LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

LA-UR -87-3163

CONF-8704208--1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

LA-UR--87-3163

DE88 000494

TITLE THREE-DIMENSIONAL, FREE-LAGRANGE HYDRODYNAMICS

AUTHOR(S) Harold Trease, X-7

SUBMITTED TO Numerical Applications for Particle Methods Meeting
Los Alamos, NM
April 6-10, 1987

DISCLAIMER

MASTER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4
ST. NO. 2629 5/81

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THREE-DIMENSIONAL, FREE-LAGRANGE HYDRODYNAMICS

HAROLD TREASE
LOS ALAMOS NATIONAL LABORATORY

1. INTRODUCTION:

The need to model high speed fluid flow that involves gross material deformation and (local) high shear flow regions is becoming a requirement for computational flow models.. These requirements dictate the use of robust, accurate numerical methods to model such fluid flow problems. Problems of this type have brought about the development of a new numerical technique called the FLM (Free-Lagrange Method). The two-dimensional development of this technique has been pioneered by Crowley [1] and Fritts[2]. Their work has led to further development by Clark[5] and Trease[4]. The first international conference on the FLM [5] was held in 1985.

This paper describes the algorithms that make up a 3-D version of the FLM (the 2-D version is described in [4]). The basic method involves the explicit integration of the fluid flow equations over control volumes formed by Voronoi cells. A Voronoi cell is an arbitrary, convex polyhedron that has plane polygon faces. The "nearest" neighbors of any given mass point are identified by noting the mass point that lies on the opposite side of a given Voronoi face. The nearest neighbors of a mass point are allowed to change in response to the (Lagrange) motion of the mass points. This implies that the mesh reorganizes itself as the mass points move with the fluid motion.

As will be described later the Voronoi meshing technique has several "nice" features about it that impact the robustness and accuracy of the solution of the equations that are solved on this type of mesh. The following is a quick summary of some of these nice features:

1) A convex polyhedron guarantees that the control volumes are not distorted or reentrant.

2) Each face of the polyhedra is described by arbitrary polygons such that each face intersects the line between two mass points as a perpendicular bisection plane. This implies that when forming the difference operators for any of the equations that (locally) the mesh spacing is always "uniform".

3) The space defined by an arbitrary distribution of mass points plus any external boundaries is tessellated completely and uniquely by a set of Voronoi polyhedra. This tessellation is guaranteed to give reciprocity of connections.

4) The maintenance of the (Voronoi) connectivity matrix as the Lagrangian mass points follow the fluid motion is a matter of applying the Voronoi "rules" for (re)defining the nearest neighbors of a point. This leads to an efficient and systematic algorithms for maintaining the global connectivity matrix.

5) During the reconnection of the mesh, where mass points change nearest neighbors, the integration control volumes for each mass point (i.e., the Voronoi polyhedra) do not change their topological form. This means that during the redefinition of the nearest neighbors of a mass point there is no need to "flux" mesh quantities because of the reconnections.

6) The distribution of mass points used to describe the geometry of a problem (and used as the basis of the discretization method) can be locally refined to resolve features of a problem. Since a connectivity matrix is constructed from the mass point distribution, and subsequently maintained by a mesh optimization phase, we can add resolution as a problem evolves or we can remove resolution if the need arises (e.g., an example would be a time-step crash caused by two mass points that come "too close").

In the remainder of this paper I will the describe the major steps and solution algorithms that are used to solve an example 3-dimensional fluid flow problem. Also, I will present the results of this problem (NOTE: This problem is a standard strong shock, test problem, called

the Noh problem [6]. This test problem represents just one of the code verification problems in our overall quality/assurance procedure). The definition of the example problem to be used to describe the FLM is shown in Fig. 1. Here we have a infinite strength shock moving into an undisturbed fluid. The problem is solved in plane geometry for the purposes of the this paper, but cylindrical and spherical versions are also used for code verification.

2. MESH GENERATION:

This step includes the initial definition of a problem. The setup of a problem's geometry requires the following:

1) A distribution of mass points over the region of space.
2) The definition of any external boundaries.
3) The types of materials in each material region along with initial material densities, temperatures, etc.

An example distribution of points and the resulting Voronoi mesh is shown in Fig. 2 (This is actually a self-generated distribution of points from a set of "seed" surface points and an automatic mesh refinement algorithm). This figure shows an arbitrary set of mass points and the corresponding Voronoi polygons that are constructed from the algorithm described by Trease [5].

The mesh that I will use in my example calculation(s) is shown in Figure 3a-c. Figure 3a shows the geometry of 1-D version of the problem. Figure 3b shows the 2-D definition of the problem and figure 3c shows the 3-D setup for the same problem.

The main features to note about the tessellation method is that the entire space is mapped by a set of non-overlapping convex polyhedra, where each mass point (located at the center of the cells) is separated from its nearest neighbors by perpendicular bisecting, plane polygon faces.

3. INTEGRATION METHOD:

To solve our example hydrodynamics problem, we must solve the fluid flow equations by using the specified initial conditions and boundary conditions. The solution method that I choose to use involves the explicit time integration of the (discretized) fluid flow equations over a set of control volumes represented by the Voronoi control volumes.

