0e-3
#     of the softening length. When this option is activated, this will
#     be surpressed and the tree construction will always fail if there
#     are particles at extremely close coordinates.
#
# - NOTYPEPREFIX_FFTW:
#     This is an option that signals that FFTW has been compiled
#     without the type-prefix option, i.e. no leading "d" or "s"
#     characters are used to access the library.
#
# - LONG_X/Y/Z:
#     These options can be used together with PERIODIC and NOGRAVITY only.
#     When set, the options define numerical factors that can be used to
#     distorts the periodic simulation cube into a parallelepiped of
#     arbitrary aspect ratio. This can be useful for idealized SPH tests.
#
# - TWODIMS:
#     This effectively switches of one dimension in SPH, i.e. the code
#     follows only 2d hydrodynamics in the xy-, yz-, or xz-plane. This
#     only works with NOGRAVITY, and if all coordinates of the third
#     axis are exactly equal. Can be useful for idealized SPH tests.
#
# - SPH_BND_PARTICLES:
#     If this is set, particles with a particle-ID equal to zero do not
#     receive any SPH acceleration. This can be useful for idealized
#     SPH tests, where these particles represent fixed "walls".
#
# - NOVISCOSITYLIMITER:
#     If this is set, the code will not try to put an upper limit on
#     the viscous force in case an implausibly high pair-wise viscous
#     force (which may lead to a particle 'reflection' in case of poor
#     timestepping) should arise. Note: For proper settings of the
#     timestep parameters, this situation should not arise.
#
# - COMPUTE_POTENTIAL_ENERGY:
#     When this option is set, the code will compute the gravitational
#     potential energy each time a global statistics is computed. This
#     can be useful for testing global energy conservation.
#
# - LONGIDS:
#     If this is set, the code assumes that particle-IDs are stored as
#     64-bit long integers. This is only really needed if you want to
#     go beyond ~2 billion particles.
#
# - ISOTHERM_EQS:
#     This special option makes the gas behave like an isothermal gas
#     with equation of state P = cs^2 * rho. The sound-speed cs is set by
#     the thermal energy per unit mass in the intial conditions,
#     i.e. cs^2=u. If the value for u is zero, then the initial gas
#     temperature in the parameter file is used to define the sound speed
#     according to cs^2 = 3/2 kT/mp, where mp is the proton mass.
#
# - ADAPTIVE_GRAVSOFT_FORGAS:
#     When this option is set, the gravitational softening lengths used for
#     gas particles is tied to their SPH smoothing length. This can be useful
#     for dissipative collapse simulations. The option requires the setting
#     of UNEQUALSOFTENINGS.
#
# - SELECTIVE_NO_GRAVITY:
#     This can be used for special computations where one wants to
#     exclude certain particle types from receiving gravitational
#     forces. The particle types that are excluded in this fashion are
#     specified by a bit mask, in the same as for the PLACEHIGHRESREGION
#     option.
#
# - FORCETEST:
#     This can be set to check the force accuracy of the code. The
#     option needs to be set to a number between 0 and 1 (e.g. 0.01),
#     which is taken to specify a random fraction of particles for
#     which at each timestep forces by direct summation are
#     computed. The normal tree-forces and the correct direct
#     summation forces are collected in a file. Note that the
#     simulation itself is unaffected by this option, but it will of
#     course run much(!) slower, especially if
#     FORCETEST*NumPart*NumPart >> NumPart. Note: Particle IDs must
#     be set to numbers >=1 for this to work.
#
```

```
# - MAKEGLASS
#     This option can be used to generate a glass-like particle
#     configuration. The value assigned gives the particle load,
#     which is initially generated as a Poisson sample and then
#     evolved towards a glass with the sign of gravity reversed.
#
#--------------------------------------------------------------------------
```

2.2 Gadget2 start file

Here is an example of the parameter file used to start zoomed initial conditions. The smoothing might need to be adjusted

```
 %  Relevant files

InitCondFile      /scratch/c706130/zoom/ics_zoom.dat
OutputDir          /scratch/c706130/zoom/output
EnergyFile         energy.txt
InfoFile           info.txt
TimingsFile        timings.txt
CpuFile            cpu.txt

