MP-Gadget User Guide

Contents

1 Introduction

Cosmological simulations evolve a distribution of N-body particles, which are supposed to be Monte-Carlo samples of an underlying matter field (and **not** physical dark matter particles), under gravity. The initial conditions for these particles are set using cosmological initial conditions and a Gaussian random field. MP-Gadget provides a comprehensive package for both initial conditions and cosmological simulation. You can, however, use separate packages to generate initial conditions. We also include physics for gas pressure, cooling and subgrid models for star formation, stellar winds and supermassive black hole accretion and feedback.

You can find more information about some of the algorithms implemented in the code in the Gadget-2 paper: `https://arxiv.org/abs/astro-ph/0505010`

1.1 Parallelization

MP-Gadget is a highly parallel code. It supports both distributed memory parallelization using MPI and shared memory parallelization using OpenMP threads. Unlike many other versions of Gadget, the shared memory parallelization is computationally efficient. On modern hyperthreaded SMP systems such as Stampede we have found that the best results are obtained by running with one MPI rank per socket, and one thread per logical core. Thus for a cluster where each node has 48 logical processors spread over 2 sockets we would recommend a simulation on 2 nodes use 4 MPI ranks and 24 OpenMP threads per MPI rank. Note that because floating point addition is not associative, and threading will operate on different particles at different times, repeats of threaded runs will not produce bit-identical output. However their outputs should be statistically close.

The number of MPI ranks is usually controlled by running the code under "mpirun", for example: "mpirun -np $NMPI". This does depend on the cluster used so consult your documentation.

The number of OpenMP threads is controlled by setting the OMP_NUM_THREADS variable. For example: "export OMP_NUM_THREADS=24" before running code. In the below we will assume that you are running the code in one of these modes.

1.2 Basic Usage steps

1.2.1 Initial Power Spectrum

Generate an initial power spectrum using linear cosmological theory as implemented in a Boltzmann code, either CLASS or CAMB. The output of this is a file called "*matterpow_redshift.dat" for CAMB or "*_pk_number.dat" for CLASS. For simulations including gas you will also need an initial transfer function file, which specifies how the gas and cold dark matter differ initially. This file will be called "*transfer_redshift.dat" for CAMB or "*_tk_number.dat" for CLASS.

You can do this automatically with the python3 script "tools/make_class_power.py", which reads an MP-GenIC parameter file and generates an initial power spectrum consistent with those cosmological parameters. To use the script you first need to install the classylss python module with "**pip install --user classylss**"

Then invoke the script with: **python make_class_power.py $paramfile_path/paramfile.genic**

This will generate a power spectrum appropriate for the initial redshift of the simulation. You will also want to check that the simulation is producing results in good agreement with linear theory. For this reason, generate lower redshift power spectra with the "–extraz" option, eg, "–extraz 49 9 2 1 0" will generate power spectra at redshifts $z = 49, 9, 2, 1, 0$.

When the code runs it will generate power spectra as it goes which can be compared to these initial power spectra.

1.2.2 Initial Conditions

Use the initial power spectrum and transfer function to generate a set of initial conditions, initial positions and velocities for the N-body particles. The built-in initial conditions generator is called MP-GenIC. Built versions of MP-GenIC can be found in the genic/ subdirectory of the main git repo. They can be run with:

MP-GenIC $paramfile_path/paramfile.genic

The initial positions are set by perturbing a homogeneous distribution of particles to match a density field. The density field is the Fourier transform of a Gaussian random field with the input power spectrum. It is set separately for each species.

A clear explanation of how initial conditions are generated can be found in Scoccimarro 97, section 2: https://arxiv.org/abs/astro-ph/9711187]

1.2.3 Running MP-Gadget

The main code, MP-Gadget, evolves the initial distribution of particles. It has numerous options and capabilities. Simple examples can be found in the examples/ directory. Code options are described in other sections of this document and each option has a short description in gadget/params.c Built versions of MP-Gadget can be found in the gadget/ subdirectory of the main git repo. They can be run with:

MP-Gadget$paramfile_path/paramfile.gadget

Note that the MP-Gadget parameter file is separate from the MP-GenIC parameter file.

MP-Gadget will run until the walltime is TimeLimitCPU (set in the parameter file) seconds, output a final snapshot and then stop. Strictly it will run until it believes there is not enough time between the current walltime and it's time limit to complete another PM timestep and write a snapshot.

Along the way, if it passes a scale factor which is an output time, it will write the particle table to disc. If it crashes it will output a backtrace showing in which routine the crash occurred.

1.2.4 Restarting

You can restart MP-Gadget from the last output simulation by appending '1' to the command line:

MP-Gadget paramfile.gadget 1

or from a specific snapshot by appending '2' and the snapshot number, eg,

MP-Gadget paramfile.gadget 2 3

restarts from the third snapshot (**this means the snapshot labeled _003, not the third snapshot since it starts with _000**). There is no loss in accuracy from doing this: the code will produce almost identical outputs when restarting from a simulation snapshot to when it was run without a restart.

2 Compile-time Options

Most compile time options have been removed from the MP-Gadget source code. Those that remain are set by defining them in the Options.mk file. You always need an Options.mk: it sets the default compiler optimization flags. Options.mk.example provides good defaults for the gcc compiler. Other defaults for specific systems are provided in the platform-options/ subdirectory.

If you find the code is outputting strange results, check for compiler bugs by running the code without optimizations.

You will need the GNU Scientific Library, or GSL. All other dependencies are included.

Remaining compile-time options are:

DEBUG. Enable various debugging options. This flag performs a number of checks at every timestep to ensure internal consistency of the code. The code will run slower, but this flag should be enabled if you suspect a bug.

VALGRIND. Disable MP-Gadget's custom memory allocator and use the system malloc. This is useful for checking memory consistency with Valgrind or address sanitizer, and can be useful for running a small simulation on a laptop.

NO_OPENMP_SPINLOCK. Uses an alternate locking implementation. This is slower and should be used only if the code does not compile without it. Necessary on Mac.

EXCUR_REION. Compile with data structures for the excursion set hydrogen reionization model.

USE_CFITSIO. Compile against the CFITSIO library. This is used only for outputting potential plane files for lensing maps.

3 Snapshot Output

MP-Gadget will output snapshots in directories called "PART_*" where * is an increasing integer from 000 to 999. It will output powerspectra every PM timestep, and friends of friends halo catalogues in PIG_*.

The snapshots are in bigfile format, which is a hierarchical tree of directories with binary files at the bottom. You can access blocks with, eg:

bf = bigfile.BigFile("PART_001") pos = bf["1/Position"][:]

which gets you a 3 x N particle position array.

The numbers 1 - 5 correspond to the different types of particle in the simulation: 0 is gas particles (baryons) 1 is CDM, 2 is neutrinos, 3 is unused, 4 is stars and 5 is black holes.

There are also header attributes, which you can read either with:

bf["Header"].attrs["Time"]

or by opening PART_001/Header/attr-v2 as a text file.

You can plot the matter distribution of a snapshot by using "tools/plot_structure.py". To do this you must first "pip install --user nbodykit"

3.1 Memory Usage

The code prints real memory requirements per particle on startup. These change sometimes and will be larger if the DEBUG compile flag is on. At time of writing they are:
Size of particle structure 168 [bytes]
Size of blackhole structure 208 [bytes]
Size of sph particle structure 176 [bytes]
Size of star particle structure 96 [bytes]

For dark matter only simulations the relevant number is 'size of particle structure'. For other simulations only the sph and particle structures matter.