The differential form of the fluid flow equations that represent the conservation of mass, momentum, and internal energy are given by:

1) Conservation of Mass (continuity equation);

$$\frac{d\rho}{dt} = -\rho \vec{\nabla} \cdot \vec{u}$$

2) Conservation of Momentum;

$$\rho \frac{d\vec{u}}{dt} = -\vec{\nabla} P - \vec{\nabla} \cdot \vec{\vec{\tau}}$$

3) Conservation of Internal energy;

$$\rho \frac{dI}{dt} = -P \vec{\nabla} \cdot \vec{u} - [\vec{\nabla} \cdot (\vec{\vec{\tau}} \cdot \vec{u}) - \vec{u} \cdot \vec{\nabla} \cdot \vec{\vec{\tau}}]$$

ρ = FLUID DENSITY

$\vec{u} = \vec{u}(x,y,z) = \frac{d\vec{x}(x,y,z)}{dt}$ = FLUID VELOCITY

P = MATERIAL PRESSURE

$\vec{\vec{\tau}}$ = (ARTIFICIAL VISCOSITY) FLUID STRESS TENSOR

$\vec{\nabla}\phi$ = GRADIENT OPERATOR

$\vec{\nabla} \cdot \vec{A}$ = DIVERGENCE OPERATOR

These equations are spatially integrated over a control volume, where we transform (on the right-hand side of the equations) the volume integrals to surface integrals. These three equations now take on the integral form:

1) Conservation of Mass (continuity equation);

$$V \frac{d\rho}{dt} = -\bar{\rho} \int_S \hat{n} \cdot \vec{u}_s \, dS$$

V = VOLUME, dS = SURFACE AREA ELEMENT

$\hat{n}$ = UNIT NORMAL

2) Conservation of Momentum;

$$m \frac{d\vec{u}}{dt} = -\int_S \hat{n} p_s \, dS - \int_S \hat{n} \cdot \vec{\tau} \, dS$$

$$m = \int_V \rho \, dV = \text{MASS}$$

3) Conservation of Internal energy;

$$m \frac{dI}{dt} = -\bar{P} \int_S \hat{n} \cdot \vec{u}_s \, dS - \int_S \hat{n} \cdot (\vec{\tau} \cdot \vec{u})_s \, dS +$$

$$+ \vec{u}_s \cdot \int_S \hat{n} \cdot \vec{\tau}_s \, dS$$

These equations are now written in difference form as follows:

1) Conservation of Mass (continuity equation);

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} = \gamma_i \sum_{j=1}^{J} \hat{n}_{ij} \cdot (\vec{u}_{s_{ij}} - u_i) \Delta S_{ij}$$

$\hat{n}_{ij}$ = UNIT NORMAL FROM POINT "i" TO "j".

ΔS_{ij} = SURFACE AREA ELEMENT BETWEEN POINTS "i" & "j".

2) Conservation of Momentum;

$$m_i \frac{\vec{u}_i^{n+1} - u_i^n}{\Delta t} = -\sum_{j=1}^{J} \hat{n}_{ij} (P_{ij} - P_i) \Delta S_{ij} +$$

$$- \sum_{j=1}^{J} \hat{n}_{ij} \cdot \vec{\tau}_{ij} \Delta S_{ij}$$

3) Conservation of Internal energy:

$$M_i \frac{I_i^{n+1} - I_i^n}{\Delta t} = -P^{n+\frac{1}{2}} \sum_{j=1}^{J} \hat{n}_{ij} \cdot (\vec{u}_{ij} - \vec{u}_i) \Delta S_{ij} +$$

$$- \sum_{j=1}^{J} \hat{n}_{ij} \cdot (\vec{\tau} \cdot \vec{u})_{ij} \Delta S_{ij} +$$

$$+ \vec{u}_i^{n+\frac{1}{2}} \cdot \sum_{j=1}^{J} \hat{n}_{ij} \cdot \vec{\tau}_{ij} \Delta S_{ij}$$

There are several things to note about these equations.

1) Local gradients are represented by the difference of a given quantity between two nearest neighbors.

2) Since we are solving strong shockwave dominated problems we must define an artificial viscosity [7]. This is done in a tensor form by using the fluid stress tensor as an artificial viscosity tensor, where the fluid viscosity is replaced by an artificial viscosity [8]. This implies that artificial fluid stress tensor in cartesian coordinates has the form, where $\vec{\vec{I}}$ is the unit tensor:

$$\vec{\vec{\tau}}_{ij} = \begin{cases} c^2 \rho_{ij} (\Delta S_{ij})^2 \left\{ \vec{\nabla} \vec{u}_{ij} - \frac{1}{3} (\vec{\nabla} \cdot \vec{u}_{ij}) \vec{\vec{I}} \right\}, & \text{IF } (\vec{\nabla} \cdot \vec{u})_{ij} < 0 \\ 0, & \text{IF } (\vec{\nabla} \cdot \vec{u})_{ij} \geq 0 \end{cases}$$

3) Each pair of mass points are separated be a perpendicular bisecting plane. This means that locally the mesh looks like an equally spaced mesh. This fact enhances the (spatial) accuracy of the differencing method, as can be shown by a Taylor series expansion about one of the points.

For completeness, we note that in order for the solution method of the three equations described above to be closed, we must have an equation-of-state which relates the pressure of a material to the density and internal energy (or temperature) of a material. For our example problem we use an ideal gas relation of the form: $P = (\gamma - 1) I \rho$

γ = IDEAL GAS LAW CONSTANT

4. MESH OPTIMIZATION:

As our computational model follows the evolution of a fluid flow problem, the discrete Lagrangian mass points move with with fluid motion. This process ordinarily results in a (hopelessly) tangled computational mesh. The power of the FLM is that the mesh can be reorganized by a reconnection algorithm to avoid this mesh tangling problem. The method that I use to reconnect the mesh is the same one used to tesselate the mesh in the generation phase. The "new" set of nearest neighbors are (re)defined by the construction of the Voronoi cell by using the "old" nearest neighbors and the nearest neighbors of the "old" nearest neighbors. This set of "possible" nearest neighbors is guaranteed to be complete due to the explicit (Courant time-step limited) nature of the time integration method (i.e., any single mass point cannot move out of its sphere of nearest neighbors in one computational cycle).

