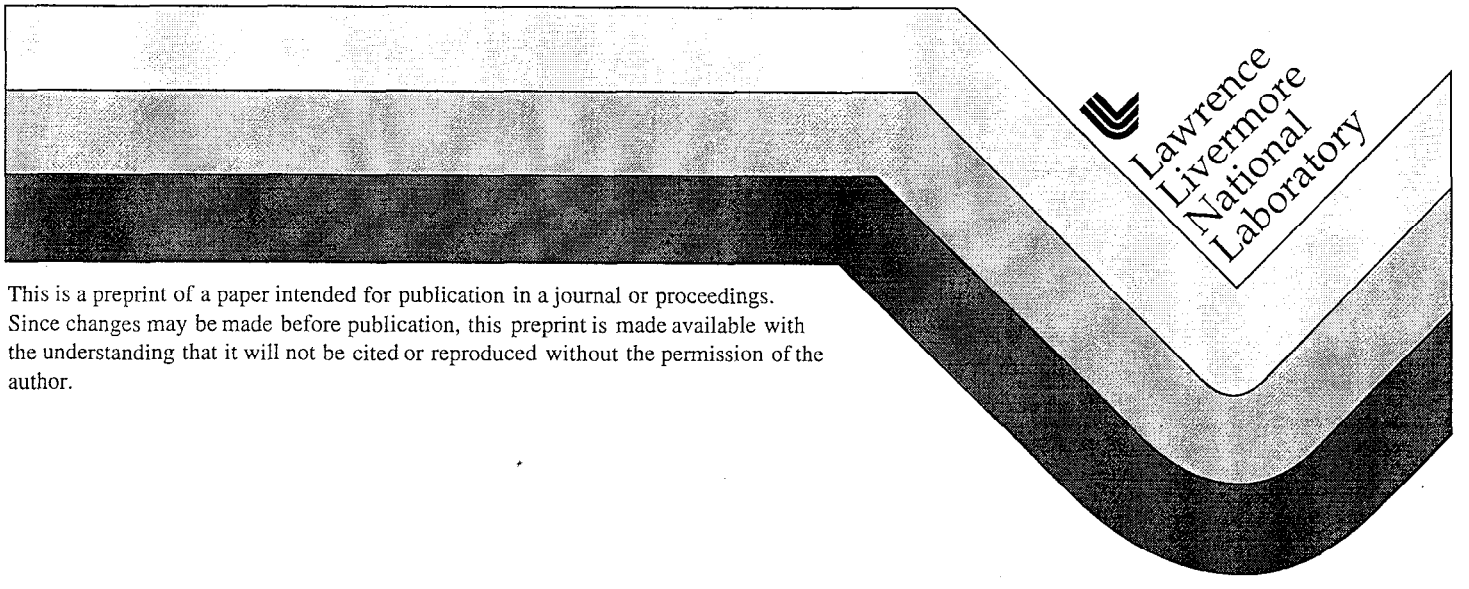
UCRL-JC-132987
PREPRINT

3D Unstructured Mesh ALE Hydrodynamics with the Upwind Discontinuous Galerkin Method

M. K. Prasad, J. L. Milovich, A. I. Shestakov,
D. S. Kershaw, M. J. Shaw

This paper was prepared for submittal to the
International Symposium on Discontinuous Galerkin Methods
Newport, RI
May 24-26, 1999

May 7, 1999



3D Unstructured Mesh ALE Hydrodynamics with the Upwind Discontinuous Galerkin Method *

Manoj K. Prasad, Jose L. Milovich, Aleksei I. Shestakov, David S. Kershaw, and Michael J. Shaw

Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

Abstract. We describe a numerical scheme to solve 3D Arbitrary Lagrangian-Eulerian (ALE) hydrodynamics on an unstructured mesh using a discontinuous Galerkin method (DGM) and an explicit Runge-Kutta time discretization. Upwinding is achieved through Roe's linearized Riemann solver with the Harten-Hyman entropy fix. For stabilization, a 3D quadratic programming generalization of van Leer's 1D minmod slope limiter is used along with a Lapidus type artificial viscosity. This DGM scheme has been tested on a variety of hydrodynamic test problems and appears to be robust making it the basis for the integrated 3D inertial confinement fusion modeling code ICF3D. For efficient code development, we use C++ object oriented programming to easily separate the complexities of an unstructured mesh from the basic physics modules. ICF3D is fully parallelized using domain decomposition and the MPI message passing library. It is fully portable. It runs on uniprocessor workstations and massively parallel platforms with distributed and shared memory.

