Hydrodynamic and N-body schemes on an unstructured, adaptive mesh with applications to cosmological simulations

Guohong Xu*

Board of Studies in Astronomy and Astrophysics, University of California at Santa Cruz, Santa Cruz, California 95064, USA

Accepted 1997 February 5. Received 1997 January 10; in original form 1996 October 14

ABSTRACT
The theory and application of numerical methods for unstructured meshes have been improved significantly in recent years. Because the grids can be placed arbitrarily in space, unstructured meshes can provide much higher spatial resolution than regular meshes. The built-in nature of mesh adaptivity for unstructured meshes gives one way to simulate highly dynamic, hierarchical problems involving both collisionless dark matter and collisional gas dynamics. In this paper, we describe algorithms to construct unstructured meshes from a set of points with periodic boundary conditions through Delaunay triangulation, and algorithms to solve hydrodynamic and N-body problems on an unstructured mesh. A combination of a local transformation algorithm and the traditional Bowyer–Watson algorithm gives an efficient approach to perform Delaunay triangulation. A novel algorithm to solve N-body equations of motion on an unstructured mesh is described. Poisson’s equation is solved using the conjugate gradient method. A gas-kinetic scheme based on the BGK model to solve Euler equations is used to evolve the hydrodynamic equations. We apply these algorithms to solve cosmological settings, which involve both dark and baryonic matter. Various cooling and heating processes for primordial baryonic matter are included in the code. The numerical results show that the N-body and hydrodynamic algorithms based on unstructured meshes with mesh refinement are well suited for hierarchical structure formation problems.

Key words: methods: numerical – galaxies: formation – cosmology: theory.

1 INTRODUCTION
Numerical simulations in astrophysics turn out to be very challenging because of the large dynamical range required in three dimensions. Examples include modelling of star-forming regions and the origin of galaxies. In cosmology, structures are believed to have formed hierarchically, requiring a simultaneous modelling of structures on scales of ~100 Mpc and ~10 kpc. Various hydrodynamical techniques have been explored to achieve such a large dynamic range, from Eulerian methods using regular meshes (c.f. Cen 1992 and Ryu et al. 1993) and recently with Adaptive Mesh Refinement (AMR; cf. Berger & Colella 1989; Klein, Colella & McKee 1992) to Langrangian methods like smoothed particle hydrodynamics (SPH; c.f. Eyrard 1998; Hernquist & Katz 1989; Katz, Weinberg & Hernquist 1996; Steinmetz 1996). Eulerian schemes without AMR are inadequate because of computational expense and are wasteful because high resolution is typically not required at all points in a simulation volume. SPH methods provide higher spatial resolution than regular mesh Eulerian methods such as TVD and PPM, but have poorer relative shock resolution than shock capturing methods.

N-body codes can be classified as direct, in the case of particle–particle methods and TREE methods (Barnes & Hut 1986; Hernquist 1987); expansion-based, including self-consistent field codes (e.g. Hernquist & Ostriker 1992) or grid-based, such as the particle–mesh method (Efstathiou et al. 1985); or hybrids, such as PM (Hockney & Eastwood 1981) or TPM (Xu 1995), depending on the potential solver. In cosmological simulations involving gas and dark matter it is desirable that the N-body and hydro solvers achieve similar spatial resolution. Normally, Eulerian schemes for N-body and gas are combined, such as PM + TVD code (Ryu *E-mail: xu@ucolick.org

© 1997 RAS

© Royal Astronomical Society • Provided by the NASA Astrophysics Data System
et al. 1993), and Lagrangian schemes go together such as TREESPH (Hernquist & Katz 1989). Numerically it is appropriate to employ similar algorithms for N-body and gas dynamics.

Recently, unstructured meshes have become increasingly popular in many fields, such as geophysics, seismology, structural mechanics and computational fluid dynamics. When combined with an accurate shock-capturing technique, codes employing unstructured meshes have many advantages over particle-based algorithms and Eulerian codes with regular grids, in principle. In an unstructured mesh, grid points are connected by triangles in two dimensions and tetrahedra in three dimensions. Since grid points can be placed arbitrarily, an optimal mesh can be configured for any applications. Mesh refinement can be achieved by simply adding more grid points and reconnecting the mesh. The refined mesh will have the same topology as the original one except that it will have more cells, thus mesh refinement adds no overhead to algorithms designed for unstructured meshes. Although the nodes in an unstructured mesh can be irregularly distributed, the internal data structure used to represent the grid is homogeneous as opposed to block-structured grids where block boundaries and interiors must be distinguished. As described below, only local grid operations are needed to solve equations on unstructured meshes, hence the codes can be easily parallelized.

In this paper, we describe techniques to construct unstructured meshes and algorithms to solve N-body and gas dynamic problems on these grids. We apply this code to perform cosmological N-body + hydrodynamic simulations, including various cooling and heating processes and mesh refinement. To avoid lengthy mathematical derivations of the algorithms, we put all the required formulae in the appendices.

2 NUMERICAL ALGORITHMS ON UNSTRUCTURED MESHES

2.1 Construction of unstructured meshes

Unstructured meshes are constructed from arbitrarily scattered points in n-dimensional space by Delaunay triangulation. A triangulation of n-points connects all the points with non-overlapping triangles in two-dimensional space and tetrahedra in three-dimensional space, with the vertices of triangles or tetrahedra being the given points. The mathematical definition of Delaunay triangulation can be found in text books on geometric design (O’Rourke 1994). Delaunay triangulation is dual to Voronoi diagram (O’Rourke 1994; van de Weygaert 1994). What is a Voronoi diagram? Imagine a vast forest containing a number of fire observation towers. Each ranger is responsible for extinguishing any fire closer to his tower than to any other tower. The set of all trees for which a particular ranger is responsible constitutes the ‘Voronoi polygon’ associated with his tower. The Voronoi diagram maps out the lines between these areas of responsibility. An example of Delaunay triangulation in two-dimensional space is illustrated in Fig. 1. The interior of
the circumcircle of any triangle contains no other point in the point set. Delaunay triangulation is unique provided that no \( n + 2 \) points are cospherical in \( n \)-dimensional space. There are many properties associated with Delaunay triangulation (c.f. Lawson 1986; Barth 1995), many of which will be cited in our paper without strict mathematical description. In Appendix A, we gave the essential formulas for geometric relations between a point and a simplex (triangle in 2D and tetrahedron in 3D).