One of the most important features of using the Voronoi mesh to define the computational control volumes becomes apparent when a nearest neighbor reconnection occurs. The topology of the mesh doesn't change as a result of the reconnection. This implies that during this step the fluid volumes associated with the control volumes will not change. Thus, none of the physical quantities associated with a given mass point (or Voronoi cell) need be fluxed between cells. Therefore, the numerical diffusion ordinarily associated with fluxing of conserved quantities is not a problem with a Voronoi mesh based version of the FLM.

5. RESULTS OF SOME CALCULATIONS:

The example calculation that I will use to demonstrate the utility of the FLM is based on what we call the Bill Noh problem [6]. This problem involves the generation of an infinite strength shock by a piston pushing into an undisturbed gas. The initial conditions and boundary conditions are shown in Fig. 1. Figures 3a-c shows the Voronoi mesh for three variations of the Noh problem. Fig. 3a is just the mesh

for a 1-D version of the problem. Fig. 3b is the mesh for a 2-D version of the problem. For the 2-D run, I have intentionally introduced a mesh perturbation into the problem. The problem is still made of a uniform undisturbed gas, but the mesh is not uniform. This will test the code's ability to pass a 1-D shock wave through a non-uniform mesh. The result should be a 1-D solution. Fig. 3c shows the final version of this problem where we run the same problem on a fully 3-dimensional mesh. The results for the 3-D case should be the same as for the 2-D case (i.e., the plane shock wave that is generated should remain 1-D even though the mesh has been perturbed. It should be noted that all of these problems were run on the same code. Only the point distributions were modified.

The results of the 1-D case are shown in Figures 4-7. The analytic solution to this problem gives the conditions of the shocked fluid as:

1) Shocked fluid density = 4.0
2) Shocked fluid pressure = 1.33333333
3) Shocked fluid (specific internal) energy = 0.5
4) Shocked fluid velocity = 0.0

The multidimensional results show a similar state of the shocked fluid behind the shock front. Figure 8 shows the 3-D time=0 Voronoi mesh. Figures 8-10 show the results of the 3-D calculation. Here the moving piston boundary moved from right to left. The effect of the mesh perturbation on the solution (indicated by the straightness and magnitude of the contour lines) is minimal as the density contour plots show.

6. CONCLUSIONS:

In this paper I have given a brief outline of the Free-Lagrange Method as applied in three dimensions, where the computational control volumes are Voronoi cells. I have described elsewhere how a Voronoi mesh can be generated from an arbitrary distribution of mass points [5] so I did not repeat that discussion. The results of a (simple) strong shock problem was presented showing the results in 1 and 3-dimensions. The

results all look good, but this is a relatively easy problem to model.

The main idea that should be gleaned from this paper is that the FLM is an extremely robust, accurate method for modelling fluid dynamics problems. Any geometry can be described, since the Voronoi mesh algorithm generates the connectivity matrix needed to define the nearest neighbors of the points. Also, the method allows for adding or removing resolution based on the local mesh refinement conditions. Lastly, the method preserves it's accuracy, independent of the fluid distortions, due to the local reconnection algorithm.

7. FUTURE WORK:

The future of the FLM looks very bright for solving highly distorted hydrodynamic problems. As discussed at the last conference on the FLM [5] there are currently two distinct, but related, approaches to using the Voronoi mesh as the basis for a code. First, the Voronoi mesh can be used to define both the nearest neighbor connectivity matrix and the topology of the computational control volumes. Second, the Voronoi mesh can be used to define the nearest neighbors but the control volume could be different from the Voronoi control volume. A good example of an alternative is to use the Median mesh to define the control volumes. A second alternative is to use the mesh that Pat Crowley describes in his paper for this same conference publication.

As I have described in this paper a code based completely on the Voronoi mesh is possible and produces good results for distorted flow problems. The major problem with the method is the computational speed of the resulting code. The 3-D code, described in this paper, consumes vast amounts of computer time on our largest computers here at Los Alamos. Therefore, the 3-D code team has embarked on a project whereby we will rewrite the code to improve the speed of the algorithms, but still retain the Free-Lagrange spirt.

The rewriting of the code is in progress as I write this paper. To improve the speed of the code we have made the following changes:

1) The basic topological element of the mesh is now a tetrahedron

rather than a polyhedron. Each tetrahedron represents one element of the Delaunay mesh (Note: The Delaunay mesh is the dual to the Voronoi mesh.)

2) The nearest neighbor generation and maintenance algorithms work with tetrahedrons. The resulting nearest neighbors are still the Voronoi nearest neighbors.

3) The integration of the hydrodynamic equations is performed with a predictor-corrector time stepping algorithm and over control volumes defined by the Median mesh (but using the Voronoi nearest neighbor connectivity).

The prognosis for the new algorithm is that it is significantly faster than the old one, but the results are comparable. Future publications will detail the results of detailed comparison between the Median and Voronoi algorithms.

REFERENCES:

[1] W. P. Crowley, "FLAG: A Free-Lagrange Method for Numerically Simulating Hydrodynamic Flows in Two Dimensions", Lecture Notes in Physics 8, 37-43, Springer (1970).

[2] M. J. Fritts, J. P. Boris, "The Lagrangian Solution of Transient Problems in Hydrodynamics using a Triangular Mesh", J. Comp. Phys. 31, 173-215 (1979).

[3] "The Free-Lagrange Method", Lecture Notes in Physics 238, 281-294, Springer (1985).

[4] H. E. Trease, "A Two-Dimensional Free Lagrangian Hydrodynamics Model", Ph.D. Thesis, University of Illinois, Urbana-Champaign (1981).

[5] "The Free-Lagrange Method", Lecture Notes in Physics 238, Springer (1985).

[6] W. F. Noh, "Artificial Viscosity for Strong Shocks", Lawrence Livermore National Laboratory, Livermore CA., UCRL-53669.

[7] J. Von Neumann, R. D. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks", J. Appl. Phys., 21, 232 (1957).

[8] W. M. Tscharnuter and K. H. Winkler, "A Method for Computing Selfgravitation Gas Flow with Radiation", MPI/PAE-Astro 163.