1 The ALE Hydrodynamics Equations

The motion of a compressible fluid is described by Euler's equations along with an equation of state (EOS). In an ALE code the computational mesh x_i, where $i = 1, 2, 3$ describes the 3D space, can move in time t:

$$x_i = x_i(x_i^0, t),\ x_i(t=0) = x_i^0,\ \frac{\partial x_i}{\partial t} = V_i^g$$

where x_i^0 are the "Lagrangian" coordinates and V_i^g is an arbitrarily specified grid velocity. Euler's equations stated in conservation form, follow the time evolution of ρ (mass density), ρv_i (momentum density) and ρE (total energy density). In general, the fluid can be subjected to a body force per unit mass G_i (such as gravitational acceleration). In the Lagrangian frame the fluid equations are (in what follows summation over repeated indices is assumed)

$$\frac{\partial A_\alpha^g}{\partial t} + \frac{\partial F_{\alpha i}^g}{\partial x_i^0} = S_\alpha^g$$

* Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

where A_α^g are the conservative state variables with corresponding fluxes $F_{\alpha i}^g$ and source terms S_α^g (α runs from 1 to 5):

$$A_\alpha^g \equiv J^g A_\alpha,\ F_{\alpha i}^g \equiv F_{\alpha j}\eta_{ji},\ S_\alpha^g \equiv J^g S_\alpha$$

$$A_\alpha \equiv \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho E \end{pmatrix},\ F_{\alpha j} \equiv \begin{pmatrix} \rho(v_j - V_j^g) \\ \rho v_1(v_j - V_j^g) + P\delta_{j1} \\ \rho v_2(v_j - V_j^g) + P\delta_{j2} \\ \rho v_3(v_j - V_j^g) + P\delta_{j3} \\ \rho E(v_j - V_j^g) + Pv_j \end{pmatrix},\ S_\alpha \equiv \begin{pmatrix} 0 \\ \rho G_1 \\ \rho G_2 \\ \rho G_3 \\ \rho v_j G_j \end{pmatrix}$$

$$J_{ij}^g \equiv \frac{\partial x_j}{\partial x_i^0}, \eta_{ji} \equiv J^g J_{ji}^{g\,-1} = \frac{1}{2}\epsilon_{jkl}\epsilon_{imn}\frac{\partial x_k}{\partial x_m^0}\frac{\partial x_l}{\partial x_n^0}, J^g \equiv |J_{ij}^g| = \frac{1}{3}J_{ij}\eta_{ji}$$

where $E \equiv I + v_i v_i/2$ is the total energy per unit mass, $I \equiv I(\rho, P)$ is the internal energy per unit mass and P is the pressure. The EOS gives the explicit form for $I(\rho, P)$.

The choice $V_i^g = 0$ leads to an "Eulerian" code where the mesh is fixed in time while $V_i^g = v_i$ leads to a "Lagrangian" code where the mesh follows the fluid. Lagrangian codes have two basic advantages over the Eulerian version. First, material interfaces are exactly resolved. Secondly, the implementation of boundary conditions (specified normal velocity or pressure) is much simpler compared to the moving boundary problems inherent to the Eulerian codes.

2 The DGM Solution of the ALE Equations

We discretize the problem domain into an arbitrary and in general unstructured set of 3D linear finite elements (tetrahedrons, pyramids, prisms, and hexahedrons). Within each element K we use a piecewise (tri)linear approximation for the coordinates x_i, the body force per unit mass G_i, and the "primitive" variables $B_\alpha \equiv (\rho, v_1, v_2, v_3, P)$:

$$\{x_i(\mathbf{x^0}, t), G_i(\mathbf{x^0}, t), B_\alpha(\mathbf{x^0}, t)\} \approx \sum_{n=1}^{n_v} \{x_{in}(t), G_{in}(t), B_{\alpha n}(t)\}\phi_n(\xi)$$

where $\phi_n(\xi)$ are the 3D finite element (tri)linear basis functions which are equal to 1 at node n and 0 at all other nodes and $n_v = (4, 5, 6, 8)$ is the number of nodes in element K=(tets,pyramids,prisms,hexes). We are using isoparametric elements and ξ are the 3D isoparametric reference coordinates. It is important to note that because we are using a piecewise linear approximation for B_α there is no requirement of continuity across faces between neighboring elements, i.e. the name "discontinuous" finite elements. However we do require the coordinates x_i to be continuous across faces