The code needs memory for the number of particles in the initial conditions, plus a bit more for imbalance and star formation, which is controlled by the parameter PartAllocFactor. By default

this is 1.5, but for pure DM with no star formation you can make it a bit smaller, maybe as low as 1.1. To do any operations on these structures, the code allocates extra memory. The peak memory usage is often during the Fourier grids during a PM step. It is: $4 \times \text{Nmesh} \times 8$ bytes. By default $\text{Nmesh} = 3 \times \text{Npart}$, but if you are short on memory you can reduce it. $\text{Nmesh} = 2 \times \text{Npart}$ will give the same results, but will be a little slower. There are also a few other allocations but they are either small or not really necessary, in the sense that if memory is low it will just make them smaller at some compute cost. The total memory for them should be about 10% of the particle allocation.

Note that unless you have VALGRIND defined, the code pre-allocates 0.6 of the available memory at startup, to avoid fragmentation. So the memory usage shown in top will not be accurate. If the code runs out of memory, it will print a message explaining which allocation caused it to run out, which can help you know which parameter to change.

3.2 Description of Header Fields in Snapshots and ICs

The following flags are found in both initial conditions and evolved snapshots:

BoxSize Size of the box in internal units.

HubbleParam Hubble parameter in $H_0/(100\text{km/s})$. Used only in the cooling and star formation code. The units of the gravity computation are independent of it.

Omega0 Total matter density at $z = 0$. Includes baryons, massive neutrinos and CDM. The code checks that this matches the total mass of particles in the snapshot.

OmegaBaryon Cosmic gas (baryon) density at $z = 0$. Used to compute some thresholds in cooling and star formation code.

OmegaLambda Dark energy density at $z = 0$. The code enforces $\Omega_K = 1 - \Omega_0 - \Omega_\Lambda$. Used in the background evolution only.

Redshift Redshift of current snapshot.

Time Scale factor of current snapshot, $a = 1/(1 + z)$.

UnitLength_in_cm Internal length unit in cm/h. By default 1 kpc/h.

UnitMass_in_g Internal length unit in g/h. By default $10^{10} M_{\odot}$/h.

UnitVelocity_in_cm_per_s Internal velocity unit in cm/s. By default 1 km/s.

UsePeculiarVelocity Specifies the velocity units of the snapshots. If UsePeculiarVelocity = 1 then snapshots save v, the physical peculiar velocity, to the snapshots. If UsePeculiarVelocity = 0 then we save $a \times v$ in snapshots. Note that neither snapshot format matches Gadget-2, which saves $v/\sqrt{a}$ in both ICs and snapshots. To preserve compatibility with existing ICs, if UsePeculiarVelocity = 0, the IC velocities are assumed to be $v/\sqrt{a}$.

TotNumPart Six element array containing the total number of particles in the simulation in each particle type.

MassTable Six element array containing the initial mass of particles of each particle type in internal units. Note that in simulations with star formation, not all gas particles will have the mass specified here. If the mass in this table is zero, you should examine the Mass array attached to each particle.

The following flags are found only in the initial conditions header:

FractionNuInParticles Fraction of the cosmic density of massive neutrinos, Ω_{ν}, found in neutrino particles rather than the linear response model.

InvertPhase Signifies if the phase of the Gaussian random field in the initial conditions has been inverted for generating paired simulations.

Seed Random seed used for generating the initial conditions.

The following flags are found only in the snapshot header:

CMBTemperature Temperature of the CMB today in K. For computing the background radiation density.

CodeVersion Code version (currently 5.0.1-dev1), followed by the git hash of the commit used to generate this snapshot.

CompilerSettings Compiler string used to compile MP-Gadget.

DensityKernel The density kernel used for by the SPH code for this snapshot. 1 for a cubic spline, 2 for a quintic spline and 4 for a quartic spline.

TimeIC Initial scale factor of the simulation

TotNumPartInit Six element array showing the number of particles of each type in the ICs.

3.3 Description of Snapshot Fields

The fields in the snapshot generally follow Gadget-2. Some fields are optional, or only present for a particular particle type. Each field has a width, which is the number of values per particle.

Position (3) Position of each particle in internal units relative to the lower-left corner of the box.

Velocity (3) Velocity of each particle in internal units. If UsePeculiarVelocity = 1 then this is the physical peculiar velocity. If UsePeculiarVelocity = 0 then this is $a \times v$ in snapshots and $v/\sqrt{a}$ in the ICs. Note that Gadget-2 saves $v/\sqrt{a}$ in both ICs and snapshots.

Mass Mass of each particle in internal units. Currently all particle types store their mass here, but we may in future omit the CDM particle mass, which is stored unambiguously in the MassTable header field.

ID 64-bit unique IDs of each particle. Newly created stars have the same lower bits in IDs as their progenitor gas particles.

Potential Gravitational potential experienced by this particle. Optional: present if OutputPotential = 1.

GroupID ID of the FoF halo to which this particle belongs. -1 is particle does not belong to a halo. Optional: present if SnapshotWithFoF = 1.

Generation For gas particles, the number of stars spawned by the gas particle. For stars and black holes, the number of particles spawned by the gas particle before this one.

SmoothingLength For gas particles only. The SPH smoothing length of the gas particle in internal length units.

Density For gas particles only. The density of the gas particle in internal units.

EgyWtDensity For gas particles only. Optional: present if DensityIndependentSphOn = 1. The entropy-weighted density of the particle.

InternalEnergy For gas particles only. The internal energy (energy per unit mass) in internal units. Used to set the Entropy of the particle on reload.

ElectronAbundance For gas particles only. The number density of free electrons per hydrogen atom. Defined so that the maximum for fully ionized primordial gas is ~ 1.15.

NeutralHydrogenFraction For gas particles only. The neutral hydrogen fraction, derived from the cooling rate network. Unlike Gadget-2, this is roughly unity even for gas on the star-forming equation of state. Written only, not read.

StarFormationRate For gas particles only. The star formation rate for particles on the star-forming equation of state.

BirthDensity For star particles only. The density of the gas particle when the star was formed.

StarFormationTime For star and black hole particles only. The scale factor at which the star or black hole particle was formed.

Metallicity For gas and star particles: the metallicity of the particle.

BlackholeMass For black hole particles only. The accretion mass of the black hole. This may differ from the dynamical mass.

BlackholeAccretionRate For black hole particles only. The accretion rate of the black hole.

BlackholeProgenitors For black hole particles only. The number of seed black holes that merged into this one.

BlackholeMinPotPos For black hole particles only. The position of the particle nearest the potential minimum of the halo the last time this black hole was active.

BlackholeMinPotVel For black hole particles only. The velocity of the particle nearest the potential minimum of the halo the last time this black hole was active.

BlackholeJumpToMinPot For black hole particles only. An internal flag which is always zero when written.

4 GenIC Parameter File

The purpose of MP-GenIC is to create simulation initial conditions. These are a list of particle positions, velocities and particle IDs, together with some simulation metadata. We support generating cosmological initial conditions, which means a periodic box filled with an almost homogeneous particle distribution. Initial conditions are generated by convolving a homogeneous pre-initial particle distribution with a displacement field. The displacement field is generated by inverse Fourier transforming a Gaussian random field generated from the cosmological matter power spectrum at the initial simulation redshift.

