Chapter 2

Simulation Methodology

2.1 The generation of cosmological initial conditions

Creating a set of initial conditions for simulations of cosmological structure formation is, on the surface, a very straightforward task. One specifies a background cosmological model, typically described as a spatially flat or open Robertson-Walker spacetime. Following that, perturbations are imposed upon this background assuming a triply periodic, finite simulation volume. In reality, numerous approximations must be made which should be carefully considered, and are discussed in some detail below.

The specification of background cosmology requires several inputs: The amount and nature of dark matter, the Hubble parameter H_0, and possibly the amount of baryonic matter and cosmological constant in the universe. Most of these quantities are typically specified in terms of the standard cosmological parameters: Ω_m, Ω_{dm} and Ω_b, which correspond to the total overall mass density, dark matter density and baryon density at the present epoch as a fraction of the critical density, $\rho_c \equiv 3H_0^2/8\pi G$. The cosmological constant is specified as Ω_Λ, which is the vacuum energy density at the current epoch as a fraction of the critical energy density, which is simply $\rho_c c^2$. The perturbations of dark matter and baryons are specified as a power spectrum which has one or more parameters. At the very least, a power spectrum has an index n which specifies the shape of the primordial power spectrum.

At the epoch of baryon-photon decoupling (z $\sim$ 1100), small-amplitude ("linear") fluctuations in density are already present in all of the components that contribute to the energy density of the universe (such as baryons, dark matter, photons, and neutrinos). The statistical nature of these fluctuations depends on their origin. There are two general classes of early universe models that are considered to provide reasonable mechanisms for perturbations: topological defects [119] and inflation [120]. Inflation predicts fluctuations that obey Gaussian statistics and defect models predict non-Gaussian fluctuations.

Gaussian fluctuations are simple since they are specified completely by a single function, the power spectrum P(k). In Gaussian models the perturbations are set down almost immediately after the Big Bang (during the inflationary epoch in the canonical model of inflationary cosmology, roughly 10^{-35} seconds after the Big Bang) and evolve according to the linearized Fokker-Planck equation. In real space, the probability distribution of density fluctuations is a multidimensional Gaussian, and it is very easy to sample a Gaussian random field by sampling its Fourier components on a Cartesian lattice, which is the technique that will be discussed here. For more information on other methods, see Bertschinger's review on simulations of cosmological structure formation [121].

Non-Gaussian models are much more complicated to model. Not only do they require more initial information than a simple power spectrum, they also are more costly in a computational sense. Typically, topological defects induce matter density fluctuations from the time of their creation in the early universe to the present day, and the dynamics of their formation and evolution are relativistic and nonlinear. For more information on creating initial conditions of topological defects, see Bertschinger [121] or Durrer et. al. [119].

A simple test of the Gaussianity of the primordial density perturbations can be made by examining the power spectrum of the cosmic microwave background. The power spectrum of temperature fluctuations in the CMB was imposed at a very early time ($z \sim 1100$, approximately 300,000 years after the Big Bang) when all perturbations were linear (and are at most $\delta\rho/\bar{\rho} \sim 10^{-5}$, where $\bar{\rho}$ is the mean density and $\delta\rho \equiv |\rho - \bar{\rho}|$), significantly before the epoch of cosmological structure formation (which is when density perturbations become nonlinear and gravitationally self-bound). Recent observations of the CMB using the Wilkinson Microwave Anisotropy Probe (WMAP) satellite have examined the CMB for signs of non-Gaussianity and have presented upper limits on the amplitude of non-Gaussian primordial fluctuations using two separate tests [122]. These observations show to a high level of confidence that the WMAP data is consistent with Gaussian primordial fluctuations.

Due to the preponderance of evidence in favor of a Gaussian perturbation spectrum, all simulations described in this thesis will assume this spectrum. Non-Gaussian initial conditions (i.e. topological defect models) will most likely produce significantly different results for structure formation on the scales of interest.

2.1.1 Creating Gaussian random fields

The creation of cosmological initial conditions using a Gaussian random field is relatively straightforward. Given a power spectrum P(k), the linear density fluctuation field is calculated at some initial time when density perturbations are highly linear ($\delta\rho/\bar{\rho} \ll 1$; a typical choice of starting redshift is z $\sim$ 100 for high-resolution cosmological simulations). From this, dark matter particle positions and velocities are determined, along with baryon velocity fields, as described below.

The Linear Density Fluctuation Field

The first step towards creating a Gaussian density field is to specify a power spectrum, which is a function that relates amplitudes of density perturbations with wavelength λ to their wavenumber k, where $k \equiv 2\pi/\lambda$. The power spectrum of the fractional density fluctuations at the redshift z = z$_{eq}$ when the energy density in matter is equal to that in radiation, can be related to the primordial power spectrum by $P(k, z_{eq}) \sim T^2(k)P_p(k)$, where T(k) is the matter transfer function as a function of wave number and describes the processing of the initial density perturbations during the radiation dominated era (see Padmanabhan [123]) and P$_p$(k) is the primordial matter power spectrum, which typically has a power law form, i.e., P$_p$(k) $\sim$ k^n, where n is the index of the primordial power spectrum. This index is equal to unity for Harrison-Zel'Dovich scale-invariant spectra, a typical model [124, 125]. The power spectrum at any redshift z in the matter dominated era may then be written in the form

$$\frac{k^3}{2\pi^2}P(k,z) = \left(\frac{ck}{H_0}\right)^{3+n} \delta_H^2 T^2(k) D_g^2(z)/D_g^2(0), \tag{2.1}$$

where the D$_g$'s are the linear grown factor for perturbations, which is defined in Peebles [126] and many other places. A closed-form fitting function (much more appropriate for computation) is given in Eisenstein & Hu [194]. δ_H is a constant describing the amplitude of density fluctuation, which can be provided from observations of the CMB or from large scale structure, or can be normalized by comparing to, e.g., σ_8, which is the RMS amplitude of the mass fluctuations in the universe when smoothed using a top-hat function with characteristic radius of 8 h^{-1} Mpc.

Once P(k) has been determined, we then proceed to calculate δ_k, namely, the density fluctuations in k-space. To simplify matters, we choose a cubic three-dimensional Cartesian grid with N grid points per dimension, though in principle this set of operations can be calculated for any rectangular solid lattice with some small additional complications. Each of the grid points has a unique identifier (n$_x$,n$_y$,n$_z$) associated with its location along the (x,y,z) axis. We sample the power spectrum P(k) discretely at each grid location (n$_x$,n$_y$,n$_z$), obtaining k in this manner:

$$k^2 = (n_x^2 + n_y^2 + n_z^2)dk^2 \tag{2.2}$$

with dk = 2π/L$_{box}$, and L$_{box}$ is the size of the simulation box in h^{-1} Mpc. It is important to note that the method of sampling P(k) at discrete points, while convenient when doing simulations on a Cartesian grid, assumes that P(k) is a smoothly and slowly varying function of k on the physical scales relevant to the simulation (namely, on scales encompassing the simulation volume and the size of the grid lattice). If this is not true, large errors will be introduced unless P(k) is sampled in a different way. We also assume that the simulation volume is a finite size and assume periodicity - i.e., an object that travels out of one side of the box re-enters it on the other side. This assumption implicitly states that the simulation volume is an "average" patch of the universe.

The k-space fluctuation δ_k is a complex value with random phase and amplitude, where the distribution of amplitudes fit a Gaussian distribution with a mean of P(k). One method to calculate it is to generate a phase angle θ, which is randomly selected in a uniform manner in the interval $[0, 2\pi]$, and an amplitude A such that

$$A = \sqrt{-log(R) * P(k)} \tag{2.3}$$

where R is randomly selected in a uniform manner in the interval (0,1). δ_k is then

$$\delta_k = Ae^{i\theta} \tag{2.4}$$

We then perform a Fourier transform on the grid of δ_k values, giving δ_x, the relative density fluctuation at each spatial grid point in the simulation volume. The actual physical (meaning, in position space rather than k-space) dark matter density is then

$$\rho_{DM}(\mathbf{x}) = (1 + \delta_x(\mathbf{x}))\overline{\rho}_{DM} \tag{2.5}$$

where $\overline{\rho}_{DM}$ is the mean dark matter density in the simulation. The perturbations in the dark matter and baryon densities are assumed to be coupled, which is a reasonable assumption in the linear regime, so the baryon density at any position is

$$\rho_b(\mathbf{x}) = \frac{\Omega_b}{\Omega_{DM}}\rho_{DM}(\mathbf{x}) \tag{2.6}$$

where Ω_b and Ω_{DM} are the ratios of the present-day mean baryon and dark matter densities to the critical density of the universe, i.e., $\Omega_b = \rho_b/\rho_c$. The assumption of complete baryon-dark matter coupling, though reasonable, is not strictly true at the redshifts at which cosmological simulations are typically initialized due to the dissipative nature of the baryons. Therefore, it is more accurate to use separate transfer functions for the dark matter and baryon components, though in practice this only makes a small difference for a certain class of simulations. For this reason we typically ignore it when generating our initial conditions.

