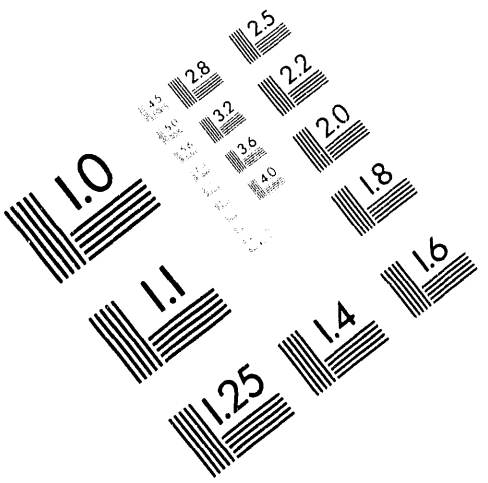
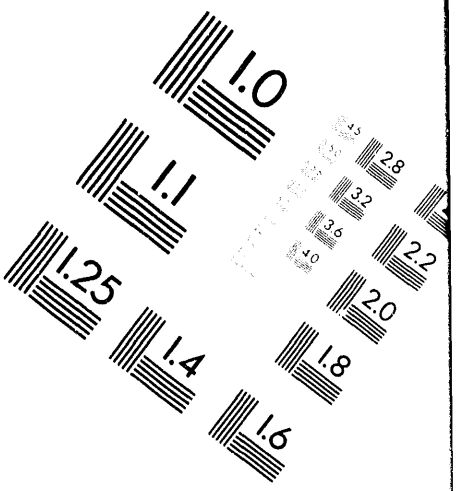
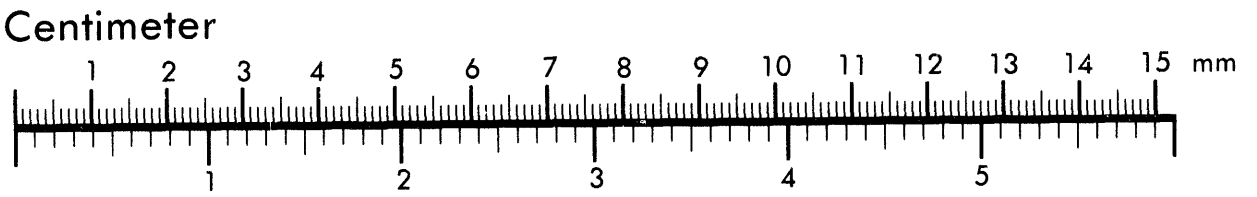


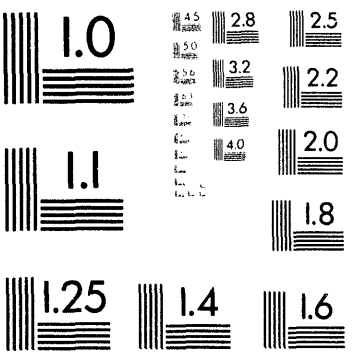
Association for Information and Image Management

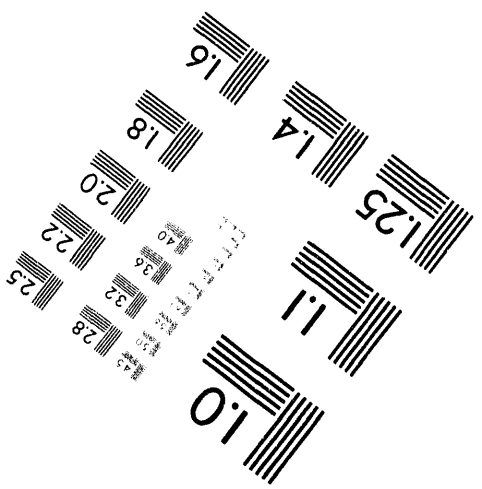
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

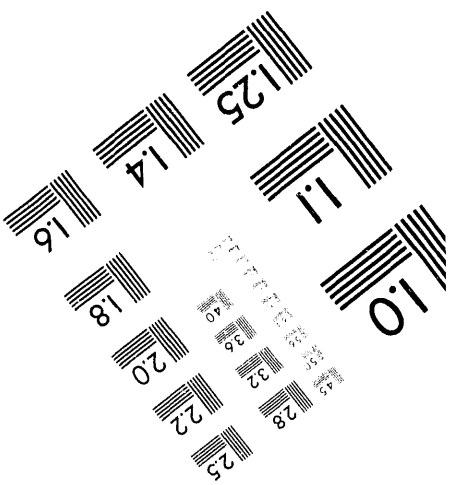
301/587-8202











MANUFACTURED TO AIIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 2

SAND93-2513
Unlimited Release
Printed March 1994

Distribution
Category UC-705

An Analysis of Smoothed Particle Hydrodynamics

J. W. Swegle
Solid and Structural Mechanics Department, 1562
S. W. Attaway
Computational Mechanics and Visualization Department, 1425
M. W. Heinstein
Engineering Mechanics & Material Model Department, 1561
F. J. Mello
Solid and Structural Mechanics Department, 1562
Sandia National Laboratories
Albuquerque, New Mexico 87185