The homogeneous pre-initial particle distribution may be a regular grid or a Lagrangian glass. Unusually our current default is for the gas to be a Lagrangian glass and the CDM to be a grid. A grid induces a systematic (visually obvious in a picture of the particle distribution) at the scale of the grid separation. A glass (particles initially distributed randomly and then evolved with reversed gravity) induces noise with a power spectrum going like k^4.

MP-GenIC currently supports only first order Lagrangian perturbation theory, because we are mostly interested in multiple fluid simulations.

4.1 *Required Parameters*

OutputDir=str, REQUIRED default=‘’

This is the pathname of the directory that holds all the output generated by MP-GenIC (the initial conditions).

FileBase=str, REQUIRED default=‘’

Base-filename of output files. ICs will be created at OutputDir/FileBase and stored in Bigfile format.

BoxSize=dbl, REQUIRED default=0

The size of the periodic box (in code units: by default comoving kpc/h) encompassing the simulation volume. For cosmology you need a box large enough that the mode at the scale of the box is still linear at the final redshift of the simulation.

Ngrid=int, REQUIRED default=0

Cube root of the number of particles of each species particles created. For example Ngrid = 128 will run a small 128^3 simulation.

WhichSpectrum=int, REQUIRED default=2

Initial power spectrum model to use. “1” selects Eisenstein & Hu spectrum, “2” selects a tabulated power spectrum in the file ’FileWithInputSpectrum’.

FileWithInputSpectrum=str, REQUIRED default=‘’

File containing input power spectrum, from CLASS or CAMB.

Omega0=dbl, REQUIRED default=0.2814

Total matter density, cdm + baryons + massive neutrinos at $z = 0$. Cosmological matter density parameter in units of the critical density at $z = 0$.

OmegaBaryon=dbl, REQUIRED default=0.0464

Baryon density in units of the critical density at z = 0. This is used even for a pure dark matter simulation to construct the combined total species transfer functions.

OmegaLambda=dbl, REQUIRED default=0.7186

Dark energy density at z=0. Cosmological vacuum energy density (cosmological constant) in units of the critical density at z = 0. The code enforces $\Omega_K = 1 - \Omega_0 - \Omega_\Lambda$, so for a flat universe it is important to set this to 1 - Omega0.

HubbleParam=dbl, REQUIRED default=0.697

Value of the Hubble constant in units of 100 km s^{-1} Mpc^{-1}. This is only used in cosmological integrations when conversions to cgs units are required (e.g. for radiative cooling physics). The value of HubbleParam is not needed (in fact, it does not enter the computation at all) in purely collisionless simulations. This is because of the definition of GADGET-2's system of units, which eliminates an explicit appearance of the value of the Hubble constant in the dynamical equations. In MP-GenIC it is used only to set the temperature of warm dark matter.

ProduceGas=int, REQUIRED default=0

Should we create baryon particles? 1 = Produce gas, 0 = no gas, just DM.

Redshift=dbl, REQUIRED default=99

Starting redshift of the simulation. To avoid transients this should usually be around 100.

Seed=int, REQUIRED default=0

seed for IC-generator. Simulations are initialised by perturbing a homogeneous particle distribution with a Gaussian random field. The power spectrum of the Gaussian random field is the cosmological power spectrum. This sets the random phases of the Gaussian random field. Simulations initialised with the same seed will have large scale objects in about the same place.

4.2 *Science Parameters*

4.2.1 `Gas and Transfer Functions`

These parameters control the way a second particle species of gas (type 0) is initialised.

UnitaryAmplitude=int, OPTIONAL default=0

This controls whether or not each individual Fourier mode sampled from the power spectrum is scattered, ie, multiplied by a zero-mean random number before displacing particles. This scattering mimics the effect of cosmic variance from modes on larger scales than are included in the box. However, you get a much better agreement with linear theory if you do not include the scatter. If UnitaryAmplitude = 0 each mode is scattered. If UnitaryAmplitude = 1 each mode is not scattered.

NgridGas=int, OPTIONAL default=Ngrid

As Ngrid but for gas (type 0) particles. Defaults to zero if ProduceGas is off, and the same as Ngrid if ProduceGas is on. Displace

DifferentTransferFunctions=int, OPTIONAL default=1

Use species specific transfer functions for baryon and CDM. The strongly recommended default for multiple species. However, its effect reduces at lower redshift and is close to zero for $z < 2$. It is a few percent for $z = 10 - 2$ and a few tens of percent for $z = 100 - 10$.

ScaleDepVelocity=int, OPTIONAL default=1

Use scale dependent velocity transfer functions obtained by differentiating sync. gauge velocity. Requires two transfer functions. This is required because the different transfer functions for baryons and CDM also impart different velocities. Never disable this when DifferentTransferFunctions=1.

FileWithTransferFunction=str, OPTIONAL default=

File containing CLASS formatted transfer functions with extra metric transfer functions=y. Can be generated with the make_class_powerpy script. You **must** use CLASS not CAMB: CAMB has different unit conventions for the velocity transfer functions and the code does not understand them.

MakeGlassGas=int, OPTIONAL default=-1

Use a Lagrangian glass for the gas particles. First a homogeneous distribution of gas particles

is generated and then the combined distribution of gas and CDM is evolved with reversed gravity so that chance juxtapositions of the two fluids are destroyed. The default is to have a glass if the CDM and baryons have different transfer function, which is required to match the relative power of the two species.

MakeGlassCDM=int, OPTIONAL default=0

Use a Lagrangian glass for the CDM particles. The regular grid is displaced from the regular grid for gas to avoid infinite forces. The default is to generate CDM on a regular grid.

4.2.2 `Neutrinos`

These parameters support the massive neutrino model. By default the total neutrino mass is zero.

NgridNu=int, OPTIONAL default=0

Number of neutrino particles, for particle or hybrid neutrino simulations. Generally this does not need to be set.

MNue=dbl, OPTIONAL default=0

First neutrino mass in eV.

Mnum=dbl, OPTIONAL default=0

Second neutrino mass in eV.

Mnut=dbl, OPTIONAL default=0

Third neutrino mass in eV.

Max_nuvel=dbl, OPTIONAL default=5000

Maximum neutrino velocity sampled from the F-D distribution. Used for hybrid neutrino simulations.

4.2.3 `Optional Cosmology Models`

These parameters exist for supporting non-standard cosmology options. Generally these cosmologies only affect the background evolution. They usually have the same meaning as in CLASS.

Omega_fld=dbl, OPTIONAL default=0

Energy density of a "dark energy fluid" at $z = 0$. This is an extra exotic species added to the background evolution.

w0_fld=dbl, OPTIONAL default=-1

Dark energy equation of state for the exotic fluid. We have $w = w_0 = (1 - a)w_a$.

wa_fld=dbl, OPTIONAL default=0

Dark energy evolution parameter.

Omega_ur=dbl, OPTIONAL default=0

Extra radiation density above the standard, following the CLASS convention. This could be used to model, e.g. a sterile neutrino. Only affects the background.

MWDM_therm=dbl, OPTIONAL default=0

Assign a thermal velocity to the DM and imposes a cutoff in the initial CDM power spectrum. Specifies WDM particle mass in keV.

CMBTemperature=dbl, OPTIONAL default=2.7255

CMB temperature in K. Sets Ω_r.