FIGURE CAPTIONS:

Figure 1: A description of the Noh problem in plane geometry.

Figure 2: Four views associated with the generation of a Voronoi mesh.
- Upper left is the inital point distribution.
- Upper right is the Voronoi mesh.
- Lower left is a rotated view of the Voronoi mesh.
- Lower right is the Delaunay mesh (dual of the Voronoi).

Figure 3: The Voronoi meshes showing the 1-D, 2-D and 3-D setups.

Figures 4-7: The results for the 1-D Noh problem.
- Figure 4: Density versus distance for the 1-D Noh problem at time=0.6.
- Figure 5: Energy versus distance for the 1-D Noh problem at time=0.6.
- Figure 6: Pressure versus distance for the 1-D Noh problem at time=0.6.
- Figure 7: Velocity versus distance for the 1-D Noh problem at time=0.6.

Figures 8-10: The results for the 3-D Noh problem.
- Figure 8: The time=0.0 setup for the 3-D Noh problem with a mesh perturbation imbedded in the problem.
- Figure 9: The time=0.38 results of the 3-D Noh problem that shows the position of the shock from in relation to the mesh.
- Figure 10: The time=0.38 results of the 3-D Noh problem that shows the position of the shock from in relation to the mesh. The position of the shock front is indicated by the leading edge (moving right to left) straight contours.

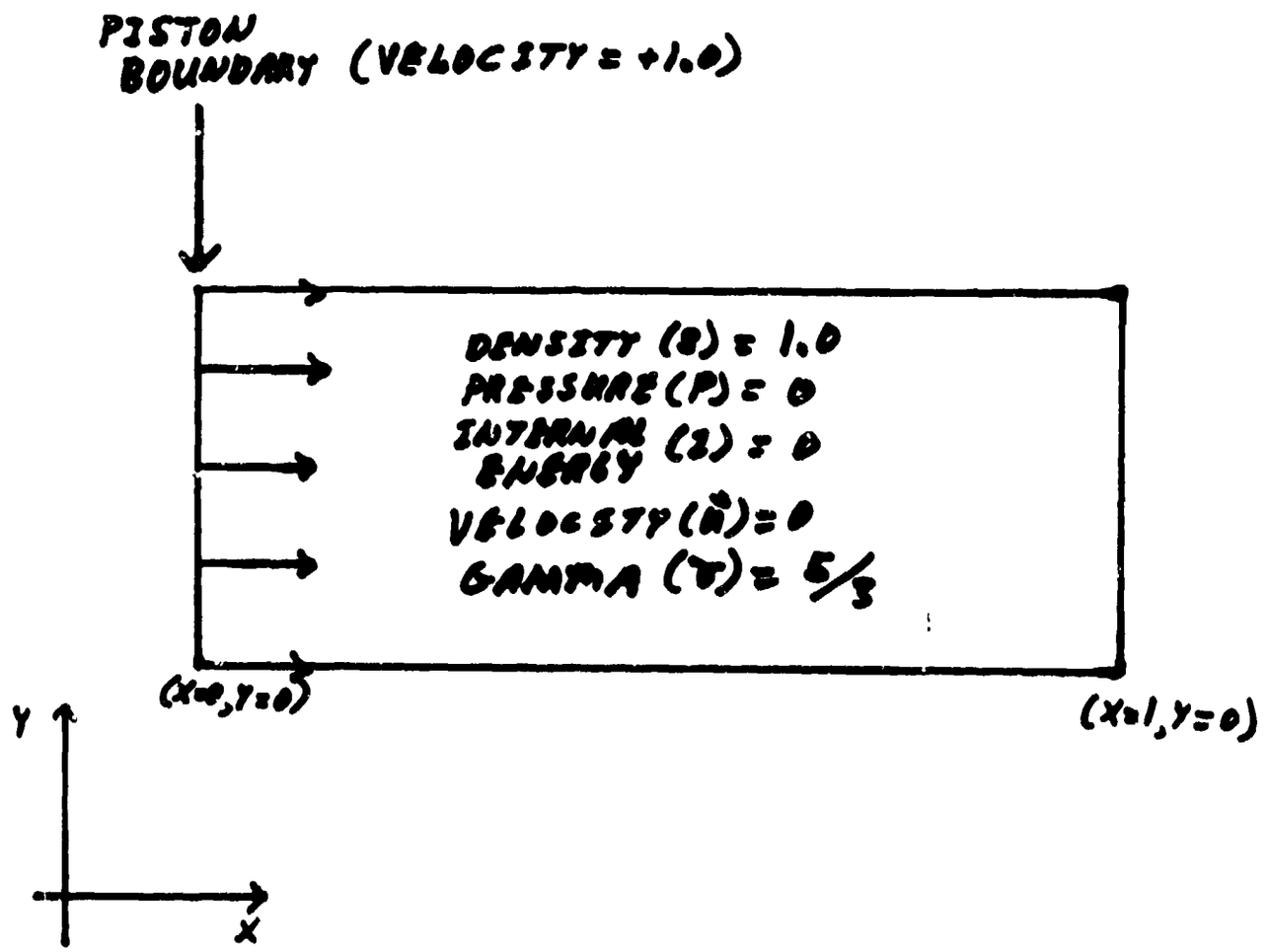
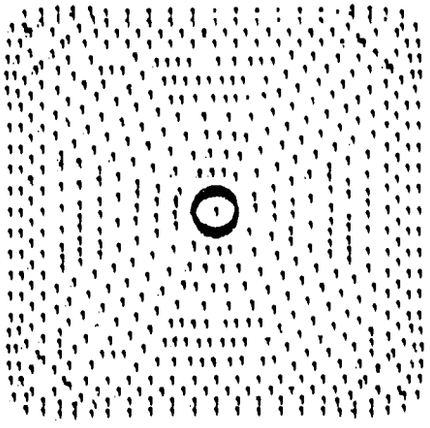
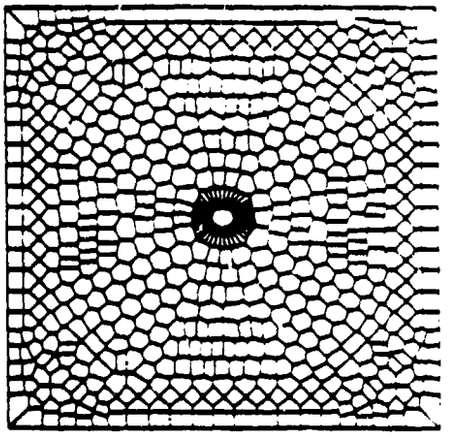