D. L. Hicks
Michigan Technological University
Houghton, Michigan 49931

Abstract

SPH (Smoothed Particle Hydrodynamics) is a gridless Lagrangian technique which is appealing as a possible alternative to numerical techniques currently used to analyze high deformation impulsive loading events. In the present study, the SPH algorithm has been subjected to detailed testing and analysis to determine its applicability in the field of solid dynamics. An important result of the work is a rigorous von Neumann stability analysis which provides a simple criterion for the stability or instability of the method in terms of the stress state and the second derivative of the kernel function. Instability, which typically occurs only for solids in tension, results not from the numerical time integration algorithm, but because the SPH algorithm creates an effective stress with a negative modulus. The analysis provides insight into possible methods for removing the instability. Also, SPH has been coupled into the transient dynamics finite element code PRONTO, and a weighted residual derivation of the SPH equations has been obtained.

Intentionally Left Blank

Contents

Figures

Tables

Intentionally Left Blank

1. Introduction

SPH (Smoothed Particle Hydrodynamics)[1-8] is a gridless Lagrangian technique which is appealing as a possible alternative to numerical techniques currently used to analyze high deformation impulsive loading events, such as hypervelocity impact or explosive loading of materials. While Eulerian techniques can easily handle the gross motions associated with the large deformations involved in such events, detailed analysis is difficult because of the lack of history and the smearing and spreading of information (referred to here as diffusion) as the mass moves through the fixed-in-space Eulerian grid. Standard Lagrangian techniques, although desirable due to their ability to keep accurate histories of the events associated with each Lagrangian element, cannot be used because the material deformations are so large that the Lagrangian grid becomes severely distorted and the calculation breaks down.

For instance, sophisticated material models which describe the behavior of matter under extreme loading conditions usually require keeping precise histories at each material element, including the evolution of internal state variables which determine the evolving material properties. While it is relatively simple to produce this information from Lagrangian wavecode calculations, it is a much more difficult task for an Eulerian wavecode. The difficulty is that while Lagrangian techniques are excellent at tracking the history associated with each material element and can easily save the required information, the basic Eulerian technique involves only the current state of the material and does not deal with its history. Some Eulerian codes have had a general internal state variable capability developed which allows information to be saved in variables which are advected with the material as it crosses the fixed-in-space cell boundaries. However, there is a problem with diffusion of these quantities as material motion occurs. The process is illustrated in Figure 1.1. The straight lines on the left of the figure represent the rectangular

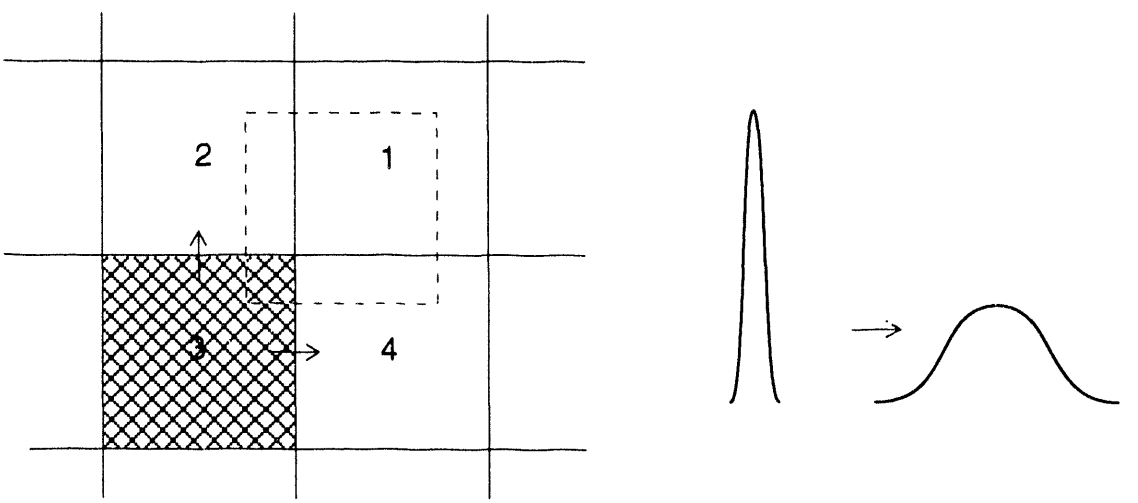


Figure 1.1 Information diffusion due to material motion in Eulerian methods.

Eulerian grid. If the cross-hatched mass in cell 3 has a velocity up and to the right as indicated by the arrows, the mass will occupy cells 1, 2, 3, and 4 at the next time step. The value of any internal state variables originally associated with cell 3 will then produce values in all four cells with magnitudes related to the fraction of the original mass currently in each cell. The right side of the figure indicates that sharply peaked distributions are thus diffused as material motion occurs. It is therefore difficult to use internal state variables to keep accurate histories when large deformations occur.