To determine the vertex values $B_{\alpha n}(t)$ we use a DGM whereby we take moments of the ALE hydro equations with the n_v basis functions $\phi_n(\xi)$ in

each element K (the DGM equations):

$$\partial_t \int_K \phi_n(\xi) A_\alpha d^3x = \int_K \phi_n(\xi) S_\alpha d^3x + \int_K F_{\alpha i} \frac{\partial \phi_n(\xi)}{\partial x_i} d^3x - \int_{\partial K} N_i F_{\alpha i} \phi_n(\xi) d\Gamma$$

where $d^3x \equiv J^g d^3x^0$, we have integrated the flux term by parts, and ∂K denotes the surface of element K with outward unit length normal N_i. An essential aspect of our discretization is that it manifestly preserves the conservation properties of the continuum field equations since $\sum_{n=1}^{n_v} \phi_n(\xi) = 1$. All integrals on the right hand side of the DGM equation are computed numerically using Gaussian quadrature. The surface integral poses a problem as to what to use for $N_i F_{\alpha i}$ since the primitive variables are in general discontinuous across a face. We resolve this ambiguity by interpreting each point on the face as a 1D Riemann initial-value (shock tube) problem. We then solve this Riemann problem by Roe's characteristic decomposition along with Roe averaging generalized to arbitrary EOS:

$$N_i F^{Roe}_{\alpha i} \equiv \frac{1}{2}\{N_i F^-_{\alpha i} + N_i F^+_{\alpha i} + [R^* sign(\Lambda^*) R^{*-1}]_{\alpha\beta}[N_i F^-_{\beta i} - N_i F^+_{\beta i}]\}$$

At each point on a face of an element we have two values of the normal component of the flux: $N_i F^+_{\alpha i}$ and $N_i F^-_{\alpha i}$ corresponding to the two sides of the face, $+$ denoting the side from which the outward pointing normal N_i emanates. The Roe average state (denoted with $*$) of the $+$ and $-$ states is defined through the following equations:

$$N_i F^-_{\alpha i} - N_i F^+_{\alpha i} \equiv J^*_{\alpha\beta}(A^-_\beta - A^+_\beta),\ J^* \equiv R^* \Lambda^* R^{*-1},\ \Lambda^*_{\alpha\beta} \equiv \lambda^*_\alpha \delta_{\alpha\beta}$$

where $\lambda_\alpha(\mathbf{v}) \equiv \{\lambda_p, \lambda_m, \lambda_o, \lambda_o, \lambda_o\}, \lambda_o = N_i(v_i - V^g_i), \lambda_p = \lambda_o + c, \lambda_m = \lambda_o - c, \lambda^*_\alpha \equiv \lambda_\alpha(\mathbf{v}^*)$ and c is the adiabatic sound speed. At *boundary* faces we use an imaginary "ghost" state on the outside such that the Roe Riemann solver for the ghost and interior states will give the desired boundary condition (e.g. specified normal velocity, pressure). A well known defficiency of the Roe flux is its inability to properly identify an expansion fan containing a sonic point. To correct this we use the Harten-Hyman entropy fix that consists of modifying $|\lambda^*_\alpha|$ as follows:

$$|\lambda^*_\alpha| \to |\lambda^*_\alpha| + max(0, \epsilon - |\lambda^*_\alpha|),\ \epsilon \equiv max(0, \lambda^*_\alpha - \lambda^-_\alpha, \lambda^+_\alpha - \lambda^*_\alpha),\ \lambda^\pm_\alpha \equiv \lambda_\alpha(\mathbf{v}^\pm)$$

At this point the DGM equations reduce to a system of n_v ordinary differential equations (ODE) in time for the n_v moments $M_{\alpha n} \equiv \int_K \phi_n(\xi) A_\alpha d^3x$ for which we use an explicit time-adapting second order Runge-Kutta integration. To implement this we require, first to compute the primitive state variables B_α from the moments M_α and second to determine Δt for stability of the numerical scheme. We compute $B_{\alpha n}$ from $M_{\alpha n}$ by first introducing auxiliary variables $\tilde{A}_\alpha$ which are linear in each element K and such that its n_v moments are equal to $M_{\alpha n}$. In each cell, a linear representation of the primitive variables B_α, is obtained by using a first order Taylor series expansion of