Much effort has been devoted to designing algorithms to perform Delaunay triangulation. Among these, incremental insertion algorithms are of particular interest, since we will incorporate mesh refinement to enhance resolution. The methods of Bowyer (1981) and Watson (1981) are very similar. When a new point is inserted into the existing triangulation, those simplices with their circumspheres enclosing the new point are deleted and new simplices corresponding to those just deleted are added. The Bowyer-Watson algorithm is straightforward to implement and is efficient \([O(N^{1+\epsilon})]\) in \( n \)-dimensional space. Another triangulation method is the edge-swapping algorithm of Green & Sibson (1977), which was extended to 3D by Joe (1989). Edge-swapping algorithms insert a new point into the simplex that encloses it, and perform a sequence of local transformations until no further local transformations can be performed. The edge-swapping algorithm is slightly lower \([O(N_{1+\epsilon})]\), but can be adopted to handle different criteria to perform local transformations.

For cosmological simulations, we require a triangulation algorithm for a set of points in a periodic box. The Bowyer-Watson algorithm appears to have difficulties with periodic boundaries when the number of points in the box is small, because some triangles might have two vertices that are periodic images of the same point. The edge-swapping algorithm, however, can be applied to periodic volumes. Thus a combined Bowyer-Watson algorithm with edge-swapping algorithm is efficient for periodic boxes. Our triangulation algorithm uses the edge swapping method in the beginning of triangulation (typically about 1000 points), then the Bowyer-Watson method will be used for the rest of triangulation. In Fig. 2 we show an example of a two-dimensional unstructured mesh for 1000 random points in a periodic box.

Our data structures to describe the geometric connections in an unstructured mesh differ from those used previously (e.g. Kallinderis & Vijayan 1993). Two data types represent an unstructured mesh: nodes and cells. A node contains a vector to describe the position of a point and a flag to record the refinement level. A cell contains \( n + 1 \) pointers to its vertices, another \( n + 1 \) pointer to its neighbour simplices, and an integer flag to record various information about the status of the cell and its relation to its neighbours. With periodic boundaries, another two integer flags record the relative position between the simplex and its vertices and its neighbour simplices. So the memory requirement with periodic boundaries is \( n + 1 \) words for

![Figure 2](https://example.com/f2.png)

Figure 2. Example of two-dimensional unstructured mesh constructed by Delaunay triangulation. 1000 nodes are randomly scattered in a periodic box the borders of which are indicated by dashed lines.
each node and \(2n + 5\) words for each cell. This implies that the total memory required to store an unstructured mesh is about \(3 + 2 \times 9 = 21\) words per node in 2D, and \(4 + 6 \times 11 = 70\) words per node in 3D.

### 2.2 N-body algorithms on unstructured meshes

Gravitational accelerations in N-body systems can be calculated either from particle-particle methods or particle-mesh techniques. Below we introduce a new particle-mesh algorithm for unstructured grids.

The discretized Poisson’s equation on an unstructured mesh can be derived as follows. Consider node 0 as illustrated in Fig. 3. We choose the control volume for node 0 with the boundary connected by the middle points of the edges and middle points of each simplex associated with node 0, such as that indicated by the dotted lines in Fig. 3. Integrating Poisson’s equations over this control volume gives

\[
\int_{\Omega} \! dV \nabla^2 \phi = \int_{\Omega} \! dV \nabla \cdot (\rho - \rho_0) = 4\pi G (m_0 - \rho_0 V_0),
\]

where \(V_0\) and \(m_0\) are the control volume and mass of node 0, respectively. If we linearly interpolate the potential field in each simplex \(T_i\), we have \(\phi(x) = \sum w_k(x) \phi_k\), where \(w_k(x)\) is the weighting to vertex \(k\). It can be shown that \(w_k(x)\) is equal to the barycentric coordinates (equation A3) of a point located at \(x\) relative to vertex \(k\) of simplex \(T_i\) (see Appendix A for more details). Using Gauss’ theorem, the left-hand side of equation (1) can be written as,

\[
\int_{\Omega} \! dV \nabla^2 \phi = \int_{\Omega} \! dS \cdot \nabla \phi = \sum_{k=0}^{n} S_k \nabla \phi_k
\]

\[
= -\sum_{k=0}^{n} \frac{V_i}{n} \nabla w_k \cdot \sum_{k=0}^{n} \nabla w_k(\phi) \phi_k,
\]

where \(V_i\) is the volume of simplex \(T_i\) and \(n\) is the dimension of space. This result is the same as that derived using the Galerkin finite-element approximation (Barth 1995).

The discrete Poisson’s equation resulting from this procedure is a system of linear equations, \(M_{ij} \phi_j = b_i\), where \(M_{ij}\) is a symmetric sparse matrix. Since the matrix \(M\) is sparse, the conjugate gradient method can be used to solve this linear system efficiently. To guarantee convergence, we require that \(M\) be positive definite. A necessary and sufficient condition for \(M\) to be positive definite is that the triangulation is a Delaunay triangulation (Barth 1995). The convergence rate of this simple conjugate gradient method is shown in Fig. 4. The error decreases exponentially with the number of iterations, given a good initial guess of the potential. In N-body simulations, we can always use the solution in the previous step as the initial guess. Our numerical experiments indicate that the number of iterations required to achieve convergence during the next time step is typically about 20–50.

In order to solve Poisson’s equation on an unstructured mesh for the N-body problem, we must interpolate particles to the mesh. For a regular mesh, particle interpolation can be done using the cloud-in-cell (CIC) interpolation (see, for example Efstathiou et al. 1985). For an unstructured mesh, a similar procedure can be used. For each particle, we identify the cell containing the particle, and calculate the barycentric coordinate \(b_i\) of the particle inside the cell according to equation (A2). Mass is assigned to each node of the tetrahedral cell with weighting factor \(b_i\). After particle mass interpolation, the density of each node obtained by dividing its mass by its dual volume.

After solving for the gravitational potential, the acceleration on each node is calculated from the average of its control volume. For example, the acceleration at node 0 in Fig. 3 is,

\[
F_0 = \frac{1}{V_0} \int_{\Omega} \! dV \nabla \phi = \sum_{k=0}^{n} \frac{V_i}{n} \phi_k \nabla w_k .
\]

The acceleration on each particle is calculated in a similar fashion as in the particle-mesh algorithm,

\[
F_i = \sum_{k=0}^{n} w_k(\phi) F_k .
\]

The above formulation, when applied to a regular mesh, is identical to the particle-mesh algorithm with CIC interpolation.

In Fig. 5, we show the force between two equal mass particles with different separations and orientation.
Figure 4. The convergence rate of our conjugate gradient algorithm. The error decreases almost exponentially as the number of iterations.