RadiationOn=dbl, OPTIONAL default=1

Include radiation in the background evolution. There is no reason to switch this off.

PrimordialAmp=dbl, OPTIONAL default=2.215e-9

This parameter is ignored by MP-GenIC. It is used by the external CLASS script to set the power spectrum amplitude.

PrimordialIndex=dbl, OPTIONAL default=0.971

The scalar spectral index of the primordial power spectrum, n_s. Used if WhichSpectum = 1 and by the external CLASS script to set the power spectrum slope.

PrimordialRunning=dbl, OPTIONAL default=0

This parameter is ignored by MP-GenIC. It is used by the external CLASS script to set the

"running" (the change in slope with scale) of the primordial power spectrum.

The following parameters are for changing the normalization of the power spectrum. A common thing for simulators to do is to generate a power spectrum at $z = 0$, and initialise the simulation with this power spectrum divided by a growth function to reach the starting redshift. We do not do this by default because it is hard to make work with massive neutrinos and when using different transfer functions for gas and cold dark matter. However the parameters are retained. It is not recommended to use them unless you have a simulation pipeline that relies on this technique.

Sigma8=dbl, OPTIONAL default=-1

Renormalise Sigma8 to this number if positive. power spectrum normalization.

InputPowerRedshift=dbl, OPTIONAL default=-1

Redshift at which the input power is. Power spectrum will be rescaled to the initial redshift. Negative disables rescaling.

4.2.4 `Unit System`

This controls the unit system used in MP-GenIC. The values are stored in the snapshot headers. It is dangerous to change them because most analysis packages assume the Gadget defaults without checking them! Default units are kpc/h, $10^10 M_\odot/h$ and km/s. Note that Gadget velocities are "comoving": they are multiplied by an extra factor of $\sqrt{a}$.

UnitVelocity_in_cm_er_s=dbl, OPTIONAL default=1e5

Velocity unit in cm/sec. Default is 1 km/s. This sets the internal velocity unit in cm/sec. The above choice is convenient – it sets the velocity unit to km/sec. Note that the specification of UnitLength in cm, UnitMass in g, and UnitVelocity in cm per s also determines the internal unit of time. The definitions made above imply that in internal units the Hubble constant has a numerical value independent of h and hence HubbleParam. For the numerical examples above, the Hubble constant has always the value 0.1 in internal units, independent of h, and the Hubble time is always 10 in internal units, with one internal time unit corresponding to 9.8×108 yr/h.

UnitLength_in_cm=dbl, OPTIONAL default=CM_PER_MPC/1000

Length unit in cm. Default is 1 kpc/h (comoving). This sets the internal length unit in cm/h, where $H_0 = 100h$ km s^{-1} Mpc^{-1}. The above choice is convenient – it sets the length unit to 1.0 kpc/h.

UnitMass_in_g=dbl, OPTIONAL default=1.989e43

Mass unit in g. Default is $10^{10} M_\odot/h$. This sets the internal mass unit in g/h, where $H_0 = 100h$ km s^{-1} Mpc^{-1}.

4.3 *Other Parameters*

These parameters are numerical or specialized and generally do not need to be changed.

MaxMemSizePerNode=dbl, OPTIONAL default=0.6

Maximum memory per node, in fraction of total memory, or MB if > 1. This should be set by default, but on some clusters (in particular the one at UCR) it is misreported and needs to be set by hand.

Nmesh=int, OPTIONAL default=0

Size of the FFT grid used to estimate displacements. Should be $>$ Ngrid. By default it is twice the mean inter-particle spacing, which should be fine for all purposes.

InvertPhase=int, OPTIONAL default=0

Flip phase for paired simulations. This implements the scheme in arXiv:1511.04090

NumPartPerFile=int, OPTIONAL default=1024 * 1024 * 128

NumWriters=int, OPTIONAL default=0

UsePeculiarVelocity=int, OPTIONAL default=0

Set up an run that uses Peculiar Velocity in IO. The default Gadget velocity units have an extra factor of $\sqrt{a}$. This is for generating FastPM compatible initial conditions.

5 Gadget Parameter File

5.1 Required Parameters

InitCondFile=str, REQUIRED default=

This sets the filename of the initial conditions to be read in at start-up. Note that the initial conditions file (or files) does not have to reside in the output directory. Initial conditions must be in the BigFile format.

OutputDir=str, REQUIRED default=

This is the pathname of the directory that holds all the output generated by the simulation (snapshot files, restart files, diagnostic files). Snapshots will be written in BigFile format.

OutputList=str, REQUIRED default=

Comma-separated list of output scale factors. eg: 0.1,0.2,0.5, 1.

TimeLimitCPU=dbl, REQUIRED default=0

This is the CPU-time limit for the current run (one submission to the computing queue) in seconds. This value should be matched to the corresponding limit of the queueing system. The run will automatically interrupt itself and write a snapshot at the end of a PM timestep when it detects that the next PM timestep is likely to take longer than the remaining time.

MetalReturnOn=int, REQUIRED default=0

Enables Metal Return

CoolingOn=int, REQUIRED default=0

Enables Cooling

SnapshotWithFOF=int, REQUIRED default=0

Enable built-in Friends-of-Friends halo finder when the snapshots are generated.

BlackHoleOn=int, REQUIRED default=1

Enable Black hole model and AGN feedback.

StarformationOn=int, REQUIRED default=0

Enables star formation.

WindOn=int, REQUIRED default=0

Enables stellar wind feedback from supernovae.

MassiveNuLinRespOn=int, REQUIRED default=0

Enables linear response massive neutrinos of 1209.0461. Make sure you enable radiation too.

DensityIndependentSphOn, REQUIRED default=1

Enable the Pressure-entropy formulation of SPH, from Hopkins et al 2010. This is more accurate, capturing shocks better, but is a little slower. Most full-physics simulations with hydro or star formation should enable this. Lyman-α forest simulations generally have it disabled, because the Lyman-α forest gas evolves adiabatically without shocks. Better shock capturing ability is thus not important. Enabling this also changes the SPH kernel to a quintic polynomial by setting DensityKernelType. Disabling this sets a cubic polynomial SPH kernel. For compatibility reasons this is implicitly disabled by setting DensityKernelType = cubic.

SnapshotWithFOF=int, REQUIRED default=0

Enable built-in Friends-of-Friends halo finder when the snapshots are generated.

5.2 `Cosmology Parameters`

TimeMax=dbl, OPTIONAL default=1.0

Scale factor to end run.

Omega0=dbl, REQUIRED default=0.2814

Total matter density, cdm + baryons + massive neutrinos at z=0. Cosmological matter density parameter in units of the critical density at z = 0. This must match the value in the initial conditions! It is required to try to reduce the number of simulations accidentally run with the wrong ICs.

OmegaBaryon=dbl, OPTIONAL default=0.0464

Baryon density (at z=0). Baryon density in units of the critical density at z = 0. Used rarely: to compute the self-shielding approximation, in the star formation module and in the timestepping for the gas particles. Default is to read from ICs.

OmegaLambda=dbl, OPTIONAL default=0.7186

Dark energy density at z=0. Cosmological vacuum energy density (cosmological constant) in units of the critical density at z = 0. The only purpose is to set the background expansion rate. As for GenIC, the code sets curvature with $\Omega_K = 1 - \Omega_0 - \Omega_\Lambda$. Default is to read from ICs.