Position and Velocity Fields

In cosmological simulations, dark matter is usually represented by discrete particles rather than a uniform grid of densities. In that case, these particles must be perturbed so as to reflect the density fluctuations described in Section 2.1.1. The standard approach for the dark matter is to displace equal-mass particles from a uniform Cartesian lattice using the Zel'Dovich approximation [127]:

$$\mathbf{x} = \mathbf{q} + D(a)\psi(\mathbf{q}) \tag{2.7}$$

and

$$\mathbf{v} = a\frac{dD}{dt}\psi = aHfD\psi, \tag{2.8}$$

where $\mathbf{q}$ labels the unperturbed lattice position, D(a) is the growth factor of the linear growing mode, H is the Hubble parameter, a is the cosmological scale factor, and $f = dlnD/dlna$ is its logarithmic growth rate [126]. The irrotational (curl-free) displacement field ψ is computed by solving the linearized continuity equation,

$$\bigtriangledown \cdot \psi = -\frac{\delta_x}{D(t)}, \tag{2.9}$$

Since the equation is linearized, ψ can be found by taking the Fourier transform of $-i\delta_k\hat{k}/kD(k)$. This approximation is then also used to calculate the baryon velocities (using the baryon perturbation field if a different transfer function is used, or a scaling factor for δ_x and δ_k otherwise).

2.1.2 Creating initial conditions for high dynamical range simulations

Compromises must be made when one is performing cosmological simulations of large-scale structure formation. In order to get a statistically accurate representation of large scale structure, one must use as large of a spatial volume as possible. The dark matter particle mass and ultimate spatial resolution of the simulation must be chosen based on the structures of interest, so that any given object (e.g. galaxy, galaxy cluster, etc.) is resolved by a large number of dark matter particles and by many grid cells. Unfortunately, given finite computational resources (in terms of both memory and processing power) these two criteria are often in direct competition. For example, in the simulations discussed later in this thesis, it is important to choose a box that is of large enough size to contain at least one $5 \times 10^5 M_\odot$ dark matter halo, which gives us a minimum box size that is approximately 0.3 h^{-1} Mpc on a side. We also need to resolve the region around this halo with dark matter particles that are small compared to the halo mass and the final protostar, typically on the order of a few solar masses for a calculation of this type. Given that $M_{dm} = \Omega_{cdm}\rho_c(L_{box}/N_{grid})^3 M_\odot/h$ (where L_{box} is the box size in Mpc/h and N_{grid} is the number of grid cells along one edge of the box), this results in a volume with approximately 1024^3 grid cells and dark matter particles. This is computationally wasteful, since we only need to resolve a few percent of the box with high spatial and mass resolution. It is important to resolve tidal forces from other structure in the simulation volume, but for this particular application they do not have to be resolved with either the spatial or the mass resolution of our object of interest.

The solution to this is to use "nested" initial conditions, which are created by nesting subvolumes of higher spatial and mass resolution within a low-resolution parent grid. The low-resolution parent grid provides necessary tidal forces and a cosmological volume which provides reasonable statistics, and the nested subgrid, which is typically only a small fraction of the overall volume, provides higher mass and spatial resolution in the area of interest. This can be done recursively and can significantly save computational resources, particularly when the grids are further refined with adaptive mesh only in the most highly refined subvolume. See Figure 2.1 for an example.

An example of the overall cost savings can be shown very simply. Let us assume that the situation is as described for the simulation discussed above, where the overall ratio of box size to spatial resolution needed in the region of interest is $\sim 10^3$. Furthermore, we will assume that each subgrid has twice the spatial resolution as the one above it, suggesting that a grid at level L has to take twice as many timesteps as its parent grid in order to advance the same physical time. This assumption about time steps is reasonable for simulations using an explicit hydrodynamics algorithm, where $\Delta t = \kappa \Delta x / c_s$, where Δx is the grid size, c_s is the local sounds speed, and κ is a "Courant parameter" which has a value $0 < \kappa \leq 1$ and a typical value of $\kappa \simeq 0.3 - 0.5$. For the sake of comparison, we will compare the amount of time that it takes a simulation generated with a single, 1024^3 cell and particle initial condition to advance a fixed amount of time compared to a simulation with a 128^3 "root grid" and three 128^3 grids that are nested inside of each other, where the "root grid" resolves the same volume as the monolithic 1024^3 grid, and the most highly refined grid in the nested grid gives an equivalent resolution of 1024^3 (i.e. the same as the large monolithic grid). This nested calculation is similar to those discussed later in this thesis.

Since the simulation with 1024^3 grid cells has 8 times better spatial resolution than the root grid of the nested calculation, it will take 8 timesteps of the monolithic grid calculation to equal a single timestep on the root grid of the nested calculation (approximately). This means that the monolithic calculation will need to calculate $N_{mon} = 8 \times 1024^3 \simeq 8 \times 10^9$ individual cell-steps (which is a reasonable, though crude, proxy for computational cost). In order to evolve the nested grid simulation the same amount of physical time, it needs to take one timestep on its root grid, two on the first subgrid,

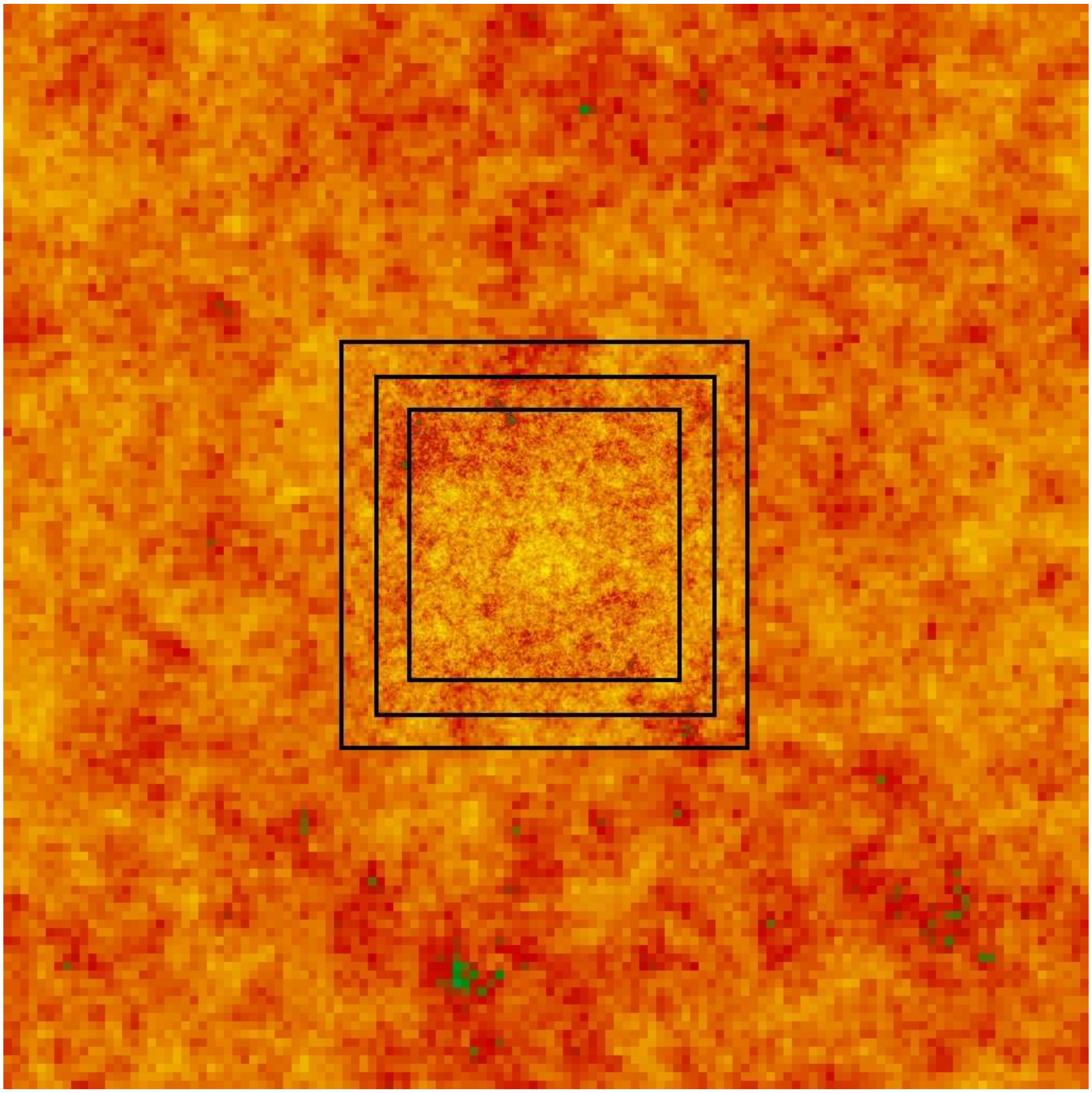

Figure 2.1: Example of a set of nested-grid initial conditions. The image is of a slice through the initial conditions that is one grid cell thick on the finest level (1/1024 of the thickness of the entire box), and shows log baryon density. The overall fluctuations are at the 20% level throughout the volume. The root grid ("level 0") is 128^3 grid cells, and each successive nested grid is shown by a black square. The Level 3 grid (central square) is the highest level of resolution and has the same resolution as a calculation with identical spatial extent to the entire simulation volume but 1024^3 grid cells. In calculations such as these adaptive mesh refinement is only allowed to take place within the highest level of nested-grid resolution, which in this case corresponds to only 1/64 of the overall volume.