Figure 5. The force between two particles on an unstructured mesh using 32^3 nodes. The 32^3 nodes are distributed uniformly in a periodic box. The solid line indicates the $r^{-2}$ law.
obtained using the above algorithm. The force behaves similarly to the PM method, i.e. accurate long-distance forces, but underestimates short-range force. The force is not very obtained using the above algorithm. The force behaves similarly to the PM method, which is slightly better than the PM algorithm because each node is connected with more cells in an unstructured mesh than in a regular mesh. The deviation from $r^{-2}$ law at large separations is the result of periodic boundaries.

As discussed later, mesh refinement can be performed at low cost, so we can achieve high spatial resolution using unstructured meshes for the N-body problem.

2.3 Hydrodynamics on unstructured meshes

The Euler equations are solved on unstructured meshes through the finite volume scheme (cf. Vijayan & Kallinderis 1994),

$$\frac{\partial U}{\partial t} = \frac{1}{V_0} \sum_{k \in A_k} F_{ak} \cdot S_{ak},$$

(6)

where $U \equiv \{ \rho, \rho v, E \}$ represents the fluid state and $F(U)$ is the flux vector. Before we write down the discrete form of the equations, we need to decide whether we want to store the fluid variables on the nodes or in the cells. The node representation uses less memory because the number of cells is typically 5–6 times the number of nodes in 3D. However, there are indications that the cell representations gives higher resolution than node representation (see Mavriplis 1993 for a discussion). We choose node representation to minimize memory usage. Consider node 0 in Fig. 3 with its control volume $\Omega$ illustrated in the figure. We have,

$$\frac{\partial U}{\partial t} \bigg|_0 = \frac{1}{V_0 \cdot \Omega} \sum_{k \in A_k} F_{ak} \cdot S_{ak},$$

where $A_k$ is the set of neighbouring vertices connected with node 0, $F_{ak}$ is the flux at the middle of edge 0–$k$, and $S_{ak}$ is the total surface area of the control volume related to edge 0–$k$. All we need is a method to calculate the flux in the middle of each edge accurately.

Previous approaches to solve the Euler equations have used upwind schemes of various types to calculate the flux at the edges (c.f. Barth 1995). Upwind schemes require artificial viscosity to achieve better than first order accuracy when using unstructured mesh. Here we introduce a new approach based on the gas kinetic theory of fluid dynamics.

The hydrodynamic equations (both in Euler equations and the Navier–Stokes equations) can be derived from Boltzmann’s equation through the Chapman–Enskog procedure (see, for example, Shu 1991, chapters 2 and 3). It is quite physical to derive numerical schemes for hydrodynamic equations using the gas-kinetic theory. Recently, Xu & Prendergast (1994) and Xu, Martinelli & Jameson (1995) successfully developed such numerical schemes based on the BGK model (Bhatnagar, Gross & Krook 1955) of the collisional Boltzmann’s equation,

$$\frac{\partial f}{\partial t} + u \cdot \nabla f = \frac{g - f}{\tau},$$

(7)

where $g(t, x, u)$ is the equilibrium distribution function and $\tau$ is the collisional time-scale. The BGK model accurately describes a large range of situations, from very high-density, high-temperature flows to very high Mach number (> $10^4$) flows. The solution to this equation can be written as

$$f(t, x, u) = \frac{1}{\tau} \int_0^t \frac{g(t', x', u') e^{-\frac{(t' - t)}{\tau} + e^{-\frac{\tau}{\tau}} f_0(x - ut, u)} dt'}{\tau},$$

(8)

where $x' = x - u(t - t')$ and $f_0(x, u)$ is the initial state.

The macroscopic quantities $U(t, x)$ and $F(t, x)$ are moments of the distribution function $f(t, x, u)$,

$$U = \int \psi f \, du \, d\xi,$n$$

(9)

$$F = \int \psi u f \, du \, d\xi,$n$$

(10)

where

$$\psi = \begin{bmatrix} 1 \\ u \\ \frac{1}{2}(u^2 + \xi^2) \end{bmatrix}$$

(11)

with $\xi$ representing the internal variable with $K$ degrees of freedom which will be discussed in Appendix C.

Consider an edge which connects two nodes. Without loss of generality, we assume that the two ends of the edge have coordinates ($-1/2, 0, 0$) and ($1/2, 0, 0$), and that the current time is $t = 0$ and the time step is $\Delta t$. The current position in consideration is $x = 0$. Following the treatment in Xu et al. (1995), we can expand the distributions $f_0$ and $g$ along the x-direction around the edge centre as follows,

$$f_0(x, u) = \begin{cases} g_0^L(1 + A_+^L x), & x < 0, \\
 g_0^R(1 + A_+^R x), & x > 0, \end{cases}$$

(12)

$$g(t, x, u) = \begin{cases} g_0^L \left[1 + A_{\text{gr}}^L x + B_t, & x < 0, \\
 1 + A_{\text{gr}}^R x + B_t, & x > 0. \end{cases}$$

(13)

Here $g_0$ is the equilibrium state, which is a Boltzmann distribution for hydrodynamic equations. The coefficients of $A_+, A_{\text{gr}}^L, A_{\text{gr}}^R, B$ can be expanded in velocity space as $A_+ = A_+^L \psi_0, A_+ = A_+^R \psi_0, A_{\text{gr}}^L = A_{\text{gr}}^L \psi_0, A_{\text{gr}}^R = A_{\text{gr}}^R \psi_0, B = B_0 \psi_0$, with $A_+^L, A_+^R, A_{\text{gr}}^L, A_{\text{gr}}^R, B_0$ being constants. The reason for the splitting of the right-hand side and the left-hand side has its physical basis (Xu et al. 1995), and serves as the mechanism for shock capturing, in the scheme.

Substituting the solutions to equation (8) after the integration, we get,

$$f(t, 0, u) = \left(1 - e^{-\frac{t}{\tau}}\right)g_0^L + \left[-\frac{\tau}{\tau} + (t + \tau) e^{-\frac{t}{\tau}}\right]u_0 A_{\text{gr}}^L g_0^L + (t + \tau + \tau e^{-\frac{t}{\tau}}) B_0^L e^{-\frac{t}{\tau}} f_0(0, u).$$

(14)
This distribution is used to calculate the flux function \( F(t, x) \) through moment integration (equation 10).

Notice in the above derivation that it does not matter if the gas is 1D or 3D. In this sense, the BGK gas-kinetic scheme is truly multiple dimensional without involving vector splitting which is usually used in TVD or PPM codes to generalize from 1D space to multipole dimensional space.