HubbleParam=dbl, OPTIONAL default=0.697

Value of the Hubble constant in units of 100 km s^{-1} Mpc^{-1}. This is only used when conversions to cgs units are required (e.g. for radiative cooling physics). Note however that the value of HubbleParam is not needed (in fact, it does not enter the computation at all) in purely collisionless simulations. This is because of the definition of GADGET-2's system of units, which eliminates an explicit appearance of the value of the Hubble constant in the dynamical equations. Default is to read from ICs.

Omega_fld=dbl, OPTIONAL default=0

Energy density of a "dark energy fluid" at $z = 0$. This is an extra exotic species added to the background evolution.

w0_fld=dbl, OPTIONAL default=-1

Dark energy equation of state for the exotic fluid. We have $w = w_0 = (1 - a)w_a$.

wa_fld=dbl, OPTIONAL default=0

Dark energy evolution parameter.

Omega_ur=dbl, OPTIONAL default=0

Extra radiation density above the standard, following the CLASS convention. This could be used to model, e.g. a sterile neutrino. Only affects the background.

RadiationOn=int, OPTIONAL default=1

Include radiation density in the background evolution.

CMBTemperature=dbl, OPTIONAL default=2.7255

Present-day CMB temperature in Kelvin, default from Fixsen 2009; affects background if RadiationOn is set.

5.3 Cooling Model Parameters

The cooling module implemented in MP-Gadget is based on Katz Weinberg & Hernquist 1992 (KWH92), reimplemented from scratch in order to allow the use of other, more modern, cooling rates. We default to the rates used in Sherwood (Bolton et al 2016). There is a reionization model from Battaglia 2010, a self-shielding correction from Rahmati 2013 and a model for adjusting the heating rate to account for helium reionization. For more details see the comments in cooling_rates.c

CoolingOn=int, REQUIRED default=0

Enables cooling.

TreeCoolFile=str, OPTIONAL default=

Path to the Cooling Table, a text file storing the parameters of the uniform ultra-violet background. These are the ionizing emissivity and the heating rate for H, He and He+. This is the file traditionally called TREECOOL.

MetalCoolFile=str, OPTIONAL default=

Path to the Metal Cooling Table, which is a BigFile. This stores cooling rates as a function of metallicity from various metal ions. An empty string (default) disables metal cooling. The Metal cooling table used is the one from Illustris, see Vogelsberger 2013.

UVFluctuationFile=str, OPTIONAL default=

Path to the UVFluctation Table. This stores a reionization redshift as a function of spatial position, and is also a BigFile. Empty string disables. The reionization model is that from Battaglia trac 2010.

UVRedshiftThreshold=dbl, OPTIONAL default=-1

Maximal reionization redshift. Positive values induce a zero UVB at redshifts higher than UVRedshiftThreshold.

CoolingRates=dbl, OPTIONAL default=Sherwood

Which cooling rate table to use. Options are KWH92 (old gadget default), Enzo2Nyx (rates used in Nyx (Lukic 2015) and in Enzo 2) and Sherwood (new default). Sherwood fixes a bug in the Cen 92 rates, but are otherwise the same as KWH.

RecombRates=dbl, OPTIONAL default=Verner96

Which recombination rate table to use. Options are Cen92 (old gadget default), Verner96 : Verner Ferland 1996, more accurate rates. Voronov 97 is used for collisional ionizations. (new default) B06 (Badnell 2006 rates, current cloudy defaults. Very similar to V96).

PhotoIonizeFactor=dbl, OPTIONAL default=1.0

Optional factor to scale the UVB stored in TREECOOL by.

PhotoIonizeFactor=int, OPTIONAL default=1

Flag to enable Photoionization.

SelfShieldingOn=int, OPTIONAL default=1

Enable a correction in the cooling table for self-shielding of hydrogen at high densities, following Rahmati et al 2013.

HeliumHeatOn=int, OPTIONAL default=0

Change photo-heating rate to model helium reionisation on underdense gas. Enable model for helium reionisation which adds extra photo-heating to under-dense gas. Extra heating has the form: $H = A * (\rho/\rho_{c(z=0)})^{Exp}$ but is density-independent when $\rho/\rho_c >$ Thresh.

HeliumHeatThresh=dbl, OPTIONAL default=10

Overdensity above which heating is density-independent.

HeliumHeatAmp=dbl, OPTIONAL default=1

Density-independent heat boost. Changes mean temperature.

HeliumHeatExp=dbl, OPTIONAL default=0

Density dependent heat boost (exponent). Changes gamma.

5.4 Star Formation Model Parameters

We implement the two-phase star forming medium model of Springel and Hernquist 2003 (SH03: `https://arxiv.org/abs/astro-ph/0206393`). The meaning of all the parameters of this model are explained in that paper. We also implement some other star formation criteria, explained below.

StarformationCriterion=enum, OPTIONAL default=density

This controls the star formation criteria used. The default is "density", which corresponds to pure SH03. Other options are: 1) "h2". Form stars depending on the molecular hydrogen gas fraction. Needs the SPH_GRAD_RHO compile-time flag enabled. 2) "selfgravity". Form stars only when the gas is self-gravitating. From Phil Hopkins. 3) "convergent" Modify self-gravitating star formation to form stars only when the gas flow is convergent. 4) "continous" Modify self-gravitating star formation to smooth the star formation threshold.

QuickLymanAlphaProbability=dbl, OPTIONAL default=0

Implements a simplified star formation criterion in which gas is turned directly into stars above the critical overdensity. This is commonly used for simulations for the Lyman-alpha forest to speed up the simulations by not following dense gas. Setting it to 1 turns dense gas directly into stars.

CritOverDensity=dbl, OPTIONAL default=57.7

Threshold over-density (in units of the critical density) for gas to be star forming. For QuickLymanAlpha models this should be 1000.

CritPhysDensity=dbl, OPTIONAL default=0

critical physical density for star formation in hydrogen number density (in cm^{-3}). An alternative way of specifying the

FactorSN=dbl, OPTIONAL default=0.1

Parameter of the SH03 model.

FactorEVP=dbl, OPTIONAL default=1000

Parameter of the SH03 model.

TempSupernova=dbl, OPTIONAL default=1e8

Temperature of the supernovae remnants in K. Parameter of the SH03 model.

TempClouds=dbl, OPTIONAL default=1000

Temperature of the cold star forming clouds in K. Parameter of the SH03 model.

MaxSfrTimescale=dbl, OPTIONAL default=1.5

Maximum star formation time in units of the density threshold. Parameter of the SH03 model.

5.5 `Wind (Stellar Feedback) Model Parameters`

We implement stellar feedback using a variety of literature prescriptions. Our feedback model is close to that used in Illustris, and includes variable strength winds with a velocity proportional to the circular velocity of the halo. See Vogelsberger 2013 (`https://arxiv.org/abs/1305.2913`), and references, especially Oppenheimer 2010. The wind model is built on top of the SH03 star formation model and is effective at suppressing low mass star formation. Wind particles are spawned by the star forming gas and are decoupled from the hydrodynamics of the gas for a short time, so that the energy does not simply cool away.

WindModel=enum, OPTIONAL default=subgrid,decouple,fixedefficiency

Possible options for the stellar feedback model. Options are: 1) "subgrid", the original model of SH03, in which winds are included by returning energy to the star forming regions. This cools away and is ineffective. 2) "decouple", specifies that wind particles are created and are temporarily decoupled from the gas dynamics. 3) "halo" Wind speeds depend on the halo circular velocity. 4) "fixedefficiency" Winds have a fixed efficiency and thus fixed wind speed. 5) "isotropic" If enabled, wind direction is random and isotropic. If disabled it goes in the direction of the local gravitational acceleration.