four on the second subgrid, and 8 timesteps on the most highly refined subgrid. This comes out to a total of $N_{nest} = 128^3 + 2 \times 128^3 + 4 \times 128^3 + 8 \times 128^3 \simeq 3.15 \times 10^7$ cell-steps. The relative nested grid calculation takes $N_{nest}/N_{mon} = 0.0037$ as many cell-steps as the monolithic simulation, which is over a factor of 200 in computational costs! The reduction in memory needed is also substantial: The nested grid calculation has a total of $4 \times 128^3 \simeq 8 \times 10^6$ cells, whereas the monolithic grid calculation has $1024^3 \simeq 10^9$ cells. This is a savings of approximately 100 in memory.

The cost savings of these simulations is obvious. Fortunately, it is extremely easy to create nested initial conditions. One simply recursively generates initial conditions as described above covering the same physical volume, but at higher and higher spatial resolutions. The coarsest set has the desired spatial resolution of the "root grid," and each additional set of initial conditions is twice as highly resolved in space (resulting in dark matter particles that are 1/8 the mass of the particles in the next coarser set of initial conditions), and the subvolume of interest is extracted and put into a separate file. This method of generating initial conditions results in each grid having the appropriate perturbation spectra, and a much smaller overall simulation size in terms of computation and memory use.

The technique of "nesting" boxes of successively higher spatial and mass refinement described here is only limited by the available memory in which the necessary FFTs are performed. This limitation can be avoided by using an out-of-core initial conditions generation technique [128], or a more sophisticated method which does not require large, global FFTs to be performed [129]. These methods are costly in other ways, primarily in CPU time, but are necessary to be able to do calculations whose initial conditions have even higher dynamical range.

2.1.3 Numerical limitations of cosmological initial conditions

The standard methods and assumptions pertaining to simulations of early structure formation and the generation of cosmological initial conditions result in limitations which the careful simulator (or consumer of simulation-based data) must be cognizant of. These limitations can generally be broken up into large-scale effects which are due to finite box size and periodicity, and small scale effects due to finite resolution. Small-scale effects are dominated by the methods used in computation of forces, and the evolution of baryon fields. Therefore, we leave discussion of small-scale errors to later in this chapter (where the relevant numerical methods are discussed) and concentrate on large-scale effects here.

Truncation of the power spectrum due to effects of finite sampling causes issues at both large and small scales. Essentially, wave modes larger than half of the simulation ($L_{box}/2$) or smaller than twice the grid resolution ($2\Delta x$) are not represented correctly, which can have very important statistical effects. Gelb & Bertschinger [130] showed that truncation of power at the scale of the simulation box can result in estimates of σ_8, which corresponds to the size of density fluctuations when smoothed by a sphere 8 h^{-1} Mpc in radius, which are 40% lower than the input value for a simulation with a 50 h^{-1} Mpc box when using the previously-described method for generating initial conditions. In order to reduce the error to 10% the simulation volume had to double in size, to $\sim$ 100 Mpc. Pen [131] showed a method to fix this statistical issue, which generates initial conditions by convolving perturbations in real space instead of k-space. His method also allows the simple creation of nested-grid ICs without using large amounts of memory. This method was later implemented into a publicly-available package by Bertschinger [129]. An alternate method to correctly model large-scale power is discussed by Sirko [132], though in order to obtain correct statistics this method requires a large number of simulations to be performed, which is somewhat undesirable when performing large simulations due to overall computational expense.

A more concrete understanding of the effect of finite box sizes and periodicity on simulation results is provided by Barkana & Loeb [133]. They show that the majority of cosmological simulations, which are limited to relatively small volumes (particularly for study of objects in the early universe), largely omit cosmic scatter. This results in effects such as the epoch of reionization being shorter in simulations than has been observed and predicted analytically, non-physical biases in the halo mass function and halo correlations (i.e. lack of halo clustering), and overall mis-estimation of the halo mass function in small simulation volumes. This points towards a failing in small boxes of modeling the overall star formation rate at high redshift, as well as estimates of the amount and effects of feedback due to the ultraviolet radiation background at high redshift.

Though all of the issues discussed above are valid, we still must perform simulations of the formation of the first generations of stars in small simulation volumes. This is due entirely to the finite (though ever-growing) computational resources available to us. Extension of our computation capabilities by using nested grids (as described in Section 2.1.2) and techniques such as adaptive mesh refinement (described later in this chapter) allow us to increase our simulation volumes significantly, but not to the point where we can state with confidence that we are adequately resolving structure formation in the high-redshift universe with a high degree of statistical accuracy. This is not such a handicap, though, since we are interested in the dynamics of individual star forming clouds, and not the global star formation rate.

This begs the question, “if we can’t trust our statistics, what can be learned from this sort of simulation?” Fortunately, quite a bit. The simulations still accurately model the hydrodynamics, primordial gas chemistry, and gravitational interactions between baryons and dark matter (as well as self-interaction of these components). This allows us to simulate the evolution of individual star-forming halos very accurately. Additionally, we can obtain qualitative estimates of the reliability of our calculations by simulating a range of simulation volumes and extrapolating to larger box size. One can study the effects of, for example, feedback from a UV background on an individual halo, using a single random realization but varying other parameters, to get an idea of the effects of various feedback processes in a general sense. In general, quite a bit can be done with these small-box simulations as long as one is mindful of possible effects due to finite box size, and assesses their results in light of this.

2.2 The Enzo Simulation Code

'Enzo' is an adaptive mesh refinement cosmological simulation code developed by Greg Bryan and his collaborators [134, 135, 136, 137, 2]. This code couples an N-body particle-mesh (PM) solver [138, 139] used to follow the evolution of collisionless dark matter with an Eulerian AMR method for ideal gas dynamics by Berger & Colella [140], which allows extremely high dynamic range in gravitational physics and hydrodynamics in an expanding universe.

Enzo solves the equations of ideal gas dynamics in a coordinate system that is comoving with the expanding universe:

$$\frac{\partial \rho_b}{\partial t} + \frac{1}{a}\mathbf{v}_b \cdot \nabla \rho_b = -\frac{1}{a}\rho_b \nabla \cdot \mathbf{v}_b \tag{2.10}$$

$$\frac{\partial \mathbf{v}_b}{\partial t} + \frac{1}{a}(\mathbf{v}_b \cdot \nabla)\mathbf{v}_b = -\frac{\dot{a}}{a}\mathbf{v}_b - \frac{1}{a\rho_b}\nabla p - \frac{1}{a}\nabla \phi \tag{2.11}$$

$$\frac{\partial E}{\partial t} + \frac{1}{a}\mathbf{v}_b \cdot \nabla E = -\frac{\dot{a}}{a}\left(3\frac{p}{\rho_b} + \mathbf{v}_b^2\right) - \frac{1}{a\rho_b}\nabla \cdot (p\mathbf{v}_b) - \frac{1}{a}\mathbf{v}_b \cdot \nabla \phi + \Gamma - \Lambda \tag{2.12}$$

Where ρ_b is the comoving baryon density, $\mathbf{v}_b$ is the baryon velocity, p is the pressure, ϕ is the modified gravitational potential (in comoving coordinates, which is related to the potential in proper coordinates Φ by $\phi \equiv \Phi + 0.5\mathrm{a}\ddot{\mathrm{a}}\mathbf{x}^2$) and a is the "expansion parameter" which describes the expansion of a smooth, homogeneous universe as a function of time. This expansion parameter is related to the redshift: $a \equiv 1/(1+z)$. All derivatives are in comoving coordinates. E is the specific energy of the gas (total energy per unit mass), and Γ and Λ represent radiative heating and cooling processes as described below. Equations 2.10, 2.11 and 2.12 represent the conservation of mass, momentum and total (e.g., kinetic plus thermal) fluid energy.