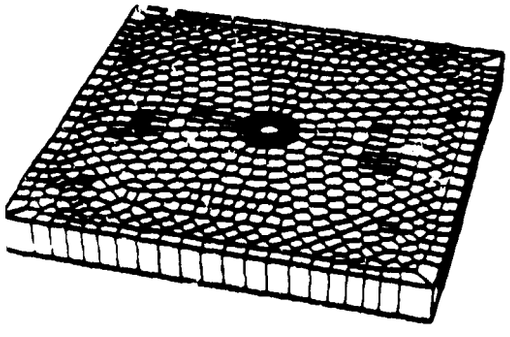
Figure 1: A description of the Noh problem in plane geometry.



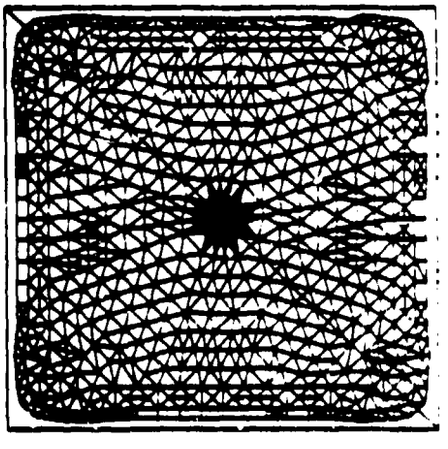
POINT DISTRIBUTION



VORONOI MESH

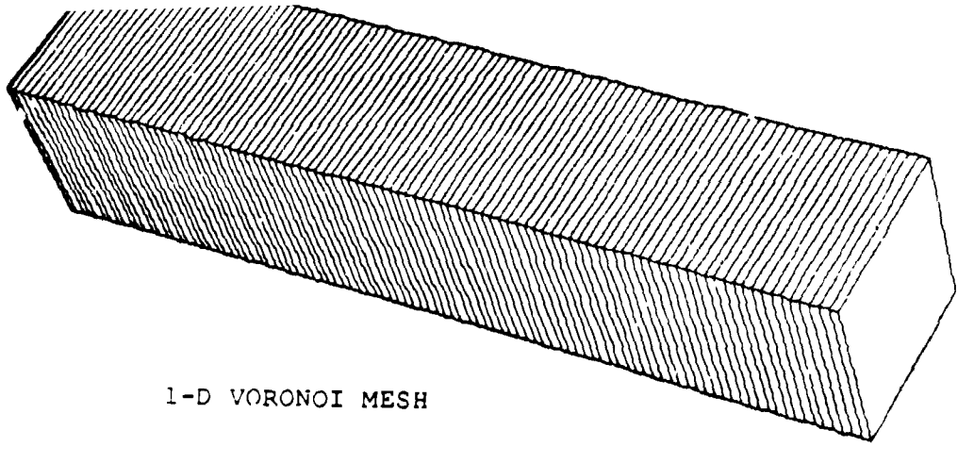


(ROTATED) VORONOI MESH

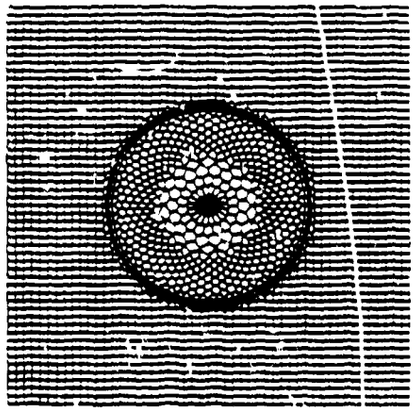


DELAUNAY MESH

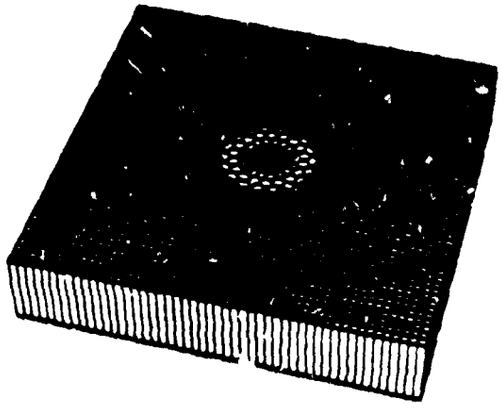
Figure 2: Four views associated with the generation of a Voronoi mesh.
- Upper left is the inital point distribution.
- Upper right is the Voronoi mesh.
- Lower left is a rotated view of the Voronoi mesh.
- Lower right is the Delaunay mesh (dual of the Voronoi).