Other options are specific combinations of these that were used in papers. The default is OFJT10, which corresponds to decoupled winds with velocities proportional to halo circular velocities.

WindEfficiency=dbl, OPTIONAL default=2.0

Fraction of the star forming mass that forms a wind.

WindEnergyFraction=dbl, OPTIONAL default=1.0

Fraction of the available energy that goes into winds.

WindSigma0=dbl, OPTIONAL default=353

Reference halo circular velocity at which to evaluate wind speed. Needs ofjt10 wind model.

WindSpeedFactor=dbl, OPTIONAL default=3.7

Factor connecting wind speed to halo circular velocity. ofjt10 wind model.

WindFreeTravelLength=dbl, OPTIONAL default=20

Expected decoupling distance for the wind in physical km.

WindFreeTravelDensFac=dbl, OPTIONAL default=0.

If the density of the wind particle drops below this factor of the star formation density threshold, the gas will recouple.

5.6 Black Holes

We implement black hole feedback and accretion roughly following di Matteo et al. The accretion is Eddington limited and the feedback is purely thermal.

BlackHoleOn=int, REQUIRED default=1

Enable Blackhole

BlackHoleAccretionFactor=dbl, OPTIONAL default=100

BH accretion rate as a fraction of the Bondi accretion rate.

BlackHoleEddingtonFactor=dbl, OPTIONAL default=3

Maximum Black hole accretion as a function of Eddington.

SeedBlackHoleMass=dbl, OPTIONAL default=5e-5

Minimal black hole seed mass in internal mass units.

BlackHoleNgbFactor=dbl, OPTIONAL default=2

Factor by which to increase the number of neighbours for a black hole.

BlackHoleMaxAccretionRadius=dbl, OPTIONAL default=99999

Maximum neighbour search radius for black holes.

BlackHoleFeedbackFactor=dbl, OPTIONAL default=0.05

Fraction of the black hole luminosity to turn into thermal energy

BlackHoleFeedbackRadius=dbl, OPTIONAL default=0

Maximum comoving (if ¿ 0) radius at which the black hole feedback energy is deposited.

BlackHoleFeedbackRadiusMaxPhys=dbl, OPTIONAL default=0

Maximum physical (if ¿ 0) radius at which the black hole feedback energy is deposited.

BlackHoleFeedbackMethod=enum, OPTIONAL default=spline, mass

AGN feedback model to use.

TimeBetweenSeedingSearch=dbl, OPTIONAL default=1e5

Time Between BH Seeding Attempts: default to a a large value, meaning never.

5.7 `Massive Neutrino Model`

We have an analytic model for effect of massive neutrinos on structure formation, following `https://arxiv.org/abs/1209.0461`. The massive neutrinos are treated perturbatively. This works extremely well for currently viable neutrino masses and for the total matter density. In the unlikely event that you care about the density of very massive neutrinos there is a hybrid neutrino model following `https://arxiv.org/abs/1803.09854`.

MassiveNuLinRespOn=int, REQUIRED default=0

Enables linear response massive neutrinos of 1209.0461. Make sure you enable radiation too.

HybridNeutrinosOn=int, OPTIONAL default=0

Enables hybrid massive neutrinos, where some density is followed analytically, and some with particles. Requires MassivenuLinRespOn.

MNue=dbl, OPTIONAL default=0

First neutrino mass in eV.

Mnum=dbl, OPTIONAL default=0

Second neutrino mass in eV.

Mnut=dbl, OPTIONAL default=0

Third neutrino mass in eV.

Vcrit=dbl, OPTIONAL default=500

For hybrid neutrinos: Critical velocity (in km/s) in the Fermi-Dirac distribution below which the neutrinos are particles in the ICs.

NuPartTime=dbl, OPTIONAL default=0.3333333

Scale factor at which to turn on hybrid neutrino particles. Before this scale factor they are tracers.

FastParticleType=int, OPTIONAL default=2

Particles of this type will not decrease the timestep. Default neutrinos.

5.8 `Output Parameters`

Output files are all stored in the Bigfile format. Each particle type is stored separately. Particle type 0 is gas, 1 is CDM, 2 is neutrinos, 4 are stars and 5 is black holes. Type 3 is presently unused. BigFiles can be read with the python bigfile package.

OutputDir=str, REQUIRED default=

This is the pathname of the directory that holds all the output generated by the simulation (snapshot files, restart files, diagnostic files). Snapshots will be written in BigFile format.

OutputList=str, REQUIRED default=

Comma-separated list of output scale factors. eg: 0.1,0.2,0.5, 1.

SnapshotFileBase=str, OPTIONAL default="PART"

From this string, the name of the snapshot files is derived by adding an underscore, and the number of the snapshot in a 3-digits format.

FOFFileBase=str, OPTIONAL default="PIG"

Base name of the fof files, _%03d will be appended to the name.

EnergyFile=str, OPTIONAL default="energy.txt"

This is the name of a file where information about global energy statistics of the simulation is stored. Checking energy conservation can sometimes be a good proxy of time integration accuracy, but it is not a particularly stringent test. Because computing the gravitational energy requires extra CPU time, it has to be explicitly enabled with the COMPUTE POTENTIAL ENERGY makefile option. Note also that an accurate computation of the peculiar gravitational binding energy at high redshift is quite subtle, so that checking for energy conservation is not a good test to evaluate the accuracy of cosmological integrations.

OutputEnergyDebug=int, OPTIONAL default=0

Should we output energy statistics to energy.txt?

CpuFile=str, OPTIONAL default="cpu.txt"

This is a log-file that keeps track of the cumulative cpu-consumption of various parts of the code. At each timestep, one line is added to this file. The meaning of the individual entries will be described later on. A detailed analysis of them allows an assessment of the code performance, and an identification of potential bottlenecks.

OutputPotential=int, OPTIONAL default=1

Save the potential in snapshots.

AutoSnapshotTime=dbl, OPTIONAL default=0

Seconds after which to automatically generate a snapshot if nothing is output.

BytesPerFile=int, OPTIONAL default=1024 * 1024 * 1024

number of bytes per file.

NumWriters=enum, OPTIONAL default=NTask

Max number of concurrent writer processes. 0 implies Number of Tasks.

MinNumWriters=int, OPTIONAL default=1

Min number of concurrent writer processes. We increase number of Files to avoid too few writers.

WritersPerFile=int, OPTIONAL default=8

Number of Writer groups assigned to a file; total number of writers is capped by NumWriters.

EnableAggregatedIO=int, OPTIONAL default=0

Use the Aggregated IO policy for small data set (Experimental).

AggregatedIOThreshold=int, OPTIONAL default=1024 * 1024 * 256

Max number of bytes on a writer before reverting to throttled IO.

5.9 Other Parameters

DensityIndependentSphOn, REQUIRED default=1

Enable the Pressure-entropy formulation of SPH, from Hopkins et al 2010. This is more accurate, capturing shocks better, but is a little slower. Most full-physics simulations with hydro or star formation should enable this. Lyman-α forest simulations generally have it disabled, because the Lyman-α forest gas evolves adiabatically without shocks. Better shock capturing ability is thus not important. Enabling this also changes the SPH kernel to a quintic polynomial by setting DensityKernelType. Disabling this sets a cubic polynomial SPH kernel. For

compatibility reasons this is implicitly disabled by setting DensityKernelType = cubic.