3 COSMOLOGICAL EQUATIONS

The dynamical equations for dark matter and gas in a comoving frame can be written as follows (cf. Peebles 1980; Cen 1992),

\[
\frac{dx}{dt} = -\frac{1}{a} \frac{\partial \phi}{\partial x},
\]

\[
\frac{d\rho}{dt} + \frac{\partial \rho v}{\partial t} + \frac{\partial \rho v}{\partial x} = 0,
\]

\[
\frac{\partial E}{\partial t} + \frac{\partial (E + P) v}{\partial x} = -2 \frac{\partial E}{\partial x} - \rho v \cdot \nabla \phi + H - \Lambda,
\]

where \( H \) is the heating function and \( \Lambda \) is the cooling function. The polytropic equation of state is normally adopted for an adiabatic gas,

\[
E = \frac{1}{2} \rho v^2 + \frac{P}{\gamma - 1}.
\]

These equations can be simplified by defining, \( dt' = a^{-2} dt \), \( v' = a v \), \( E' = a^2 E \), \( P' = a^\gamma P \) and \( \phi' = a^2 \phi \). This gives

\[
\frac{\partial \rho}{\partial t'} + \frac{\partial \rho v'}{\partial x_k} = 0,
\]

\[
\frac{\partial E'}{\partial t'} + \frac{\partial (E' + P') v'}{\partial x_k} = -\rho v_k' \frac{\partial \phi'}{\partial x_k} + a^4 (H - \Lambda),
\]

\[
E' = \frac{1}{2} \rho v'^2 + \frac{P'}{\gamma - 1}.
\]

3.1 Time integration scheme

The equations of motion for the dark matter are integrated using a time-centred second-order-accurate leapfrog algorithm. The particle positions are one half time-step ahead of their velocities.

\[
x_i^n + \frac{1}{2} = x_i^{n-1/2} + a^{-1} v'_i \Delta t,
\]

\[
v_i^{n+1} = \frac{1 - (1/2)H(t) \Delta t}{1 + (1/2)H(t) \Delta t} v_i^n + a^{-1} F_i^{n+1/2} \Delta t
\]

where \( F \equiv -\nabla \phi \).

In applying mesh refinement, we allow the system time-step based on gravity to adjust according to,

\[
\Delta t_{grav} \leq c_{grav} \min_{ij} \left[ \frac{a \delta l_{ij}}{\max(|F_i|, |F_j|)} \right],
\]

where \( \delta l_{ij} \) is the length of the edge between nodes \( i \) and \( j \) in the unstructured mesh, and \( F_i \) is the gravitational acceleration at node \( i \). The constant \( c_{grav} \) has a meaning similar to the Courant–Friedrichs–Lewy (CFL) condition in hydrodynamics. Our numerical experiments show that \( c_{grav} \approx 0.3 \) is a good choice.

When the time-step changes from \( \Delta t_i \) to \( \Delta t_{i+1} \), we adjust the positions of the particles from \( t + \Delta t_i/2 \) to \( t + \Delta t_{i+1}/2 \) using the following second-order-accurate formula,

\[
x_i(t + \Delta t_{i+1}/2) = x_i(t + \Delta t_i/2) + \frac{(\Delta t_{i+1} - \Delta t_i)}{2} \hat{x}(t)
\]

\[
+ \left( \frac{(\Delta t_{i+1}^2 - \Delta t_i^2)}{8} \right) \hat{x}(t - \Delta t_{i+1}/2).
\]

The CFL stability criterion determines the hydrodynamic time-step for the system. We use a simplified version of the CFL criterion in an unstructured mesh,

\[
\frac{\delta_{hydro}}{\min_{ij} \left[ \frac{\delta l_{ij}}{\max_{i,j} (v_i \cdot n_i + c_{ijk})} \right]} < \gamma,
\]

where \( \delta_{ij} \) is the length of edge \( i-j \), \( n_i \) is the unit vector indicating the edge direction and \( c_{ijk} \) is the sound speed. We argue that the above criterion is sufficient to satisfy the CFL stability criterion described in Barth (1995).

The gravitational terms in the cosmological hydrodynamic equations can be solved consistently in the gas-kinetic scheme by including the force term in Boltzmann's equation. However, it would be rather expensive to do so. Instead, we treat these terms as source terms resulting from an external force. The fluxes owing to gravitational acceleration are calculated as follows,

\[
\Delta^G \rho = 0,
\]

\[
\Delta^G \rho v = \frac{1}{2} \left[ (\rho (n+1) + \rho (n)) F^{(n+1/2)} - (n+1/2) \rho v \right]
\]

\[
\Delta^G E = \frac{1}{2} \left[ (\rho v (n+1) + \rho v (n)) F^{(n+1/2)} - (n+1/2) \rho v \right]
\]

where \( F \equiv -\nabla \phi \). Since the hydrodynamic quantities are synchronized with the velocity field of the dark matter, when the system time-step changes, we still need only to change...
the particle positions. For the hydrodynamic time-step, we allow for variable CFL constant from one time-step to another in order to limit the change of system time-step.

3.2 Radiative cooling

Various cooling and heating processes relevant to primordial gas have been included in the code, which includes the collisional excitation of $H^0$ and $He^+$, collisional ionization of $H^0$, $He^0$ and $He^+$, radiative recombinations of $H^+$, $He^+$ and $He^{++}$, dielectronic recombinations of $He^+$, bremsstrahlung emission, inverse Compton cooling of the microwave background, and photoionization of $H^0$, $He^0$ and $He^+$ due to UV background. The rates of these processes can be found in Black (1981), Cen (1992), Katz et al. (1996), Abel et al. (1996) and Ikeuchi & Ostriker (1986). Since the cooling/heating time can be very short compared with the hydrodynamic time, we have to be very careful with time integration of the energy equation when cooling processes are turned on. In our implementation, we integrated the cooling function with adjustable time-steps within one system time-step. The variable-step, fifth-order-accurate Runge-Kutta integrator described in Press et al. (1992) is used to integrate the following equation,

$$\frac{d\mu}{dt} = \Delta t + \mathcal{H} - \Lambda,$$

where $\mu$ is the thermal energy, $\Delta t$ is the thermal energy change owing to gravity and hydrodynamics, $\Delta t$ is the system time-step and $\Lambda$ is the cooling function. This equation is integrated from 0 to $\Delta t$ using many time-steps depending on the cooling time-scale. Our numerical experiments show that sometimes about $10^4$ time-steps are required within one dynamic time-step $\Delta t$.

3.3 Mesh refinement algorithms

For an unstructured mesh, cells can be refined arbitrarily. Vijayan & Kallinderis (1994) discuss strategies for cell division. One can put a node in the centre of an edge, in the middle of a face, in the middle of a cell, or a combination of all of them.