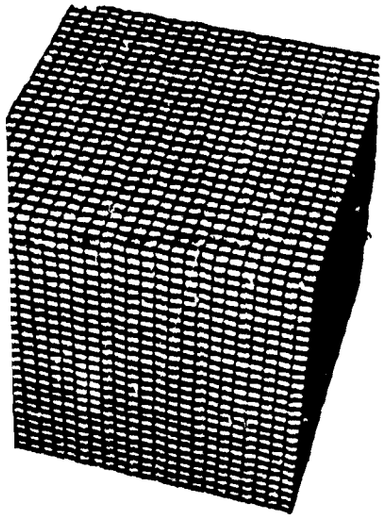
1-D VORONOI MESH



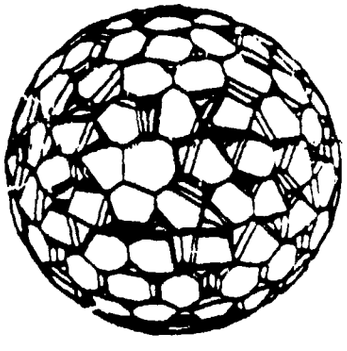
2-D VORONOI MESH



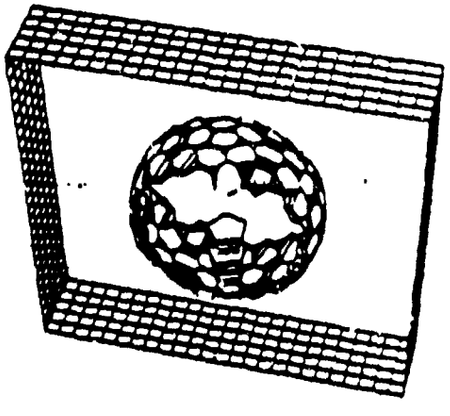
(ROTATED) 2-D PROBLEM



ORONOI MESH



VORONOI MESH
FOR IMBEDDED BALL



CLIPPED VORONOI MESH

Figure 3: The Voronoi meshes showing the 1-D, 2-D and 3-D setups.

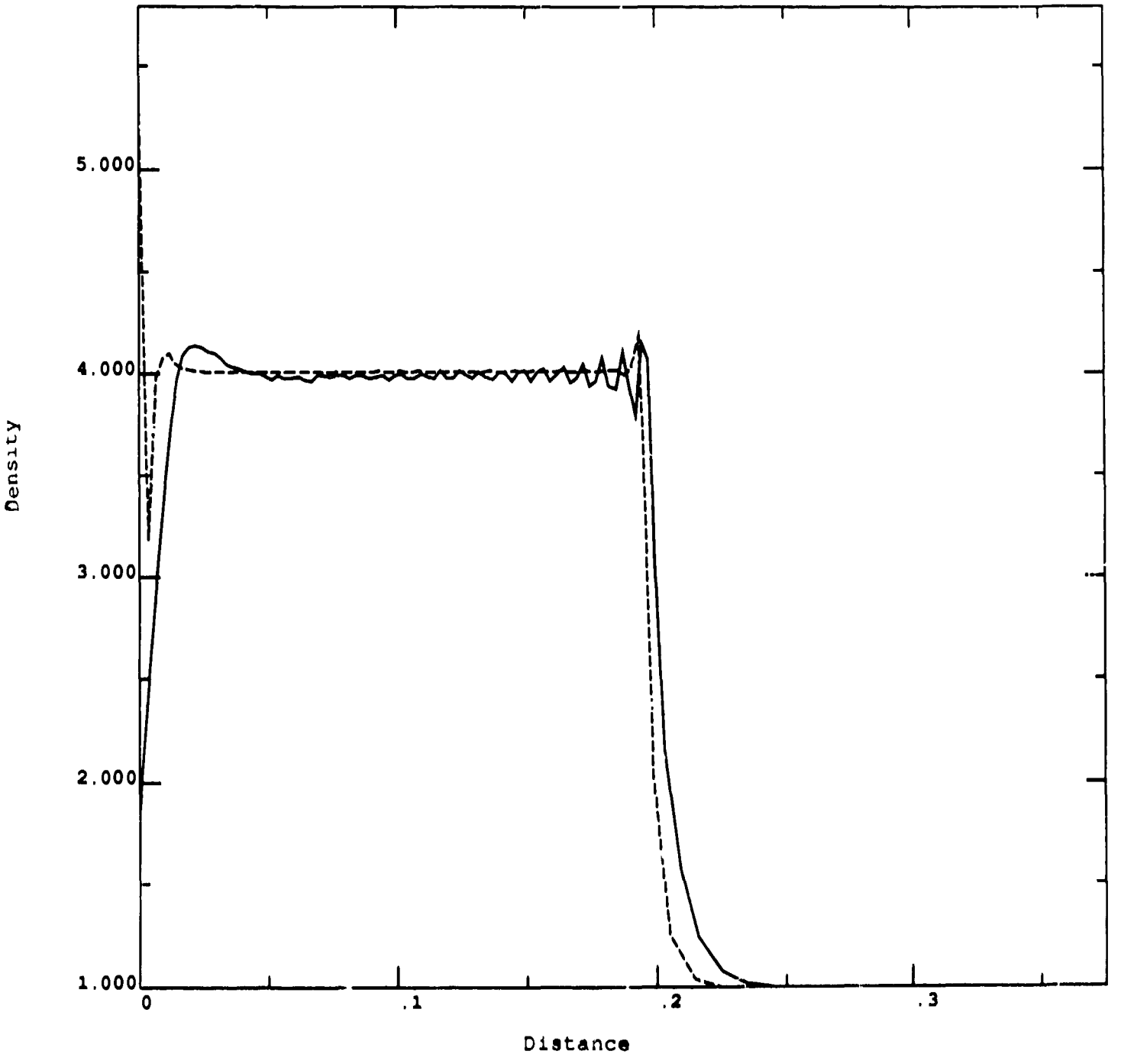


Figure 4: Density versus distance for the 1-D Noh problem at time=0.6.