The equations above are closed with three more equations:

$$E = p/[(\gamma - 1)\rho_b] + \mathbf{v}^2/2 \tag{2.13}$$

$$\nabla^2 \phi = \frac{4\pi G}{a}(\rho_b + \rho_{dm} - \rho_0) \tag{2.14}$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3a^3}(\rho_0 + 3p_0/c^2) + \Lambda/3. \tag{2.15}$$

where ρ_{dm} is the comoving dark matter density, ρ_0 is the comoving background density ($\rho_0 \equiv \Omega_{matter}\rho_{crit}$) and p_0 is the background pressure, γ is the ratio of specific heats and Λ is the cosmological constant. Equations 2.13, 2.14 and 2.15 are the equation of state, Poisson's equation in comoving form and an equation that describes the evolution of the comoving coordinates (i.e. the formula for the expansion of an isotropic, homogeneous universe). All particles in the simulation are governed by Newton's equations in comoving coordinates:

$$\frac{d\mathbf{x}_{part}}{dt} = \frac{1}{a}\mathbf{v}_{part} \tag{2.16}$$

$$\frac{d\mathbf{v}_{part}}{dt} = -\frac{\dot{a}}{a}\mathbf{v}_{part} - \frac{1}{a}\nabla\phi \tag{2.17}$$

Where $\mathbf{x}_{part}$ and $\mathbf{v}_{part}$ refer to the position and peculiar velocity of any particles in the system. Note that the system of equations 2.10-2.17 is valid only in regimes where relativistic effects are not important ($v_b, v_{dm} \ll c$, where c is the speed of light) and where cosmological curvature effects are unimportant, meaning that the simulation volume is much smaller than the radius of curvature of the universe, as defined as $r_{hub} \equiv c/H_0$, where c is the speed of light and H_0 is the Hubble constant.

Two different hydrodynamic methods are implemented in Enzo: the piecewise parabolic method (PPM) developed by Woodward & Colella [144] and extended to cosmology by Bryan et al. [145], and the hydrodynamics method used in the magnetohydrodynamics (MHD) code ZEUS [146, 147]. Below, we describe both of these methods in turn (PPM in Section 2.2.2 and the ZEUSmethod in Section 2.2.3), noting that PPM is viewed as the preferred choice for cosmological simulations since it is higher-order-accurate and is based on a technique that does not require artificial viscosity to resolve shocks. Enzo solves for the gravitational potential using the adaptive particle-mesh (APM) method, which is described in Section 2.2.4. The primordial chemistry and radiative cooling/heating packages are described in Section 2.2.5, and the star formation and feedback algorithms are described in Section 2.2.6.

2.2.1 The AMR machinery

Unlike moving mesh methods [141, 142] or methods that subdivide individual cells [143], Berger & Collela's AMR (also referred to as *structured* AMR) utilizes an adaptive hierarchy of grid patches at varying levels of resolution. Each rectangular grid patch (referred to as a "grid") covers some region of space in its *parent grid* which requires higher resolution, and can itself become the parent grid to an even more highly resolved *child grid.* ENZO's implementation of structured AMR places no restrictions on the number of grids at a given level of refinement, or on the number of levels of refinement. However, owing to limited computational resources it is practical to institute a maximum level of refinement ℓ_{max}.

The Enzo implementation of AMR allows arbitrary integer ratios of parent and child grid resolution. However, in practice refinement factors (defined as the ratio of parent grid resolution to child grid resolution) of two or four are typically used, with a refinement factor of two being most commonly used for cosmological simulations for reasons of efficiency. The ratio of box size to the maximum grid resolution of a given simulation is therefore $L/e = N_{\mathrm{root}} \times 2^{\ell_{\mathrm{max}}}$, where N_{root} is the number of cells along one edge of the root grid, ℓ_{max} is the maximum level of refinement allowed, and L and e are the box size and grid resolution of the most highly refined region, respectively.

The AMR grid patches are the primary data structure in Enzo. Each patch is treated as an individual object which can contain both field variables and particle data. Individual grids are organized into a dynamic, distributed hierarchy of mesh patches. Every processor keeps a description of the entire grid hierarchy at all times, so that each processor knows where all grids are. However, baryon and particle data for a given grid only exists on a single processor. See Figure 2.2 for a schematic example of this, and Figure 2.3 for an example of an Enzo simulation performed using this AMR scheme where both the baryon density and AMR grid hierarchy are shown together. The code handles load balancing on a level-by-level basis such that the workload on each level is distributed as uniformly as possible across all processors. Spatial locality of grids is not forced during message passing, for maximum flexibility (though not necessarily maximum efficiency). The MPI message passing library[1] is used to transfer data between processors.

Each grid patch in Enzo contains arrays of values for baryon and particle quantities. The baryon quantities are stored in arrays with the dimensionality of the simulation itself, which can be 1, 2 or 3 spatial dimensions. Grids are partitioned into a core of *real zones* and a surrounding layer of *ghost zones.* The real zones store field values and ghost zones are used to temporarily store values which have been obtained directly from neighboring grids or interpolated from a parent grid. These zones are necessary to accommodate the computational stencil of the hydrodynamics solvers (Sections 2.2.2 and 2.2.3) and the gravity solver (Section 2.2.4). The hydro solvers typically require ghost zones which are three cells deep and the gravity solver requires 6 ghost zones on every side of the real zones. This can lead to significant memory and computational overhead, particularly for smaller grid patches at high levels of refinement.

Since the addition of more highly refined grids is adaptive, the conditions for refinement must be specified. The criteria of refinement can be set by the threshold value of the overdensity of baryon gas or dark matter in a cell (which is really a refinement on the mass of gas or DM in a cell), the local Jeans length, the local density gradient, or local pressure and energy gradients. A cell reaching any or all of these criteria will then be flagged for refinement. Once all cells at a given level have been examined, rectangu-

[1] http://www-unix.mcs.anl.gov/mpi/

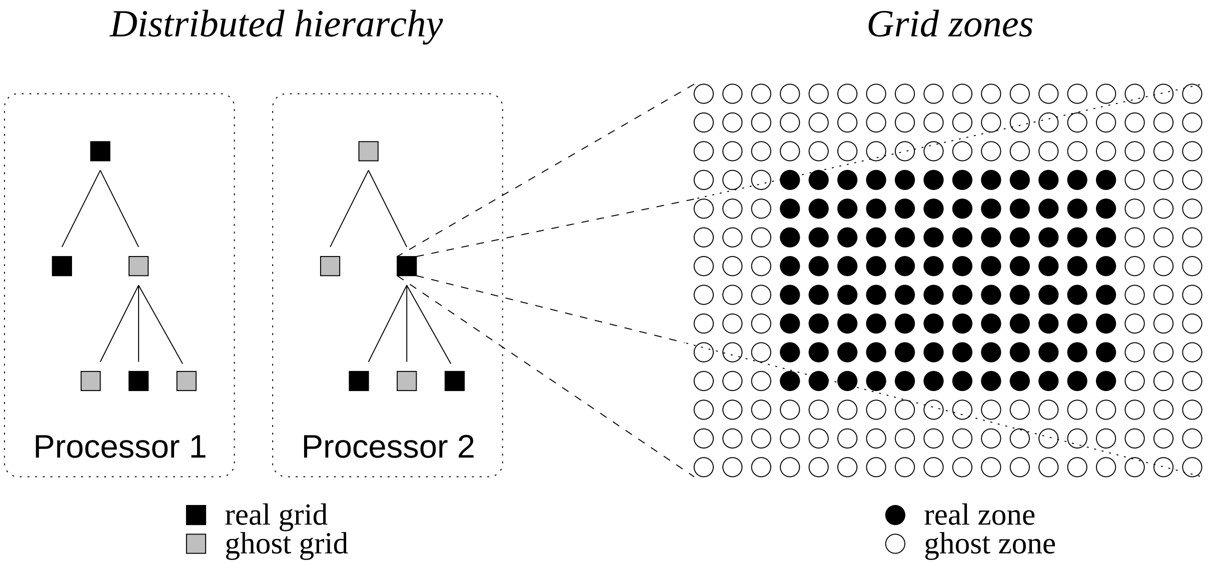

Figure 2.2: *Left:* Example of a simple, distributed AMR hierarchy showing real and ghost grids. *Right:* Example 2D Enzo grid showing real and ghost zones, as needed for the PPM hydro stencil. Image courtesy of James Bordner (Cent. Astrophysics and Space Sciences, UCSD).