In cosmological simulations, we want to resolve forming structures. For N-body problems, we use a mass criterion $m_*$ to determine mesh refinement. After we interpolate particle data to mesh nodes, each node carries a mass. For each face of a cell in the unstructured mesh, we put a refining node at the centre of the face if the linearly interpolated mass at the centre is above $m_*$. For hydrodynamic problems, following the criteria for galaxy formation in Cen & Ostriker (1993), we put a refining node in the middle of a cell if the gas in this cell, (a) is contracting, $\nabla \cdot v < 0$, and (b) has a mass greater than the Jeans' mass, $m_J > m_\ast$. When cooling processes are included we also require the cooling time to be shorter than the dynamical time, $t_{cool} < t_{dyn}$.

4 CODE TESTS AND PERFORMANCE

4.1 Testing the 1D gas-kinetic scheme

In Fig. 6, we show the results of a Lax shock tube test. The initial conditions for this test are $U=(0.445, 0.311, 8.928)$ for $x < 0$ and $U=(0.5, 0, 1.4275)$ for $x > 0$. The result is at $t=0.15$. The contact discontinuity is resolved with about two cells, the rarefaction shock was sharply captured with about two cells and no post-shock oscillation is observed.

One test that is closely related to structure formation in cosmology is described in Ryu et al. (1993) with the following initial conditions: $\rho=0$, $v(x)=\sin(2\pi x)/2\pi$, $p=10^4$, and periodic boundary condition for $0 \leq x \leq 1$. Our results at $t=3$ are presented in Fig. 7. We notice that the BGK gas-kinetic scheme can successfully resolve features within two cells without any artificial viscosity or adjustment for the tem-
perature term owing to high Mach number. Our scheme successfully reproduces the density caustic, the saw-shape velocity field, and segmented pressure field with small oscillation. The Mach number involved in this test is much higher (~10^4) than that in usual shock tube tests. This result demonstrates that the BGK gas-kinetic scheme is very robust in high Mach number situations.

4.2 Cosmological N-body simulations

We construct the initial conditions for the cold dark matter (CDM) model using the Zel'dovich approximation (cf. Efstathiou et al. 1985). The simulation we are showing has CDM power spectrum with \( \sigma_8 = 0.65, \Omega_0 = 1, h = 0.5 \) the box size is 64 Mpc. Initially, particles are almost uniformly distributed. As the system evolves, structure forms as a result of gravitational clustering. More and more massive objects form as time passes. We show our results with a 32^3 uniform mesh in Fig. 8, and results with mesh refinement from a 32^3 uniform mesh in Fig. 9. The final mesh nodes are shown in Fig. 10, which indicates that our mesh refinement traces the particle distribution very well. Here, the mesh refinement is performed on the faces of tetrahedra. A new node is put in the middle of a face if the mass on all the three nodes is above a certain value. In this test case, the critical mass is taken to be \( 5m_i \), where \( m_i \) is the mass for each particle.

Visually, we can already see the great improvement of the resolution with mesh refinement. The two-body correlation function \( \xi(r) \) (see Peebles 1980 for a definition) for two simulations with and without mesh refinement is shown in Fig. 11. The resolution improvement of the simulation with mesh refinement over the simulation without mesh refinement is well above a factor of 10, while the running time between the two simulations for each time-step is less than a factor of 2.

4.3 Cosmological N-body + gas simulations

Recently Frenk et al. (in preparation) have proposed a comparison between different cosmological hydrodynamic codes. They set up a constrained initial condition for a CDM model with \( \sigma_8 = 0.65, \Omega_0 = 1, \Omega_b = 0.1, h = 0.5 \) and box size of 64 Mpc. The initial conditions for the results described below are generated from their density field using the Zel'dovich approximation (Efstathiou et al. 1985). Readers can compare some of our results to others presented in their paper (Frenk et al., in preparation) with the same initial conditions. All the tests shown below are obtained with 32^3 particles, 32^3 raw mesh with mesh refinement and the cooling/heating turned off.

In Fig. 12, we show the density and temperature contours of a slice in the simulation. The density field shows some

![Figure 8. Result of a pure N-body unstructured mesh simulation without mesh refinement. All the 32^3 particles are projected in the X–Y plane. The raw mesh is a 32^3 uniform mesh.](https://academic.oup.com/mnras/article-abstract/288/4/903/974684/2844193076/664/bj4364)

© 1997 RAS, MNRAS 288, 903–919

© Royal Astronomical Society • Provided by the NASA Astrophysics Data System
Figure 9. Result of a pure N-body unstructured mesh simulation with mesh refinement. All the 32^3 particles are projected in the X–Y plane. The raw mesh is a 32^3 uniform mesh. Refinement is done by a mass criterion $m_c = 5m_i$, where $m_i$ is the mass of each particle.

Figure 10. Distribution of the mesh nodes projected to the X–Y plane. The particle distribution of this simulation is shown in Fig. 9.
Hydrodynamic and N-body schemes

Figure 11. The two-point correlation function $\xi(r)$ for two dark matter only simulations. The solid line is for the simulation with mesh refinement, and the dashed line is without mesh refinement. In both simulations, $32^3$ particles are used on a raw mesh of $32^3$ nodes. The cell size for the raw mesh is 1 Mpc $h^{-1}$.

Figure 12. Panel (a) shows the density contours of a slice in the simulation. The contour levels are $\rho/\langle\rho\rangle = 10^k$ with $k=0, 1, \ldots$ Panel (b) shows the temperature contours of the same slice with contour levels $T = 10^k + 0.3K$ with $k=0, 1, 2, \ldots$

tracer of filamentary structure, and the temperature field is almost isothermal in high-density regions. In Fig. 13, we show the density profiles of dark matter and gas scaled to their corresponding average density. As expected, the gas traces dark matter in the outer part of the cluster, and has lower relative density than the dark matter in the central region. In Fig. 14, we show the fraction of mass contours of baryonic matter with various density and temperature. This figure summarizes the thermal state of the intergalactic medium. From this figure we know that most of the baryonic matter stays at the average density and in a temperature range of $10^4 - 10^6$ K. Only a small fraction of the baryonic matter is in high-density regions, and that at high-density regions remains at high temperatures ($\sim 10^7$ K). The material at underdense regions is cold ($T \lesssim 10^5$ K).

When the heating and cooling processes are turned on, they change the thermal state of the intergalactic medium dramatically. One important feature is that the gas in a high-density region can actually stay very cold (see fig. 3 in Katz, Weinberg & Hernquist 1996). Our simulations with heating and cooling processes turned on actually have some fraction of gas in these states.