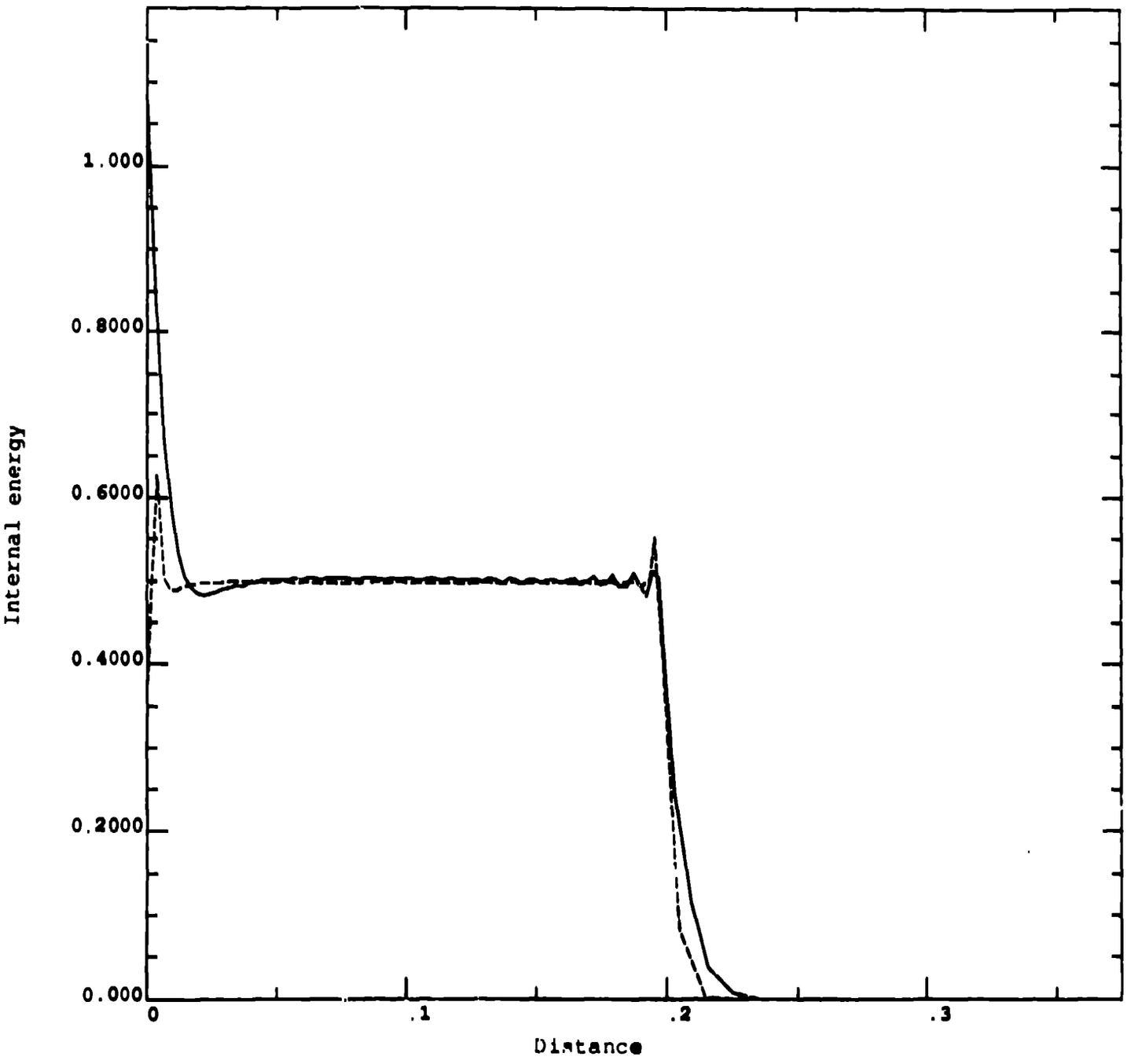


Figure 5: Energy versus distance for the 1-D Noh problem at time=0.6.