RestartFile        restart
SnapshotFileBase   snapshot

OutputListFilename gadget_timer_350.txt

% CPU time -limit

TimeLimitCPU      3600000  % = 10 hours
ResubmitOn        0
ResubmitCommand   my-scriptfile


% Code options


ICFormat                 1
```

```
SnapFormat              1
ComovingIntegrationOn   1

TypeOfTimestepCriterion  0
OutputListOn            1
PeriodicBoundariesOn    1

%  Caracteristics of run

TimeBegin           0.0196078     % Begin of the simulation (z=23)
TimeMax             1.0

Omega0               0.276 %WMAP7 values
OmegaLambda           0.724
OmegaBaryon           0.045 %not relevant
HubbleParam           70.3
BoxSize               100000
% Output frequency



TimeBetSnapshot        0.040207
TimeOfFirstSnapshot    1.8384163

CpuTimeBetRestartFile     36000.0    ; here in seconds
TimeBetStatistics         0.05

NumFilesPerSnapshot       1
NumFilesWrittenInParallel 1



% Accuracy of time integration

ErrTolIntAccuracy      0.025

MaxRMSDisplacementFac  0.2

CourantFac             0.15
```

```
MaxSizeTimestep       0.03
MinSizeTimestep       0.0




% Tree algorithm, force accuracy, domain update frequency

ErrTolTheta            0.75
TypeOfOpeningCriterion 0
ErrTolForceAcc         0.005


TreeDomainUpdateFrequency    0.1


%  Further parameters of SPH

DesNumNgb              50
MaxNumNgbDeviation     2
ArtBulkViscConst       0.8
InitGasTemp            0
MinGasTemp             0


% Memory allocation

PartAllocFactor       4.0
TreeAllocFactor       4.0
BufferSize            600          % in MByte


% System of units

UnitLength_in_cm         3.085678e21        ;  1.0 kpc
UnitMass_in_g            1.989e43           ;  1.0e10 solar masses
UnitVelocity_in_cm_per_s 1e5                ;  1 km/sec
GravityConstantInternal  0
```

```
% Softening lengths

MinGasHsmlFractional 0.25


SofteningGas       15.0
SofteningHalo      15.0
SofteningDisk      0.0
SofteningBulge     0.0
SofteningStars     15.0
SofteningBndry     40.0

SofteningGasMaxPhys       6.0
SofteningHaloMaxPhys      6.0
SofteningDiskMaxPhys      0.0
SofteningBulgeMaxPhys     0.0
SofteningStarsMaxPhys     6.0
SofteningBndryMaxPhys     20.0

CoolingOn                          1
StarformationOn                    1

FactorSN                           0.1
FactorEVP                          1000
TempSupernova                      1e+08
TempClouds                         1000
CritOverDensity                    57.7
CritPhysDensity                    0
MaxSfrTimescale                    2.1
WindEfficiency                     2
WindEnergyFraction                 0.5
WindFreeTravelLength               0.25
WindFreeTravelDensFac              5
```

2.3 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 $H_0 = 100h\ \mathrm{km\,s^{-1}Mpc^{-1}}$. 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 $10^{10}\mathrm{M}_\odot$/h.

I therefore suppose that if we would like to simulate a "physical" universe with $H_0 =$ 70km/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?).

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 repsect 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 $\frac{\mathrm{d}x}{\mathrm{d}t}$

Physical velocity $\frac{\mathrm{d}r}{\mathrm{d}t} = H(a) \cdot r + a \cdot \frac{\mathrm{d}x}{\mathrm{d}t}$

Peculiar velocity $v = a \cdot \frac{\mathrm{d}x}{\mathrm{d}t}$

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 suposed, 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!.

2.4 VisualÃzing Gadget Snapshots

2.5 Splotch

Splotch makes very nice renderings of Gadget Files. It can be obtained from `http://www.mpa-garching.mpg.de/~kdolag/Splotch/` and it compiles fine without bigger problems. It is important to note, that Splotch can only read the Snapshot format 2 of Gadget files (the snapshot format has to be specified in the parameter file of the gadget simulation).

Then it works with parameter files, where the particle types and the viewpoint have to be specified. To see how it works, on can use the example parameter and binary file (Homepage under the Section "Test Example and Documentation").

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