5 SUMMARY AND DISCUSSION

In this paper, we described a new cosmological N-body + gas dynamics code based on algorithms for an adaptive, unstructured mesh. The novel elements of this code are: (1) the mesh construction; (2) solving N-body systems; (3) solving hydrodynamic equations; (4) time-step estimation and time integration; (5) mesh refinement; and (6) relevant heating and cooling processes for primordial gas.

The mesh construction with periodic boundary conditions is performed using a combined Bowyer–Watson algorithm and local transformation algorithm. This initial mesh for cosmological simulations is a uniform, staggered mesh. When some refining grids are required, new grid points are added to the mesh structure through the incremental Bowyer–Watson algorithm, which modifies the previous mesh structure slightly. The incorporation of mesh refinement in unstructured mesh gives one way to achieve high spatial resolution at a relatively low cost. For best results, a good
Figure 13. Density profiles of dark matter and gas: density is measured relative to their average density. Solid curve shows the dark matter profile, and the dashed curve shows the gas profile.

Figure 14. Mass contour of gas at different density ($\rho$) and temperature ($T$). The total mass in the box is assumed to be 1. The contour levels are in $10^i$, $i = -4, -8$. $\rho_0$ is the average gas density.
refinement criterion is essential. In general, refinement criteria can be derived both on physical and numerical bases. Poisson's equation is discretized on an unstructured mesh and solved using the conjugate gradient method. Particles are interpolated to the mesh nodes using linear interpolation. The resulting N-body algorithm is similar to the particle–mesh method with CIC interpolation. Because each node in an unstructured mesh is associated with cells, the new N-body algorithm has slightly higher force resolution than the PM algorithm.

We solve Euler's equations using the finite-volume method. Flux functions are calculated using the BGK gas-kinetic scheme. The gas-kinetic scheme constructs a time-dependent distribution in the middle of an edge and calculates flux functions by moments of the distribution function. This scheme provides high resolution for shock capturing and is very stable for high Mach number flows.

The new cosmological code solves dark matter and gas dynamics with the same resolution. Our tests demonstrate that this code can provide high spatial resolution by mesh refinement. We include relevant cooling and heating processes for the primordial gas to simulate the evolution of intergalactic medium accurately.

This cosmological code takes 33.0 s for one time-step when both gravity and hydrodynamics are turned on with 43,917 nodes and 32,768 particles. It takes only 10.5 per time-step when only the N-body part is used with the same number of nodes. The machine used for this timing is an SGI Power Challenge machine with 196-MHz R10000 processors. Notice that this timing will not change even if the particles are strongly clustered.

Since the internal data structures for an unstructured mesh are homogeneous, unstructured mesh codes can be easily parallelized. The difficult part of parallelization is the Delaunay triangulation procedure. A parallel version of the code is being developed using the portable Message Passing Interface (MPI) library functions.

ACKNOWLEDGMENTS

The author thanks Michael Norman for suggesting the investigation of unstructured mesh schemes, and Timothy J. Barth for kindly providing many references and some discussions. I greatly appreciate numerous detailed discussions with Kun Xu about the gas-kinetic scheme and sharing opinions about computational fluid dynamics. It is a great pleasure to thank Lars Hernquist and Jeremiah P. Ostriker for their support and encouragement. This work is supported by the NSF grant ASC 93-18185 rewarded to the Grand Challenge Cosmology Consortium.

APPENDIX A: GEOMETRIC RELATIONS FOR UNSTRUCTURED MESHES

For unstructured meshes, we need to determine the relation between a point and a simplex (triangle or tetrahedron), and also some geometric quantities, such as the volume of a simplex and the surface area of its faces. In this appendix, we give the formulas to calculate these quantities generally in n-dimensional space.

Let $p_1, p_2, \ldots, p_{n+1}$ be $n+1$ distinct points in $n$-dimensional space. The $n$-dimensional volume of the simplex $T$ with vertices $p_1, \ldots, p_{n+1}$ is given by

REFERENCES

Abel T., Anninos P., Zhang Y., Norman M. L., 1996, astro-ph/9608040
Barthes J., Hut P., 1986, Nat, 324, 446
Barth T. J., 1995, Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations, VKI Lecture series
Berger M., Colella P., 1989, J. Comp. Phys., 82, 64
Bhatnagar P., Gross E., Krook M., 1954, Phys. Rev., 94, 511
Black J., 1981, MNRAS, 197, 553
Bouyer A., 1981, Comp. J., 24, 162
Cen R., 1992, ApJS, 78, 341
Cen R., Ostriker J., 1993, ApJ, 417, 404
Efstathiou G., Davis M., Frenk C. S., White S. D. M., 1985, ApJS, 57, 241
Evard A. E., 1988, MNRAS, 235, 911
Fortune S., van Wyk C. J., 1993, in Proc. 9th Annual Symp. Computational Geometry. ACM, New York, p. 163
Green P. J., Sibson R., 1997, Comp. J., 21, 168
Hernquist L., 1987, ApJS, 64, 715
Hernquist L., Katz N., 1989, ApJS, 70, 419
Hernquist L., Ostriker J. P., 1992, ApJ, 386, 375
Hockney R. W., Eastwood J. W., 1981, Computer Simulation Using Particles. McGraw Hill, New York
Ikeuchi S., Ostriker J. P., 1986, ApJ, 301, 522
Jameson A., 1995, Int. J. Numerical Methods in Fluids, 20, 743
Joe B., 1989, SIAM J. Sci. Stat. Comp., 10, 718
Kallinderis Y., Vijayan P., 1993, AIAA Journal, 31, 1440
Karz N., Weinberg D. H., Hernquist L., 1996, ApJS, 105, 19
Klein R., Colella P., McKee C., 1992, in Dannevik W., Buckingham A., Leith C., eds, Advances in Compressible Turbulent Mixing. GPO, Washington DC, p. 452
Lawson C. L., 1986, Computer Aided Geometric Design, 3, 231
Mavriplis D. J., 1993, in Mapolitano M., Sabetta F., eds, Unstructured Mesh Algorithms for Aerodynamic Calculations, Lecture notes in physics. Springer-Verlag, Berlin, p. 57
O'Rourke J., 1994, Computational Geometry in C. Cambridge Univ. Press, Cambridge
Peebles P. J. E., 1980, The Large Scale Structure of the Universe. Princeton Univ. Press, Princeton
Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 1992, Numerical Recipes in C. Cambridge Univ. Press, Cambridge
Ryu D., Ostriker J. P., Kang H., Cen R., 1993, ApJ, 414, 1
Shu F., 1991, The Physics of Astrophysics: Volume II, Gas Dynamics. University Science Books, Mill Valley, CA
Steinmetz M., 1996, MNRAS, 278, 1005
van de Weygaert R., 1994, A&A, 283, 361
Vijayan P., Kallinderis Y., 1994, J. Comp. Phys., 113, 249
Watson D., 1981, Comp. J., 24, 167
Xu G., 1995, ApJS, 98, 355
Xu K., 1993, PhD thesis, Columbia Univ.
Xu K., Martinelli L., Jameson A., 1995, J. Comp. Phys., 120, 48
Xu K., Prendergast K., 1994, J. Comp. Phys., 114, 9
Let \( q \) be an arbitrary point in \( n \)-space. If the simplex \( T \) with vertices \( p_1, \ldots, p_{n+1} \) is non-degenerate, i.e., if \( \text{Vol}(p_1, \ldots, p_{n+1}) \neq 0 \), the numbers, \( b_1, b_2, \ldots, b_{n+1} \), satisfying