B_α in $\tilde{A}_\alpha$ around the average values $< \tilde{A}_\alpha >=< A_\alpha >\equiv \int_K A_\alpha d^3x / \int_K d^3x$. For numerical stability of the time integration we require a Courant-type time step control: Δt must be smaller than the time it takes for a wave originating on a face of an element to cross it. We heuristically implement this requirement as follows: $\Delta t \leq CFL\ t_{Courant}$ where $CFL \approx .3$ [1] and

$$t_{Courant} \equiv \min_K \frac{\int_K d^3x}{\int_{\partial K} \max(|\lambda_p^*|, |\lambda_m^*|, |\lambda_o^*|) d\Gamma}$$

3 3D Shock Stabilization

In second order schemes the values of the primitive variables $B_{\alpha n}$ may develop local maxima and minima behind discontinuities due to dispersive truncation errors. These violations of physical stability constraints can usually be controlled by a slope limiting technique which modifies or "stabilize" the nodal values $B_{\alpha n}$. To this end we have generalized VanLeer's 1D minmod slope limiter to an unstructured 3D mesh through a quadratic programming formulation. The central idea is to require each nodal value $B_{\alpha n}$, within an element, to be bounded by the minimum and maximum of the average values $< B_\alpha >$ of all elements surrounding the node n. In general this will not be true. To satisfy this requirement, we replace $B_{\alpha n} with B'_{\alpha n}$ obtained from a least squares formulation *subject to the constraint:* $< B'_\alpha >=< B_\alpha >$. In addition, in order to keep our scheme as second order accurate as possible, we construct within each element a *hybrid* primitive state variable:

$$B_{\alpha n}^{hybrid} \equiv (1-s)(rB'_{\alpha n} + (1-r)B_{\alpha n}^{Godunov}) + sB_{\alpha n}, \quad B_{\alpha n}^{Godunov} \equiv < B_\alpha >$$

where r measures the strength of the shock ($0 \leq r \leq 1$, $r \approx 0$ near a very strong shock), and s measures the adiabaticity of the solution by examining the entropy production rate ($0 \leq s \leq 1$, $s \approx 1$ for small entropy production rates). Once $B_{\alpha n}^{hybrid}$ are obtained, the "stabilized" moments $M_{\alpha n}^{hybrid}$ are constructed by the inverse of the algorithm used to derive the primitive variables from the moments. $M_{\alpha n}^{hybrid}$ are then used in the next Runge-Kutta iteration.

While the *hybrid* stabilization works well for a variety of problems involving shocks there are cases where it is insufficient. This has led us to implement a Lapidus type artificial viscosity, i.e. adding a source term of the type $\nabla(D^L \nabla A_\alpha)$ to the hydro equations. We have empirically found the following implementation of the Lapidus flux correction to work well - in the vicinity of shocks add the following source term to the right hand side of the DGM equations for *interior* faces:

$$\frac{D^L}{l}[< A_\alpha^- >_n - < A_\alpha^+ >_n] \int_{\partial K} \phi_n(\xi) d\Gamma, \quad < A_\alpha^\pm >_n \equiv \frac{\int_{K^\pm} \phi_n(\xi) A_\alpha^\pm d^3x}{\int_{K^\pm} \phi_n(\xi) d^3x},$$

where D^L is an artificial diffusion coefficient that vanishes in the continuum limit, $\pm$ refers to the two elements at either side of the face, and l is some

length scale across the face (e.g. distance between centers of elements + and −). We *define* the Lapidus diffusion coefficient through:

$$\frac{D^L}{l} \equiv \kappa \lambda_K^* \quad , \quad \lambda_K^* \equiv \frac{\int_{\partial K} \max(|\lambda_p^*|, |\lambda_m^*|, |\lambda_o^*|) d\Gamma}{\int_{\partial K} d\Gamma}$$

where κ is a dimensionless adjustable parameter around 0.3 (for stability of the explicit scheme) and λ_K^* is a characteristic wave speed on the face.