HydroOn=int, OPTIONAL default=1

Enables hydro force.

DensityOn=int, OPTIONAL default=1

Enables SPH density computation.

TreeGravOn=int, OPTIONAL default=1

Enables tree gravity.

DensityKernelType=enum, OPTIONAL default=quintic

Sets the SPH kernel in use. Common choices are quintic (the modern quintic kernel) or cubic (the old school kernel). For compatibility reasons setting DensityKernelType = cubic implicitly sets DensityIndependentSphOn = 0. Setting DensityIndependentSphOn = 0 implies DensityKernelType = cubic and DensityIndependentSphOn = 1 implies DensityKernelType = quintic.

PartAllocFactor=dbl, OPTIONAL default=1.5

Each processor allocates space for PartAllocFactor times the average number of particles per MPI rank. This number needs to be larger than 1 to allow the simulation to achieve a good work-load balance. Ideally the code tries to move particles between processors so that each processor takes an equal time to process its particle load. If the code is unable to do this while remaining within the boundary for particle balancing, it will instead balance for equal numbers of particles on each processor, which will generally be a little slower. A reasonable value is 2.0 if in pure MPI mode and 1.5 if in threaded mode. Must always be > 1.0.

Nmesh=int, OPTIONAL default=-1

Size of the PM mesh, which sets the minimal resolution of the long-range force. Because the PM force is more computationally efficient than the Tree force to compute, larger values are generally faster, but use more memory. By default it is the same as the grid on which the

ICs are computed: twice the cube root of the number of CDM particles. Note that this is a numerical parameter: gravitational forces are computed using the tree down to the softening scale. The factor 2 is chosen because this is the largest integer which does not dominate the memory usage. It was found that increasing the factor to 3 increases memory use by 10% and computation time by another 10%. The PM time became larger by a factor of $3^3/2^3$, reflecting the increase in grid size, but the tree force because faster by a smaller factor, partially compensating. Increasing Nmesh did markedly reduce transients from the particle grid, because the particle mesh force was no longer a multiple of the particle spacing.

MaxMemSizePerNode=dbl, OPTIONAL default=0.6

Preallocate this much memory per computing node/ host, in MB. Defaults to 60% of total available memory per node. Passing < 1 allocates a fraction of total available memory per node.

DomainOverDecompositionFactor=int, OPTIONAL default=4

Create on average this number of topNodes per MPI rank. Higher numbers improve the load balancing but make domain more expensive.

DomainUseGlobalSorting=int, OPTIONAL default=1

Determining the initial refinement of chunks globally. Enabling this produces better domains at costs of slowing down the domain decomposition.

ErrTolIntAccuracy=dbl, OPTIONAL default=0.02

This dimensionless parameter controls the accuracy of the timestep criterion selected by TypeOfTimestepCriterion.

ErrTolForceAcc=dbl, OPTIONAL default=0.005

Force accuracy required from tree. Controls tree opening criteria. Lower values are more accurate. This controls the accuracy of the relative cell-opening criterion (if enabled), where a cell is opened if $Ml^2 > \alpha|\mathbf{a}_{old}|r^4$. Here, α is set by ErrTolForceAcc, M is the mass inside a node, l is its cell side-length, and aold is the magnitude of the total acceleration of the

particle. The latter is usually estimated based on the gravitational force on the particle in the previous timestep. Note that independent of this relative criterion, the code will always open nodes if the point of reference lies within the geometric boundaries of the cubical cell (enlarged by an additional 10% on each side). This protects against the rare occurrence of very large force errors, which would otherwise be possible.

Asmth=dbl, OPTIONAL default=1.25

The scale of the short-range/long-range force split in units of FFT-mesh cells. Gadget-2 paper says larger values may be more accurate.

MinGasHsmlFractional=dbl, OPTIONAL default=0

Minimal gas Hsml as a fraction of gravity softening. This parameter sets the minimum allowed SPH smoothing length in units of the gravitational softening length of the gas particles. The smoothing length will be prevented from falling below this value. When this bound is actually reached, the number of smoothing neighbours will instead be increased above DesNumNgb, even if this means that the upper limit DesNumNgb+MaxNumNgbDeviation is exceeded. The parameter MinGasHsmlFractional can be zero, allowing the smoothing radius to vary without bounds, which is usually adequate.

MaxGasVel=dbl, OPTIONAL default=3e5

Maximum gas velocity in km/s. Defaults to speed of light.

MaxSizeTimestep=dbl, OPTIONAL default=0.1

This parameter sets the maximum size of the PM (and thus overall code) timestep. This practically reduces timesteps at very high redshifts when accelerations and displacements are small. Usually, a few percent up to tens of percent of the dynamical time of the system gives sufficient accuracy. This is specified as $\Delta \ln a$. The new definition is equivalent to specifying the maximum allowed timestep for cosmological simulations as a fraction of the current Hubble time. The default is 0.1. The effect of reducing it to 0.05 was measured to be a scale-independent increase in the large scale power of $1 + 2 \times 10^{-3}$ or 0.2%.

MinSizeTimestep=dbl, OPTIONAL default=0

If a particle requests a timestep smaller than the specified value, the simulation is terminated with a warning message. This can be used to prevent simulations to continue when the timestep has dropped to a very small value, because such behaviour typically indicates a problem of some sort. However, if NOSTOP WHEN BELOW MINTIMESTEP is set in the Makefile, the timestep is forced instead to be at least MinSizeTimestep, even if this means that the other timestep criteria are ignored. This option should hence be used with care. It can be useful however to force equal and constant timesteps for all particles.

ForceEqualTimesteps=int, OPTIONAL default=

Force all timesteps to be the same, the smallest required.

MaxRMSDisplacementFac=dbl, OPTIONAL default=0.2

This sets the timestep of the long-range (PM) force. The code computes the average velocity magnitude $|\tilde{v}|^2$ for all particles, and restricts the PM timestep such that $|v|^2 d\log a < a^2 H(a)$MaxRMSDisplacementFac $\times$ MIN$((d, r_{\rm smth})$. Here $r_{\rm smth}$ is the smoothing scale of the short-range/long-range force split (1.25$\times$ Box/Nmesh). d is the "radius" of a single particle, specified by $(M_{\rm part}/\bar{\rho_{\rm part}})^{1/3}$. Where particles have variable mass the smallest mass is used. The default is 0.2. It was checked that the effect of decreasing to 0.1 is a scale-independent increase in the large scale power of $1 + 5 \times 10^{-4}$, or 0.05%.

ArtBulkViscConst=dbl, OPTIONAL default=0.75

This sets the value of the artificial viscosity parameter αvisc used by GADGET-2. See code paper for details.

CourantFac=dbl, OPTIONAL default=0.15

This sets the value of the Courant parameter used in the determination of the hydrodynamical timestep of SPH particles. Note that GADGET-2's definition of the SPH smoothing length differs by a factor of 2 from that found in large parts of the SPH literature. As a consequence, comparable settings of CourantFac are a factor of 2 smaller in GADGET-2 when compared with codes using the different convention.