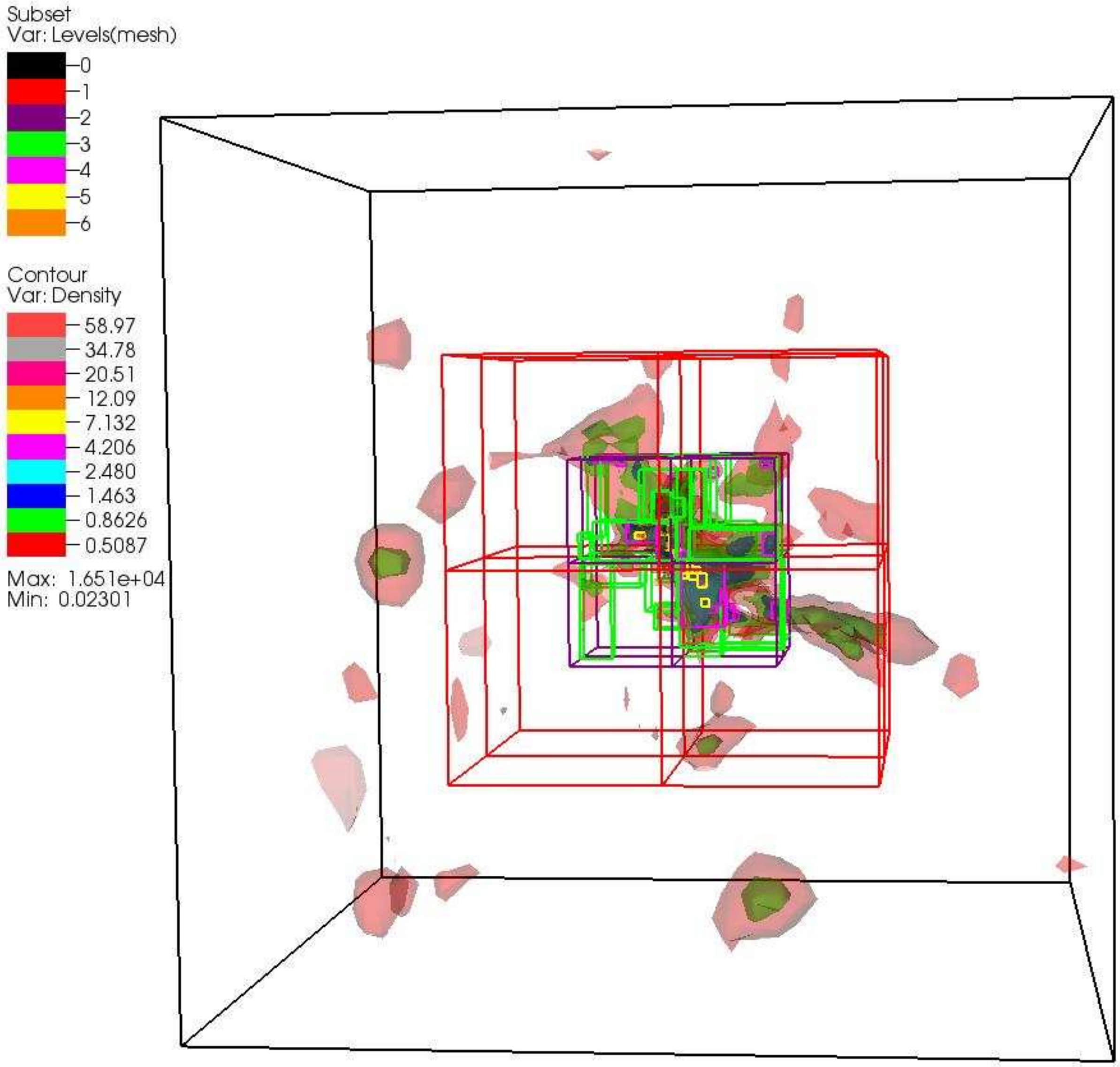

Figure 2.3: Example of an Enzo simulation showing the AMR grid hierarchy. This is a simulation of the collapse of a single massive halo with a 32^3 root grid and two 32^3 static nested grids. AMR is only allowed within the most highly refined static nested grid. Log baryon density is shown by colored isocontours with values corresponding to the legend at center left. The rectangular solid wire frames correspond to individual AMR grids, with different colors corresponding to level as indicated by the legend at top left.

lar boundaries are determined which minimally encompass the flagged cells. A refined grid patch is introduced within each such bounding rectangle. Thus the cells needing refinement, as well as adjacent cells within the patch which do not need refinement, are refined. While this approach is not as memory efficient as cell-splitting AMR schemes, it offers more freedom with finite difference stencils. For example, PPM requires a stencil of seven cells per dimension. This cannot easily be accommodated in cell-splitting AMR schemes. In the simulations discussed in this thesis we typically use baryon and dark matter overdensities as our refinement criteria, though for some higher-resolution simulations we also refine on other criteria as needed.

In Enzo, resolution of the equations being solved is adaptive in time as well as in space. The timestep in Enzo is satisfied on a level-by-level basis by finding the largest timestep such that multiple criteria are satisfied on each level. The timestep criteria used by Enzo are (showing the one-dimensional case for clarity):

$$\Delta t_{hydro} = min\left(\kappa_{hydro}\frac{a\Delta x}{c_s + |v_x|}\right)_L, \tag{2.18}$$

$$\Delta t_{dm} = min\left(\kappa_{dm}\frac{a\Delta x}{v_{dm,x}}\right)_L, \tag{2.19}$$

$$\Delta t_{exp} = f_{exp}\left(\frac{a}{\dot{a}}\right), \tag{2.20}$$

$$\Delta t_{accel} = min\left(\sqrt{\frac{\Delta x}{\mathbf{g}}}\right)_L \tag{2.21}$$

In equations 2.18-2.21, the $min(\ldots)_L$ formalism means that this value is calculated for all cells on a given level L and the minimum overall value is taken as the timestep. Equation 2.18 ensures that all cells satisfy the Courant-Freidrichs-Levy (CFL) condition for accuracy and stability of an explicit finite difference discretization of the Euler equations. Effectively this condition forces the timestep to be small enough such that any changes in the fluid propagate less than a single grid spacing, Δx. In this equation, κ_{hydro} is a numerical constant with a value of $0 < \kappa_{hydro} \leq 1$ (with a typical value of $\kappa_{hydro} \sim 0.3 - 0.5$) that ensures that the CFL condition is always met, and c_s and v_x are the sound speed and peculiar baryon velocity in a given cell. Equation 2.19 is analogous to Equation 2.18 and ensures accuracy in the N-body solver by requiring that no dark matter particle travels more than one cell width. The parameter κ_{dm} is analogous to $\kappa_{dthydro}$, with an identical range of values. Equation 2.20 limits the timestep such that the expansion parameter a can only change by a fractional amount of $f_{exp} = \Delta a/a$, where f_{exp} is a user-defined parameter and has typical values of $f_{exp} = 0.01 - 0.02$. This is required for the stability of the PPM algorithm in comoving coordinates, and typically limits the timestep only at high redshifts when densities are relatively homogeneous. Equation 2.21 is supplementary to equation 2.18 in that it takes into account the possibility of large accelerations causing numerical instabilities by violating the Courant condition. In this equation, **g** is the gravitational acceleration in each cell on level L.

For simplicity and stability, all cells on a given level are advanced using the same timestep, which is taken to be the minimum value of Equations 2.18-2.21. Once a level L has been advanced in time by Δt_L, all grids at level $L + 1$ are advanced, using the same criteria for timestep calculation described above, until they reach the same physical time as the grids at level L. At this point grids at level $L + 1$ exchange flux information with their parents grids, providing a more accurate solution on level L. This step, controlled by the parameter *FluxCorrection* in Enzo, is extremely important, and can significantly affect simulation results if not used in an AMR calculation. At the end of every timestep on every level each grid updates its ghost zones by exchanging information with its neighboring grid patches (if any exist) and/or by interpolating from a parent grid. In addition, cells are examined to see if they should be refined or de-refined, and the entire grid hierarchy is rebuilt at that level (including all more highly refined levels). The timestepping and hierarchy advancement/rebuilding process described here is repeated recursively on every level to the specified maximum level of refinement in the simulation.

2.2.2 Hydrodynamics with the piecewise parabolic method

The primary hydrodynamic method used in Enzo is based on the piecewise parabolic method (PPM) of Woodward & Colella [144], which has been significantly modified for the study of cosmological fluid flows. The method is described in Bryan et al. [145], but we provide a short description here for clarity.

PPM is an explicit, higher order-accurate version of Godunov's method for ideal gas dynamics with third order-accurate piecewise parabolic monotonic interpolation and a nonlinear Riemann solver for shock capturing. It does an excellent job of capturing strong shocks in at most two cells. Multidimensional schemes are built up by directional splitting and produce a method that is formally second order-accurate in space and time which explicitly conserves mass, linear momentum, and energy. The conservation laws for fluid mass, momentum and energy density are written in comoving coordinates for a Friedman-Robertson-Walker space-time, as described previously in Equations 2.10 through 2.12. Both the conservation laws and the Riemann solver are modified to include gravity, which is solved using an adaptive particle-mesh (PM) technique (see Section 2.2.4). The terms due to cosmological expansion, as well as primordial chemistry and radiative heating and cooling, are solved in a separate step because they have different numerical behavior, and therefore must be treated differently to ensure stability. Note that unlike the ZEUS hydro scheme, PPM does not need to use artificial viscosity to resolve shocks.