\[
\begin{pmatrix}
  1 & 1 & \ldots & 1 \\
  p_1 & p_2 & \ldots & p_{n+1} \\
  b_1 & b_2 & \vdots & b_{n+1}
\end{pmatrix} = \begin{pmatrix}
  1 \\
  q 
\end{pmatrix}
\]  

(A2)

are called \textit{barycentric coordinates} of a point \( q \) relative to simplex \( T \). It can be shown that

\[
b_k = \frac{\text{Det}(p_1, \ldots, q, \ldots, p_{n+1})}{\text{Det}(p_1, \ldots, p_{n+1})}.
\]

(A3)

Obviously, \( b_k \) is a linear function of \( q \). Thus, \( b_k \) indicates the position of \( q \) relative to the hyperplane \( H_k \) containing the facet of simplex \( T \) opposite to vertex \( p_k \). \( b_k = 0 \) when \( q \) is in \( H_k \), \( b_k > 0 \) when \( q \) is on the same side of \( H_k \) from \( p_k \) and \( b_k < 0 \) when \( q \) is inside simplex \( T \) if and only if all \( b_k \geq 0 \). The surface area vector \( S_k \) of hyperplane \( H_k \) with its direction pointing away from \( p_k \) is

\[
S_k = -\text{Vol}(T) b_k(x).
\]

(A4)

It can be shown that the integration of function \( f(x_1, \ldots, x_n) \) over the volume of a simplex \( T \) can be expressed as

\[
\int f(x, y, z) \, d^nx = \text{Vol}(T) n! \int_0^1 db_1 \int_0^{1-b_1} db_2 \cdots \int_0^{1-b_1 \cdots - b_{n-1}} db_n f(x_1, \ldots, x_n).
\]

(A5)

If a simplex \( T \) is non-degenerate, it has a unique circumsphere \( S \). Given an arbitrary point \( q \) in \( n \)-space, we can determine if \( q \) is inside, outside or on the sphere \( S \) by the following function,

\[
\text{InSphere}(q, T) = \frac{\text{Det} \begin{pmatrix}
  1 & 1 & \ldots & 1 \\
  w_q & w_1 & \ldots & w_{n+1} \\
  q & p_1 & \ldots & p_{n+1}
\end{pmatrix}}{\text{Det} \begin{pmatrix}
  1 & 1 & \ldots & 1 \\
  p_1 & p_2 & \ldots & p_{n+1}
\end{pmatrix}},
\]

(A6)

where \( w_p = \sum_{i=1}^{n} x_i z_i \). \text{InSphere}(q, T) > 0 \) when \( q \) is inside \( S \), \text{InSphere}(q, T) = 0 \) when \( q \) is on \( S \) and \text{InSphere}(q, T) < 0 \) when \( q \) is outside \( S \). When calculating the value of \( \text{InSphere}(q, T) \), we should be aware of round-off errors (Barth 1995), because the result of triangulation could be wrong owing to floating-point inaccuracies. To avoid the problems caused by floating-point round-off errors, we calculate the above \( \text{InSphere}(T, q) \) function using the following formula instead,

\[
\text{InSphere}(T, q) = \frac{\text{Det} \begin{pmatrix}
  w'_1 & w'_2 & \ldots & w'_{n+1} \\
  p'_1 & p'_2 & \ldots & p'_{n+1}
\end{pmatrix}}{\text{Det} \begin{pmatrix}
  1 & 1 & \ldots & 1 \\
  p'_1 & p'_2 & \ldots & p'_{n+1}
\end{pmatrix}} \min_{k=1,2,\ldots,n+1} (w'_k)
\]

(A7)

where \( p'_k = p_k - q \) and \( w'_k = \sum_{i=1}^{n} x_i z_i \). The value of the above function is compared with a small number \( \epsilon \), instead of 0, to determine the result of the sphere test. For single-precision floating-point operations, we use \( \epsilon = 10^{-4} \). Our numerical experiments show that the above estimate is sufficient to give the correct Delaunay triangulation. Another way to avoid the round-off error is to use exact redundant expression calculation (see Barth 1995 for more discussion). However, there are a lot of extra calculations related to the exact redundant expression (Fortune & van Wyk 1993).
APPENDIX B: COEFFICIENT CALCULATION IN BGK FORMALISM

In this appendix, we will give detailed formulas to calculate the coefficients $A_L^\beta, A_R^\beta, A^{GR}_\beta, B_\beta$, and other quantities for the BGK scheme described earlier in the paper.

At the beginning of each time-step, we know the fluid state at the two ends of each edge $U_L$ and $U_R$. The interpolated fluid states in the middle of the edge, $U_{\ell,0}^L$ and $U_{\ell,0}^R$, which are interpolated from left side and right side respectively, can be constructed from the SLIP (Symmetric Limited Positive) formulation (Jameson 1995), which is derived from the local extremum diminishing principle. The constructed fluid state can be expressed as

$$
\hat{U}_L^\ell = U_L^\ell + \frac{1}{2} e_L^\ell, \quad \text{and} \quad \hat{U}_R^\ell = U_R^\ell + \frac{1}{2} e_R^\ell,
$$

where $e_{L,R} = L(\Delta U_{n+1/2}, \Delta U_{n-1/2})$ is the limited average, $L(u,v)$ is a limiter, and $\Delta U_{n+1/2} = U_{n+1} - U_n$. An example is the van Leer limiter, $L(u,v) = \frac{2uv}{u+v}$ when $uv > 0$, and $L(u,v) = 0$ otherwise. The equilibrium distribution functions $g^L_\beta$ and $g^R_\beta$ are constructed from $\hat{U}_L^\ell$ and $\hat{U}_R^\ell$ respectively.