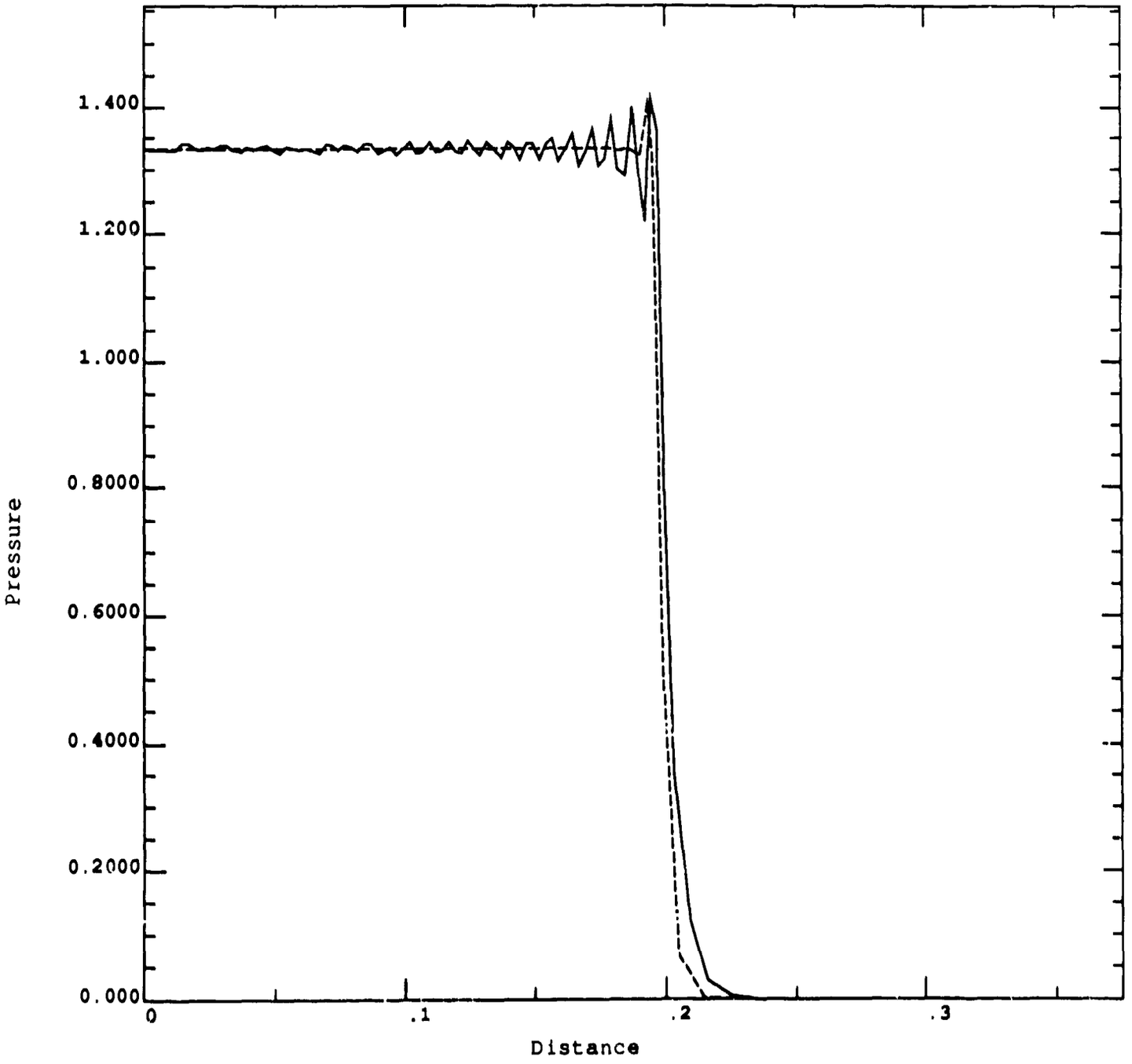


Figure 6: Pressure versus distance for the 1-D Noh problem at time=0.6.

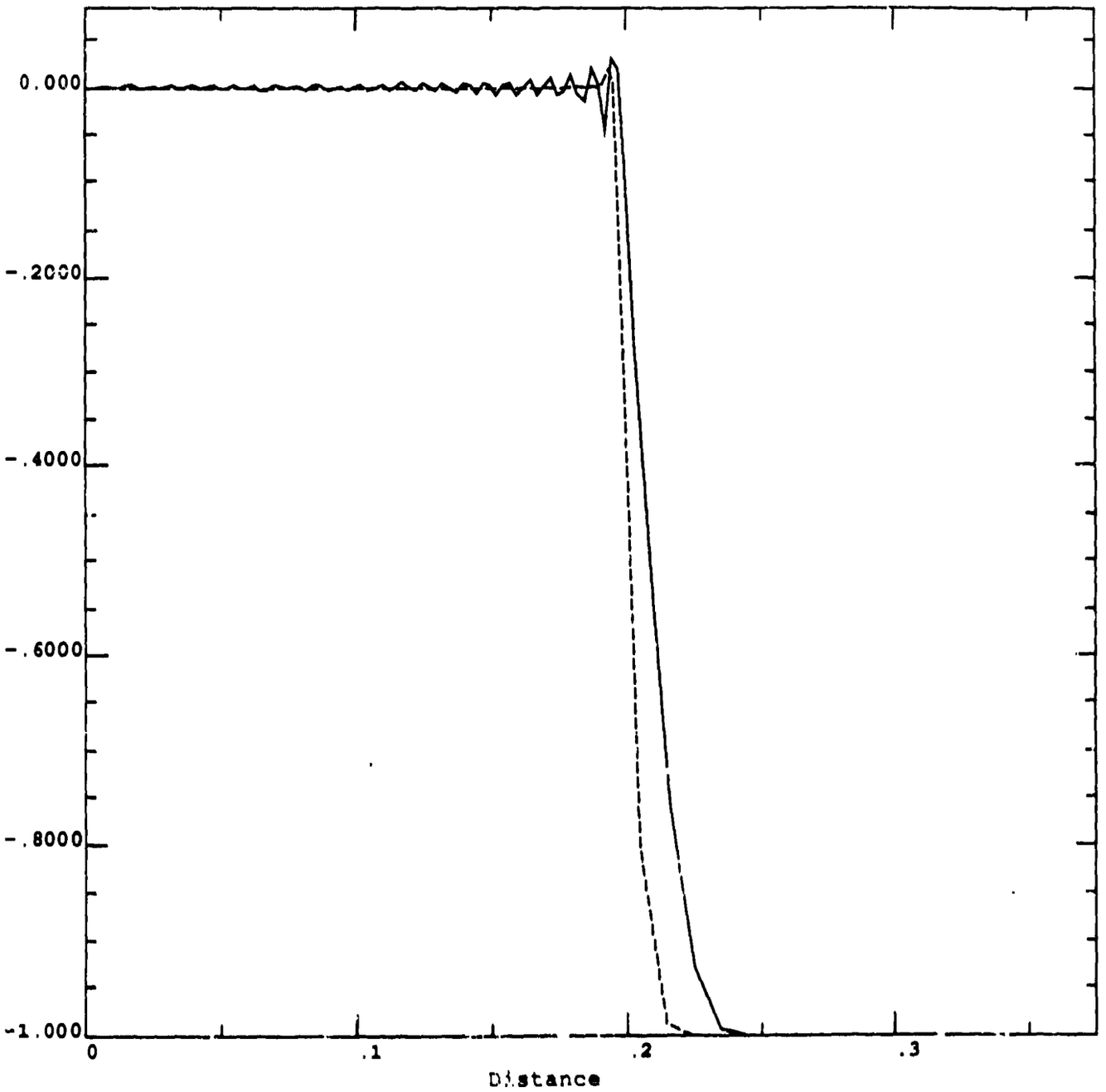


Figure 7: Velocity versus distance for the 1-D Noh problem at time=0.6.