The system of equations described above works well for systems with relatively low

Mach numbers, as long as these systems are well resolved. However, cosmology is replete with situations where there are bulk hypersonic flows. In these situations, the ratio of kinetic to thermal energy can be very high – in some situations up to $10^6 - 10^8$. This implies that the thermal energy is an extremely tiny fraction of the kinetic energy, which can cause numerical problems when one is interested in just the thermal energy of the gas, since Equation 2.12 solves for the total energy. In this system of equations, the thermal energy E_{therm} is calculated as $E - E_{kin}$, where E is the total specific energy as calculated in equation 2.12 and E_{kin} is the specific kinetic energy, $0.5\mathbf{v}_b^2$. In hypersonic flows E and E_{kin} are nearly the same, and any number calculated as the difference of these is going to be strongly affected by numerical error. To avoid this problem, Enzo also solves the internal energy equation in comoving coordinates:

$$\frac{\partial e}{\partial t} + \frac{1}{a}\mathbf{v}_b \cdot \nabla e = -\frac{3(\gamma - 1)\dot{a}}{a} e - \frac{p}{a\rho}\nabla \cdot \mathbf{v}_b \tag{2.22}$$

In this equation e is the internal energy and the other terms are as described previously. The code still conserves total energy (E) as well. In order to maintain consistency, both equations are solved at all times in all cells, with the equation for the total energy (eqtn. 2.12) being used for hydrodynamics routines and the internal energy e being used when temperature is required. When pressure is required for dynamic purposes, the total energy is used if the ratio of thermal energy to total energy is less than some threshold value η, and the internal energy is used for values of the ratio larger than η. A typical value of this parameter is 10^{-3}. This *dual energy formulation* ensures that the method produces the correct entropy jump at strong shocks and also yields accurate pressures and temperatures in cosmological hypersonic flows.

2.2.3 Hydrodynamics with the ZEUS hydrodynamic method

As a check on PPM, Enzo also includes an implementation of the finite-difference hydrodynamic algorithm employed in the compressible magnetohydrodynamics code 'ZEUS' [146, 147]. Fluid transport is solved on a Cartesian grid using the upwind, monotonic advection scheme of van Leer [148] within a multistep (operator split) solution procedure which is fully explicit in time. This method is formally second order-accurate in space but first order-accurate in time. It is important to note that this method conserves internal energy rather than total energy, so the "dual energy formulation" discussed in Section 2.2.2 is unnecessary.

Operator split methods break the solution of the hydrodynamic equations into parts, with each part representing a single term in the equations. Each part is evaluated successively using the results preceding it. The individual parts of the solution are grouped into two steps, called the *source* and *transport* steps.

The ZEUS method uses a von Neumann-Richtmeyer artificial viscosity to smooth shock discontinuities that may appear in fluid flows and can cause a break-down of finite-difference equations. The artificial viscosity term is added in the source terms as

$$\rho\frac{\partial \mathbf{v}}{\partial t} = -\nabla p - \rho\nabla\phi - \nabla\cdot\mathbf{Q} \tag{2.23}$$

$$\frac{\partial e}{\partial t} = -p\nabla\cdot\mathbf{v} - \mathbf{Q}:\nabla\mathbf{v}, \tag{2.24}$$

where $\mathbf{v}$ is the baryon velocity, ρ is the mass density, p is the pressure, e is the internal energy density of the gas and $\mathbf{Q}$ is the artificial viscosity stress tensor, such that:

$$Q_{ii} = \begin{cases} Q_{\mathrm{AV}}\rho_b(a\Delta v_i + \dot{a}\Delta x_i)^2, & \text{for } a\Delta v_i + \dot{a}\Delta x_i < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

and

$$Q_{ij} = 0 \quad \text{for} \;\; i \neq j. \tag{2.26}$$

Δx_i and Δv_i refer to the comoving width of the grid cell along the i-th axis and the corresponding difference in gas peculiar velocities across the grid cell, respectively, and a is the cosmological scale factor. Q_{AV} is a constant with a typical value of 2. We refer the interested reader to Anninos et al. [149] for more details.

The limitation of a technique that uses an artificial viscosity is that, while the correct Rankine-Hugoniot jump conditions are achieved, shocks are broadened over 6-8 mesh cells and are thus not treated as true discontinuities. This may cause unphysical pre-heating of gas upstream of the shock wave, as discussed in Anninos et al. [149].

2.2.4 Gravity Solver

There are multiple methods to compute the gravitational potential (which is an elliptic equation in the Newtonian limit) in a structured AMR framework. One way would be to model the dark matter (or other collisionless particle-like objects, such as stars) as a second fluid in addition to the baryonic fluid and solve the collisionless Boltzmann equation, which follows the evolution of the fluid density in six-dimensional phase space. However, this is computationally prohibitive owing to the large dimensionality of the problem, making this approach unappealing for the cosmological AMR code.

Instead, Enzo uses a particle-mesh N-body method to calculate the dynamics of collisionless systems [139]. This method follows trajectories of a representative sample of individual particles and is much more efficient than a direct solution of the Boltzmann equation in most astrophysical situations. The gravitational potential is computed by solving the elliptic Poisson's equation (Eqtn. 2.14).

These equations are finite-differenced and for simplicity are solved with the same timestep as the hydrodynamic equations, as discussed in Section 2.2.1. The dark matter particles are distributed onto the grids using the cloud-in-cell (CIC) interpolation technique to form a spatially discretized density field (analogous to the baryon densities used to calculate the equations of hydrodynamics). After sampling dark matter density onto the grid and adding baryon density if it exists (to get the total matter density in each cell), the gravitational potential is calculated on the periodic root grid using a fast Fourier transform. In order to calculate more accurate potentials on subgrids, Enzo resamples the dark matter density onto individual subgrids using the same CIC method as on the root grid, but at higher spatial resolution (and again adding baryon densities if applicable). Boundary conditions are then interpolated from the potential values on the parent grid (with adjacent grid patches on a given level communicating to ensure that their boundary values are the same), and then a multigrid relaxation technique is used to calculate the gravitational potential at every point within a subgrid. Forces are computed on the mesh by finite-differencing potential values and are then interpolated to the particle positions, where they are used to update the particle's position and velocity information. Potentials on child grids are computed recursively and particle positions are updated using the same timestep as in the hydrodynamic equations. Particles are stored in the most highly refined grid patch at the point in space where they exist, and particles which move out of a subgrid patch are sent to the grid patch covering the adjacent volume with the finest spatial resolution, which may be of the same spatial resolution, coarser, or finer than the grid patch that the particles are moved from. This takes place in a communication process at the end of each timestep on a level.

At this point it is useful to emphasize that the effective force resolution of an adaptive particle-mesh calculation is approximately twice as coarse as the grid spacing at a given level of resolution. The potential is solved in each grid cell; however, the quantity of interest, namely the acceleration, is the gradient of the potential, and hence two potential values are required to calculate this. In addition, it should be noted that the adaptive particle-mesh technique described here is very memory intensive: in order to ensure reasonably accurate force resolution at grid edges the multigrid relaxation method used in the code requires a layer of ghost zones which is very deep – typically 6 cells in every direction around the edge of a grid patch. This greatly adds to the memory requirements of the simulation, particularly because subgrids are typically small (on the order of $12^3 - 16^3$ real cells for a standard cosmological calculation) and ghost zones can dominate the memory and computational requirements of the code.

2.2.5 Radiative processes and non-equilibrium primordial chemistry

Though the equations of hydrodynamics described above are a closed system, they are still missing a crucial piece of physics: radiative heating and cooling. Radiative cooling is extremely important in many astrophysical situations, as is the heating of gas from some sort of radiation background. Enzo has a very simple Sutherland and Dopita equilibrium cooling function [150] implemented, which uses a cooling table assuming a fixed metallicity of $Z = 0.3 Z_\odot$, and also a nonequilibrium heating/cooling model that assumes gas of primordial composition exposed to a uniform metagalactic ultraviolet (UV) background that varies with time [151].