The macroscopic quantities are moments of distributions. We have $\hat{U}_L^\ell (\varepsilon = -1/2) = U_L^L$, and after some algebra, we get the solutions of the coefficients $A_L^\beta$ and $A_R^\beta$,

$$
\langle \psi \psi \rangle^L_\beta A_L^\beta = e_L^\ell \quad \text{and} \quad \langle \psi \psi \rangle^R_\beta A_R^\beta = e_R^\ell.
$$

The notation $\langle \ldots \rangle$ will be defined later in Appendix C.

The constructed fluid state at the middle of an edge $\hat{U}_G^\ell$ is defined to be,

$$
\hat{U}_G^\ell = \int_{u_{a,0}} \psi \, g^\ell_G \, du \, dx + \int_{u_{b,0}} \psi \, g^\ell_G \, du \, dx.
$$

The equilibrium distribution $g^\ell_G$ is constructed from $\hat{U}_G^\ell$. Taking the limit $\varepsilon \to 0$, we have $\hat{U}_R^\ell (\varepsilon = 1/2) = U_R^R$, and $\hat{U}_L^\ell (\varepsilon = -1/2) = U_L^L$. This gives the solutions of coefficients $A^{GR}_\beta$.

The distribution functions $f(t,x,u)$ and $g(t,x,u)$ must be compatible with each other. Conservation laws give the following compatibility condition,

$$
\int \psi (f-g) \, du \, dx = 0.
$$

Applying the integrated solution of $f(t,x,u)$ (equation 14) to the above compatibility equation and integrate over the whole time-step $T$, we have,

$$
\int_0^T \int_0^T \psi \, dt \, dx = \int_0^T \int_0^T \int_0^T \psi \, f(t,0,u) \, du \, dx \, dt
$$

These equations give the solutions to the coefficients $B_\beta$.

For convenience, we give the formula to calculate the time integrated flux functions,

$$
\int_0^T \int_0^T \psi \, f(t,0,u) \, du \, dx \, dt
$$

© 1997 RAS, MNRAS 288, 903-919

© Royal Astronomical Society • Provided by the NASA Astrophysics Data System
The collision time \( \tau \) can be derived from classical statistical mechanics to be the mean free path divided by the rms velocity of the atoms. We use the following formula to estimate \( \tau \),

\[
\tau = C_1 \frac{\sqrt{\lambda}}{\rho} + C_2 \frac{|p^b - p^\mu|}{|p^b + p^\mu|},
\]

where \( \Delta T \) is the time step and \( C_1, C_2 \) are constants. We take \( C_1 = 0.01 \) and \( C_2 = 1 \) in our calculations. The results are not sensitive to the choice of the actual values of \( C_1, C_2 \).

**APPENDIX C: VELOCITY MOMENTS**

We define the moment of a quantity \( w \) of the equilibrium state \( g_0 \) as the following,

\[
\langle w \rangle = \frac{1}{\rho} \int w g_0 \, du \, d\xi = \frac{1}{\rho} \int w D \, e^{-\lambda(|u-v|^2 + \xi^2)} \, du \, d\xi,
\]

where \( V \) is the macroscopic velocity, \( \lambda = \rho/2p \) is the gas temperature, and \( D = \rho (\lambda/\pi)^{K+\gamma} \) is the normalization factor. Here \( K \) is the degree of the internal variable \( \xi \) and \( N \) is the space dimension (Xu 1993). For a polytropic gas, classical statistical mechanics gives \( \gamma = (n+2)/n \), where \( n \) is the total number of effective degrees of freedom of the molecule: thus a monoatomic gas has \( n = 3, \gamma = 5/3 \), and a diatomic gas with two rotational degrees of freedom has \( n = 5, \gamma = 7/5 \). For a flow in \( N \)-dimensional space, we have \( K = n - N = 2(\gamma - 1) - N \).

Following the above definition, we obtain the iterative relation,

\[
\langle u^{n+2} \rangle = V \langle u^{n+1} \rangle + \frac{n+1}{2\lambda} \langle u^n \rangle,
\]

and the following specific values of moments,

\[
\langle u^0 \rangle = 1,
\]

\[
\langle u^1 \rangle = V,
\]

\[
\langle \xi^2 \rangle = \frac{K}{2\lambda},
\]

\[
\langle \xi^4 \rangle = \frac{K(K+2)}{4\lambda^2},
\]

\[
\langle \xi^6 \rangle = \frac{K(K+2)(K+4)}{8\lambda^3}.
\]

For velocity moments involving integration over half of the velocity space, the above iterative relation (equation C2) still holds true, except for the following first few moments,

\[
\langle u^0 \rangle_{u > 0} = \frac{1}{2} \text{erfc}(\sqrt{\lambda} V)
\]

\[
\langle u^1 \rangle_{u > 0} = V \frac{1}{2} \text{erfc}(\sqrt{\lambda} V) + \frac{e^{-\mu^2}}{2\sqrt{\pi} \lambda}
\]

\[
\langle u^0 \rangle_{u < 0} = \frac{1}{2} \text{erfc}(\sqrt{\lambda} V)
\]

\[
\langle u^1 \rangle_{u < 0} = V \frac{1}{2} \text{erfc}(\sqrt{\lambda} V) - \frac{e^{-\mu^2}}{2\sqrt{\pi} \lambda}.
\]
The moments of $u^n\psi_s$ and $u^n\psi_p$ can be derived from the moments of $u^n$ and $\xi^n$. We explicitly write them out for reference.

\[
\langle u^n \psi_s \rangle \equiv \frac{1}{\rho} \int u^n \psi_s D e^{-\frac{(u^n)^2 + \xi^n^2}{2}} \, d\Xi = \begin{bmatrix}
\langle u^n \rangle \\
\langle u^{n+1} \rangle \\
\frac{1}{2} (\langle u^{n+2} \rangle + \langle u^n \rangle \langle \xi^n \rangle)
\end{bmatrix},
\]

(C12)

\[
\langle u^n \psi_p \rangle \equiv \frac{1}{\rho} \int u^n \psi_p H e^{-\frac{(u^n)^2 + \xi^n^2}{2}} \, d\Xi = \begin{bmatrix}
\langle u^n \rangle \\
\langle u^{n+1} \rangle \\
\langle u^{n+2} \rangle \\
\frac{1}{2} (\langle u^{n+2} \rangle + \langle u^n \rangle \langle \xi^n \rangle)
\end{bmatrix}.
\]

(C13)

For moments of $u^n\psi_s$ and $u^n\psi_p$, integrating over half velocity space, the above expressions are still good except that one must be aware that $\langle u^n \rangle$ may not be equal to 1 in the above formulae.