DensityResolutionEta=dbl, OPTIONAL default=1.0

Resolution eta factor (See Price 2008) 1 = 33 for Cubic Spline

DensityContrastLimit=dbl, OPTIONAL default=100

Max contrast for hydro force calculation.

MaxNumNgbDeviation=dbl, OPTIONAL default=2

HydroCostFactor=dbl, OPTIONAL default=1

Cost factor of hydro calculation, default to 1.

UVRedshiftThreshold=dbl, OPTIONAL default=-1.0

Earliest Redshift that UV background is enabled. This modulates UVFluctuation and TreeCool globally. Default -1.0 means no modulation.

GravitySoftening=dbl, OPTIONAL default=1./30.

Softening for collisionless particles; units of mean separation of DM. ForceSoftening is 2.8 times this.

GravitySofteningGas=dbl, OPTIONAL default=1./30.

Softening for collisional particles (Gas); units of mean separation of DM; 0 to use Hsml of last step.

TopNodeAllocFactor=dbl, OPTIONAL default=0.5

InitGasTemp=dbl, OPTIONAL default=-1

Initial gas temperature. By default set to CMB temperature at starting redshift. This sets the initial gas temperature in Kelvin when initial conditions are read. If the temperature is below 104 K, a mean molecular weight corresponding to neutral gas of primordial abundance is assumed, otherwise complete ionisation is assumed. However, the gas temperature is only set to a certain temperature if InitGasTemp > 0 and if at the same time the temperature of

the gas particles in the initial conditions file is zero, otherwise the initial gas temperature is left at the value stored in the IC file.

MinGasTemp=dbl, OPTIONAL default=5

The minimum allowed gas temperature in Kelvin (this is converted by the code to a minimum thermal energy per unit mass assuming the mean molecular weight of neutral gas). This may be set to zero in principle, but often it is desirable to prevent the gas from becoming too cold, e.g. for resolution reasons or because of lower limits in the implemented cooling function.

FOFSaveParticles=int, OPTIONAL default=1

Save lists of particles in each halo in the FOF catalog.

FOFHaloLinkingLength=dbl, OPTIONAL default=0.2

Linking length for Friends of Friends halos.

FOFHaloMinLength=int, OPTIONAL default=32

Minimum number of particles within each FoF halo.

MinFoFMassForNewSeed=dbl, OPTIONAL default=5e2

Minimal halo mass within which a black hole will be seeded (in internal mass units).

RandomSeed=int, OPTIONAL default=42

Random number generator initial seed. Used to form stars.

6 Example Simulations

The code repository contains a number of example parameter files for specific use cases. These examples are designed to be possible to run on a reasonably powerful laptop or desktop, and not to require a cluster. For any useful production simulation you would want to substantially increase the particle number. Most example simulations use 128^3 or 256^3 particles. A production run uses $512^3 - 1024^3$ particles. Current examples are described below.

6.1 dm-only

A small simulation which includes only collisionless dark matter. This can be used for seeing how structure formation progresses, and for counting halos. The box size used is large enough to produce an accurate cosmology even at $z = 0$.

6.2 fastpm-compat

A small simulation to generate output compatible with FastPM. Principally used for FastPM development.

6.3 glass

An initial conditions only example to show how to generate a Lagrangian glass file and to be a test case.

6.4 hydro

A small "full-physics" simulation. This example includes star formation, stellar feedback and black holes. It has a very small box so will not produce a correct cosmology, but you can use it to see how the star formation, wind and black hole parameters affect the simulation.

6.5 linear_growth

A simulation with the initial power suppressed by a factor of 100, to test structure growth in the fully linear regime. Star formation and gas are included using the simplified Lyman-alpha forest prescription.

6.6 lya

A example with the preferred setup for simulating the Lyman-α forest. This uses a simplified star formation criterion, gas, and a resolution of order 100 kpc, sufficient for the Lyman-α forest clouds. Cosmologically useful runs need more particles and a larger box size.

6.7 neutrinos

A small dark matter only cosmological simulation which enables the perturbative linear response neutrino model. Designed to show the effect of massive neutrinos on the formation of structure.

6.8 small

A very low particle count (32^3) small box full-physics simulation. Designed to form stars and black holes quickly.

6.9 travis

The test simulation that runs whenever anyone submits a pull request. It is designed to exercise as much of the code as possible as quickly as possible.

7 Snapshot Format

The snapshot output

8 Basic Cluster Submission Script Usage

8.1 *mpirun*

mpirun typically works like this: mpirun -np <number of processes> <program name and arguments>

The options for mpirun must come before the program you want to run and must be spelled out completely (no abbreviations). Unrecognized options will be silently ignored: mpirun [mpirun options] <progname> [program options]

If you are simply looking for how to run an MPI application, you probably want to use a command line of the following form: mpirun [-np X] [–hostfile <filename>] <program>

This will run X copies of <program> in your current run-time environment (if running under a supported resource manager, Open MPI's mpirun will usually automatically use the corresponding resource manager process starter, as opposed to, for example, rsh or ssh, which require the use of

a hostfile, or will default to running all X copies on the localhost), scheduling (by default) in a round-robin fashion by CPU slot. See the rest of this page for more details.

Please note that mpirun automatically binds processes as of the start of the v1.8 series. Three binding patterns are used in the absence of any further directives:

Bind to core: when the number of processes is <= 2 Bind to socket: when the number of processes is > 2 Bind to none: when oversubscribed

If your application uses threads, then you probably want to ensure that you are either not bound at all (by specifying –bind-to none), or bound to multiple cores using an appropriate binding level or specific number of processing elements per application process.

8.2 *Slurm*

8.2.1 `sbatch`

sbatch submits a batch script to SLURM. The batch script may be given to sbatch through a file name on the command line, or if no file name is specified, sbatch will read in a script from standard input. The batch script may contain options preceded with "#SBATCH" before any executable commands in the script. After the first line of the batch script, which typically identifies the shell to be used, sbatch will stop processing options at the first line which does NOT begin with "#SBATCH".

sbatch exits immediately after the script is successfully transferred to the SLURM controller and assigned a SLURM job ID. The batch script is not necessarily granted resources immediately, it may sit in the queue of pending jobs for some time before its required resources become available.

By default both standard output and standard error are directed to a file of the name "slurm-%j.out", where the "%j" is replaced with the job allocation number. The file will be generated on the first node of the job allocation. Other than the batch script itself, Slurm does no movement of user files.

When the job allocation is finally granted for the batch script, SLURM runs a single copy of the batch script on the first node in the set of allocated nodes.

A Installing on Mac

- Install and/or update Xcode (most easily done via the App Store), and install homebrew.
- Install GSL with homebrew: "brew install gsl"
- Install wget with homebrew: "brew install wget"
- Install sed with homebrew: "brew install gnu-sed" and then enable it with "export PATH="/usr/local/opt/gn sed/libexec/gnubin:$PATH"
- Install Open-MPI using "brew install open-mpi" and set the compiler for openmpi with "export OMPI_CXX=g++-9" "export OMPI_CC=gcc-9" (replace gcc-9 with whatever the current version of gcc is, check with gcc –version or look in /usr/local/bin).
- For "The bottle needs the Xcode CLT to be installed" run "xcode-select –install"
- Check that it is using gcc compiler instead of clang via "mpicc –show-me"
- Copy platform-options/Options.mk.macos to Options.mk
- Type 'make'

The two binaries, MP-Gadget and MP-Genic, will be in the folders gadget and genic respectively.