The simulations discussed in this thesis almost exclusively use the nonequilibrium routines, described in great detail by Abel et al. and Anninos et al. [152, 153] and summarized in Appendix A. These routines follow the non-equilibrium chemistry of a gas of primordial composition with 9 total species: $H, H^+, He, He^+, He^{++}, H^-, H_2^+, H_2$, and e^-. The code also calculates radiative heating and cooling, following atomic line excitation, recombination, collisional excitation, free-free transitions, molecular line excitations, and Compton scattering of the cosmic microwave background, as well as any of approximately a dozen different models for a metagalactic UV background that heat the gas via photoionization and photodissociation. The multispecies rate equations are solved out of equilibrium to properly model situations where, e.g., the cooling rate of the gas is much shorter than the hydrogen recombination time. The effect of this nonequilibrium cooling is to leave behind a much larger fraction of residual free electrons than one would expect if the assumption of equilibrium were being made. The practical effect of this is that more H^- is formed, which subsequently produces hydrogen molecules. If large amounts of H_2 is formed it can greatly increase the cooling rate of primordial gas at relatively low temperatures ($T \leq 10^4$ K). This can efficiently cool the gas to approximately 200 K, which significantly reduces the Jeans mass of the gas. Correct modeling of the formation of molecular hydrogen is crucial to the study of star formation in a primordial gas.

A total of 9 kinetic equations are solved, including 29 kinetic and radiative processes, for the 9 species mentioned above. See Table 2.1 for the collisional processes and Table 2.2 for the radiative processes solved.

The chemical reaction equation network is technically challenging to solve due to the huge range of reaction timescales involved – the characteristic creation and destruction timescales of the various species and reactions can differ by many orders of magnitude. As a result, the set of rate equations is extremely stiff, and an explicit scheme for integration of the rate equations can be exceptionally costly if small enough timesteps are taken to keep the network stable. This makes an implicit scheme much more preferable for such a set of equations. However, an implicit scheme typically require an iterative procedure

Collisional Processes

(1)	H	+	e^-	→	H^+	+	$2\mathrm{e}^-$
(2)	H^+	+	e^-	→	H	+	γ
(3)	He	+	e^-	→	He^+	+	$2\mathrm{e}^-$
(4)	He^+	+	e^-	→	He	+	γ
(5)	He^+	+	e^-	→	He^{++}	+	$2e^-$
(6)	He^{++}	+	e^-	→	He^+	+	γ
(7)	H	+	e^-	→	H^-	+	γ
(8)	H^-	+	H	→	H_2	+	e^-
(9)	H	+	H^+	→	H_2^+	+	γ
(10)	H_2^+	+	H	→	H_2	+	H^+
(11)	H_2	+	H^+	→	H_2^+	+	H
(12)	H_2	+	e^-	→	$2\mathrm{H}$	+	e^-
(13)	H_2	+	H	→	$3\mathrm{H}$		
(14)	H^-	+	e^-	→	H	+	$2\mathrm{e}^-$
(15)	H^-	+	H	→	$2\mathrm{H}$	+	e^-
(16)	H^-	+	H^+	→	$2\mathrm{H}$		
(17)	H^-	+	H^+	→	H_2^+	+	e^-
(18)	H_2^+	+	e^-	→	$2\mathrm{H}$		
(19)	H_2^+	+	H^-	→	H_2	+	H
(20)	$3\mathrm{H}$			→	H_2	+	H

Table 2.1: Collisional processes solved in the Enzo nonequilibrium primordial chemistry routines.

Radiative Processes

(21)	H	+	γ	→	H^+	+	e^-
(22)	He	+	γ	→	He^+	+	e^-
(23)	He^+	+	γ	→	He^{++}	+	e^-
(24)	H^-	+	γ	→	H	+	e^-
(25)	H_2	+	γ	→	H_2^+	+	e^-
(26)	H_2^+	+	γ	→	H	+	H^+
(27)	H_2^+	+	γ	→	$2\mathrm{H}^+$	+	e^-
(28)	H_2	+	γ	→	H_2^*	→	$2\mathrm{H}$
(29)	H_2	+	γ	→	$2\mathrm{H}$		

Table 2.2: Radiative processes solved in the Enzo nonequilibrium primordial chemistry routines.

to converge, and for large networks (such as this one) an implicit method can be very time-consuming, making it undesirable for a large, three-dimensional simulation.

Enzo solves the rate equations using a method based on a backwards differencing formula (BDF) in order to provide a stable and accurate solution. This technique is optimized by taking the chemical intermediaries H^- and H_2^+, which have large rate coefficients and low concentrations, and grouping them into a separate category of equations. Due to their fast reactions these species are very sensitive to small changes in the more abundant species. However, due to their low overall concentrations, they do not significantly influence the abundance of species with higher concentrations. Therefore, reactions involving these two species can be decoupled from the rest of the network and treated independently. In fact, H^- and H_2^+ are treated as being in equilibrium at all times, independent of the other species and the hydrodynamic variables. This allows a large speedup in solution as the BDF scheme is then applied only to the slower 7-species network on timescales closer to those required by the hydrodynamics of the simulation. Even so, the accuracy and stability of the scheme is maintained by subcycling the rate solver over a single hydrodynamic timestep. These subcycle timesteps are determined so that the maximum fractional change in the electron concentration is limited to no more than 10% per timestep.

It is important to note the regime in which this model is valid. According to Abel et al. and Anninos et al. [152, 153], the reaction network is valid for temperatures between $10^0 - 10^8$ K. The original model discussed in these two references is only valid up to $n_H \sim 10^4$ cm^{-3}. However, addition of the 3-body H_2 formation process (equation 20 in Table 2.1) allows correct modeling of the chemistry of the gas up until the point where collisionally-induced emission from molecular hydrogen becomes an important cooling process, which occurs at $n_H \sim 10^{14}$ cm^{-3}. A further concern is that the optically thin approximation for radiative cooling breaks down, which occurs before $n_H \sim 10^{16} - 10^{17}$ cm^{-3}. Beyond this point, modifications the cooling function that take into account the non-negligible opacity in the gas must be made, as discussed by Ripamonti & Abel [47]. Even with these modifications, a completely correct description of the cooling of this gas will require some form of radiation transport, which will greatly increase the cost of the simulations.

Several processes are neglected. The deuterium atom and its processes are completely ignored, which may have some effect. Recent work shows that HD is a more effective coolant than previously thought [154]. However, the fractional abundance of HD is so low that under circumstances relevant to the formation of a Population III star in an un-ionized region it should be sub-dominant. However, the enhanced electron fraction in fossil HII regions (as discussed later in this thesis) could result in the HD molecule becoming a dominant cooling mechanism at relatively low ($\sim$ few hundred K) temperatures, and could potentially cool the gas down to below 100 K, which can enhance fragmentation and could have important consequences for the IMF of primordial stars forming in a relic HII region [78].

Aside from deuterium, the chemical reactions involving lithium are also neglected. According to Galli & Palla [27], these are not important for the density and temperature regimes explored by the simulations discussed in this thesis. However, at higher densities it is possible that there are regimes where lithium can be an important coolant.

2.2.6 Star Formation and Feedback Algorithms

While the physics discussed previously is all crucial to the study of cosmological structure formation, most cosmological observations are of stars and related phenomena. Additionally, stars eject energy and metal- enriched gas throughout their lives, drastically modifying their own environment. The formation of galaxies and clusters cannot be completely modeled without including the feedback of energy and metals. In particular, it is thought that feedback is crucial for suppressing the large numbers of dwarf-like galaxies that CDM theories predict [16, 155]. An early burst of star formation could remove a large fraction of cold gas from such systems [156, 157]. Also, the unexpectedly low luminosities of small clusters and groups (relative to the richest clusters) is commonly explained through feedback [158]. Energy ejected during the formation of the cluster ellipticals lowers the central density and hence the X-ray luminosity of such clusters [159]. Therefore, the inclusion of star formation and the feedback of energy and metals into the intergalactic medium in a cosmological code is crucial for many reasons.

We have extended the Enzo code to include multiple models for star formation and feedback. It is difficult to directly simulate the formation of individual stars in the context of simulations of galaxy formation and evolution due to the immense computational cost. Therefore, we adopt parametric methods which attempt to model star forming regions. One model is based on Kravtsov's method [160], and the other is based on the method of Cen & Ostriker [170], which has been modified for use in an AMR code. The basic ideas behind the methods are straightforward and observationally motivated. Similar (though somewhat more advanced) methods have been employed for the smoothed particle hydrodynamics (SPH) method, most recently by Springel & Hernquist [162, 163, 164]. This method assumes a multiphase IGM and has been shown to accurately reproduce the cosmic star formation rate, and will be implemented into Enzo in the near future.

In the following sections we will detail both the Kravtsov and Cen & Ostriker star formation methods separately. Tests of the star formation and feedback algorithms will not be shown, since this work is currently in progress (and an extension of the AMR/SPH code comparison described in Chapter 3).