4 3D ALE Grid Velocity

The grid velocity V_i^g at node n is arbitrary in an ALE code. If $V_i^g = 0$ everywhere in the mesh one has an Eulerian code. A "Lagrangian" code would require $V_i^g = v_i$. However, v_i is discontinuous across faces while we require V_i^g to be continuous. We therefore formulate an *almost* "Lagrangian" code by using a "least squares" estimate of the fluid velocity at node n for V_i^g:

$$\text{Minimize} \sum_{\{f_n\}} [N_{if_n}^g V_{in}^g - Y_{f_n}]^2, \quad Y_{f_n} \equiv N_{if_n}^g V_{if_n}^g,$$

where the sum is over the set $\{f_n\}$ of all faces with vertex n. Y_{f_n} is determined by requiring the first component of the Roe flux, the mass flux, to vanish, i.e. $N_i F_{1i}^{Roe}[Y_{f_n}] = 0$ for interior faces. For boundary faces Y_{f_n} is set to either the prescribed normal component of the velocity or is determined from the specified boundary pressure through the vanishing of the mass flux. While this procedure determines V_{in}^g fully, there are instances when constraining the grid velocity may prove beneficial in avoiding mesh tangling. Therefore, we have implemented a set of either $n_c = 1, 2, 3$ linear constraints on the components of V_{in}^g. If $n_c = 3$, V_{in}^g is completely determined (e.g. a center symmetry node may be required not to move).

5 Hydrodynamic Test Problems and the ICF3D Code

Details of the DGM algorithm, as described above, can be found in [2] and [3]. The algorithm has been implemented in C++ using an object oriented (OO) approach. The OO design allows us to untangle the complexities of the unstructured mesh data structure from the basic physics algorithm modules. Thus cells (tets,hexes..) and faces belong in separate classes and calculations such as flux integrals become virtual functions using pointers to access neccessary data. This design enables efficient code development.

We have tested the hydro code on a suite of problems relevant to inertial confinement fusion (ICF). A non-exhaustive list includes: LeBlanc shock tube problem (initial conditions of 10^3 density and 10^9 pressure ratios), Noh problem of an initially cold ($P = 0, \rho = 1$) gas moving with unit velocity into

a symmetry boundary, Sedov point explosion, 2D and 3D Rayleigh-Taylor instability. These problems have exact analytical solutions that can be checked against the computed ones. Our comparisons have shown extremely good agreement. For example, the computation of the linear growth rates for 2D and 3D Rayleigh-Taylor instability falls within 1% of the analytical value.

The success in simulating a wide span of test problems has made the hydro code the basis of the fully integrated three-dimensional ICF modeling code [4], ICF3D. Besides multiple material hydro, the physics modules of ICF3D include diffusive radiation and heat conduction transport, laser ray tracing, and realistic EOS.

The DGM hydro algorithm is ideally suited for parallelization. Indeed all physics modules of ICF3D have been parallelized [6] using domain decomposition and the MPI message passing library.

6 Open Problems

The *hybrid* 3D shock stabilization we employ is applied to each of the primitive variables, in particular each of the components of the 3D velocity vector. We have noticed that for problems with symmetry the relation between the cartesian velocity components (e.g. for cylindrical symmetry the azimuthal and axial velocity components vanish) may be destroyed by the hybrid stabilization. It is our conjecture, that understanding hybrid stabilization as the discretization of a continuum "artificial viscosity" operator could shed some light in devising a stabilization scheme that would maintain the symmetry requirements. Until we better understand this issue, we have instead extended our hydro code, to 3D cylindrical and spherical geometries [2], and used it in situations where cylindrical or spherical symmetry is critically important.

References

1. B. Cockburn and C. W. Shu, Math. Comput. **52**, 411 (1989).
2. D. S. Kershaw, M. K. Prasad and M. J. Shaw, 3D Unstructured Mesh ALE Hydrodynamics with the Upwind Discontinuous Finite Element Method. Lawrence Livermore National Laboratory report UCRL-JC-122104 (1995).
3. D. S. Kershaw, M. K. Prasad, M. J. Shaw and J. L. Milovich: 3D Unstructured Mesh ALE Hydrodynamics with the Upwind Discontinuous Finite Element Method. Comput. Methods Appl. Mech. and Engrg, **158**, 81-116 (1998).
4. A. I. Shestakov, M. K. Prasad, J. L. Milovich, N. A. Gentile, J. F. Painter and G. Furnish: The Radiation-Hydrodynamic ICF3D Code. to appear in Comput. Methods Appl. Mech. and Engrg, (1999).
5. A. I. Shestakov, J. L. Milovich and M. K. Prasad: ICF3D Results on Hydrodynamic Test Problems. Lawrence Livermore National Laboratory internal report (1997).
6. A. I. Shestakov, J. L. Milovich and D. S. Kershaw: Parallelization of a 3D Unstructured-grid, Laser Fusion Design Code. SIAM NEWS, News journal of the Society for Industrial and Applied Mathematics, **32**, 3, p. 6, April 1999.