The Kravtsov star formation and feedback algorithm

Kravtsov's method of star formation is designed to reproduce the global Schmidt law of star formation [160, 165]. This algorithm is deliberately minimal, and is explicitly geared towards modeling star formation in a phenomenological way on kiloparsec scales. Stars are assume to form with a characteristic gas timescale τ_* such that $\dot{\rho}_* = \rho_{gas}/\tau_*$ where ρ_{gas} and ρ_* are the baryon gas and stellar densities, respectively. This "constant efficiency" model on the scale of star formation regions is well motivated observationally [166, 167]. Star formation is only allowed to take place in very dense regions with $\rho_{gas} \geq \rho_{SF}$, where ρ_{SF} is a constant proper density threshold above which star formation is allowed to occur. No other criteria are imposed. Kravtsov's typical choices for τ_* and ρ_{SF} are $\tau_* = 4$ Gyr and $\rho_{SF} = 1.64$ M$_\odot$ pc^{-3} ($n_H \sim 50$ cm^{-3}). The adopted timescale is derived from the observationally-determined normalization of the Schmidt law, and the density threshold is determined by observations of star forming regions on ~ 100 pc scales. Note that the density threshold is in proper, not comoving, units.

Algorithmically, the star formation events in Kravtsov's code are assumed to occur once every global time step $\Delta t_0 \leq 10^7$. In cells where star formation is determined to occur (i.e. $\rho_{gas} \geq \rho_{SF}$) a collisionless "star particle" is assumed to form, with a mass $m_* = \dot{\rho}_* V_{cell} \Delta t_0$, where $\dot{\rho}_*$ is described above and V_{cell} is the volume of the mesh cell. These star particles are formed instantaneously, with all m_* of gas being taken out of the cell and immediately deposited into the star particle. This particle is then given the velocity of the gas in the cell, and thereafter treated as a collisionless particle. The Enzo implementation of this algorithm is similar, except that instead of forming stars only at the root grid time step, we allow stars to form at the time step of the highest level of resolution at any particular point in space. As can be seen from the equation for m_* above, this can result in very small stellar masses. To avoid memory and processor time issues related to having very large numbers of star particles we impose a threshold mass $M_{*,min}$ such that a star particle only forms if $m_* \geq M_{*,min}$. An appropriate choice of $M_{*,min}$ does not significantly change the star overall star formation history of a simulation, though it may delay the onset of star formation in a given cell relative to a simulation without a particle mass threshold.

Each "star particle" is assumed to represent an ensemble of stars and is treated as a single-age stellar population. Kravtsov assumes that the IMF is described by a Miller & Scalo functional form with stellar masses between 0.1 and 100 $M_\odot$ [168]. All stars in this IMF with $M_* > 8$ $M_\odot$ deposit 2×10^{51} ergs of thermal energy and a mass $f_z M_*$ of metals into the cell in which they form without delay, with $f_z \equiv min(0.2, 0.01\ M_* - 0.06)$ (i.e. instantaneous deposition of metals). The definition of f_z is a rough approximation of the results of Woosley & Weaver [169].

Kravtsov reports that simulations with this algorithm reliably reproduce the star formation rate-gas surface density relation of the Schmidt law, and are not particularly sensitive to numerical parameters [160]. He also notes that this is surprisingly insensitive to the presence or absence of feedback and details of the cooling and heating properties of the gas, which suggests that the global star formation rate is determined by gravitationally- driven supersonic turbulence (on large scales) rather than stellar feedback or thermal instabilities on small scales.

The Cen & Ostriker star formation algorithm

The Cen & Ostriker method is a heuristic model of star formation on galactic scales. This method, first described in a 1992 paper [170], is similar in some ways to the Kravtsov algorithm but has more complex criteria for determining where stars should be formed. In this method, cells that form stars must have a baryon overdensity higher than some threshold $\rho_b/\bar{\rho}_b \geq \eta$ where ρ_b is the baryon density in that cell, $\bar{\rho_b}$ is the mean baryon density in the simulation, and η is the user-defined overdensity threshold. Additionally, the gas in the cells must be contracting, cooling rapidly, and gravitationally unstable, e.g.:

$$\nabla \cdot \mathbf{v}_b < 0, \tag{2.27}$$

$$t_{cool} < t_{dyn} \equiv \sqrt{3\pi/32G\rho_{tot}}, \tag{2.28}$$

$$m_b > m_J \equiv G^{-3/2}\rho_b^{-1/2}C^3\left[1+\frac{\delta\rho_d}{\delta\rho_b}\right]^{-3/2} \tag{2.29}$$

where $\mathbf{v}$ is the velocity of the gas in the cell, ρ_b and ρ_d are the cell's baryon and dark matter density, respectively, $\rho_{total} = \rho_b + \rho_d$, m_b and m_j are the baryonic mass in the cell and jeans mass of the cell, and C is the isothermal sound speed in the cell. If all of these criteria are met, the mass of a star particle is calculated as $m_* = m_b \frac{\Delta t}{t_{dyn}} f_{*eff}$, where f_{*eff} is the star formation efficiency parameter.

If m_* is greater than a minimum star mass m_{*min}, a particle is created and given several attributes: Mass, a unique index number, the time of formation t_{form}, the local dynamical free-fall time t_{dyn} and the metallicity fraction of the baryon gas in the cell f_{Zb}. There is a user-defined minimum dynamical time $T_{dyn,min}$ which is observationally motivated and affects the feedback rates (see below). The particle is placed in the center of the cell and given the same peculiar velocity as the gas in the cell, and is then treated in the same manner as the dark matter particles. An amount of baryon gas corresponding to the new particle's mass is removed from the cell.

In addition, we have added a stochastic star formation algorithm that keeps track of all of the sub-mass threshold stars that should have been created and when the total mass of uncreated stars is greater than the minimum mass, a star particle is created.

The feedback of energy and metals into the baryon gas is similar to the Kravtsov feedback, with some important differences. The star formation algorithm creates each star particle instantaneously. However, feedback should take place over a significant timescale, as all of the stars contained within the "star particle" would form over a long period of time. Therefore, we assume that for the purposes of feedback that the mass of stars formed at a time t with finite timestep Δt is:

$$\begin{aligned}\Delta m_{sf} &= m_* \, [\, \left(1 + \frac{t - t_{form}}{t_{dyn}}\right) exp\left(\frac{-(t - t_{form})}{t_{dyn}}\right) \\ &\quad - \left(1 + \frac{t + \Delta t - t_{form}}{t_{dyn}}\right) exp\left(\frac{-(t + \Delta t - t_{form})}{t_{dyn}}\right) \,] \end{aligned} \tag{2.30}$$

which can be represented more clearly in integral form:

$$\int_t^{t+Dt} \frac{dM}{dt} dt = \int_t^{t+Dt} m_* \frac{dt}{t_{tyn}} \left(\frac{t - t_{form}}{t_{dyn}}\right) exp\left(- \frac{t - t_{form}}{t_{dyn}}\right) \tag{2.31}$$

During this timestep, we assume that the star particle feeds back metal-enriched gas and thermal energy from supernovae and from stellar winds. Since massive stars have very short lifetimes, we assume that there is an immediate feedback of some fraction f_{SN} of the rest energy from the stars created that timestep into the baryon gas, such that $E_{add} = f_{SN} \Delta m_{sf} c^2$, where c is the speed of light. In addition, a fraction f_{Z*} of the metal from the star particle is fed back into the baryon gas, which takes into account the effects of metal recycling. Finally, a fraction of the mass f_{m*} from all stars (rather than just supernovae) is fed back into the gas along with momentum in order to simulate the mass ejection from non-exploding stars via stellar winds.

There are six user-defined parameters in this algorithm: three deal with the star formation (η, m_{*min} and $T_{dyn,min}$), and three deal with feedback (f_{SN}, f_{Z*} and f_{m*}). Some of these parameters are completely free, while others can be guided by observation or theory. For example, the supernova feedback parameter, f_{SN}, can be constrained assuming that, for every $200 M_\odot$ of stars created, one supernova occurs, and this event feeds back approximately 10^{51} ergs of thermal energy, giving:

$$f_{SN} = \frac{10^{51} ergs}{200 \ M_\odot c^2} \simeq 3 \times 10^{-6} \tag{2.32}$$

The metal yield f_{Z*}, defined as the mass in metals produced per unit mass of stars created, can be constrained by a theoretical model of Woosley & Weaver [169]. This model suggests that $f_{Z*} = 0.02$ is an appropriate number. The minimum dynamical time is set to be $T_{dyn,min} = 10^7$ years to agree with timescales seen in nearby OB associations.

The other parameters, such as the overdensity threshold η, minimum star mass m_{*min}, and mass ejection fraction f_{m*} are not well constrained either theoretically or observationally. Indeed, m_{*min} is a purely numerical parameter designed to keep the code from producing too many star particles, and thus has no observational or theoretical counterpart. These parameters have to be set by performing parameter studies and comparing to observations. Unfortunately, the range of parameter space is large, and the results may be degenerate for some combinations of these parameters.