In many high deformation calculations material fracture will occur and bodies will separate into individual fragments. Treatment of this phenomenon in Lagrangian codes usually requires complicated (and often quite ad hoc) slide line algorithms. Eulerian codes also have difficulties in this regime, especially for bodies whose dimensions are small relative to the cell size. Since the Eulerian grid does not deform to follow material interfaces, in addition to single-material cells the method must treat mixed cells which contain either multiple materials or a single material and void. The strength associated with a single material cell is equal to the fracture strength of the material. However, mixed cells are usually indicative of the presence of a material interface which should support little or no tensile stress. Thus, the strength of a mixed cell should be characteristic of the strength of the interface, rather than the strengths of the individual materials in the cell. The tensile stress in a mixed cell therefore never exceeds the strength of the interface, with the result that fracture, based on the tensile stress exceeding the fracture strength of the material, cannot be detected in mixed cells. As indicated in Figure 1.2 , only

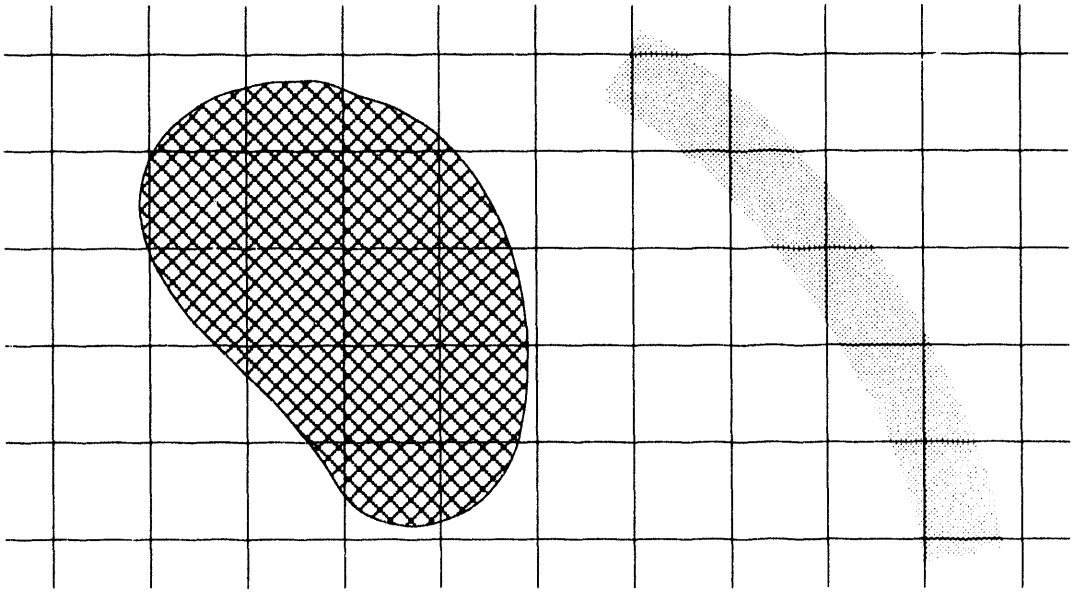


Figure 1.2 Partially-filled Eulerian cells

bodies large enough to occupy entire cells, such as the cross-hatched shape on the left, will allow fracture to be detected. However, only the mass in those cells

which are completely covered by the body, roughly half the mass, can fracture. Fracture can never be detected in thin shells, such as the one on the right side of the figure, which never fully occupy an entire cell. Thus, while Eulerian codes can be used to calculate large deformations, the validity of calculations using sophisticated internal state variable models or involving the production of individual fragments is problematical at best.

SPH offers a possible solution to these difficulties. The technique is Lagrangian and thus provides complete history information and should be well-suited for tracking details of the deformation process associated with each material element. SPH is actually quite similar to standard Lagrangian methods. In fact, the term hydrodynamic in the name is a misnomer, since strength is easily included. The difference from standard techniques is that spatial gradients are approximated by a method which is applicable to an arbitrary distribution of interpolation points so that no grid is required.Thus, the technique is gridless and should be applicable to arbitrary deformations, including the production of individual fragments. The lack of a grid also means that 3D calculations are as easy as 1D. Various organizations which have chosen SPH as a natural technique for large deformation calculations have used it to produce numerous results and are strongly supportive of its capabilities. The purpose of the present effort is to closely investigate the behavior of SPH algorithms and determine the applicability of the method for the detailed analysis of high deformation events.

Also, SPH has been coupled to the transient dynamics finite element code, PRONTO[9], and a new weighted residual derivation for the SPH method has been obtained. The coupling embeds the SPH method within the finite element code and treats each SPH particle as an element within the finite element architecture. Contact surface algorithms used in the finite element method were modified to couple the SPH particles with the finite elements. The ability to couple particle methods and finite element method allows fluid-structure interaction problems to be solved efficiently.

2. SPH Description

The basic objective of the SPH method is the numerical solution of the initial-boundary value problem defined by the partial differential equations which express the laws of conservation of mass, momentum, and energy, plus constitutive relations describing the materials in the problem. It is called an initial-boundary value problem because additional information is required which consists of 1) initial values for all variables at all positions at time zero and 2) boundary values for all variables at the boundary positions at all times. The task is to extend this information to all positions and times of interest. The primary difference between the smoothed particle method and standard Lagrangian finite-difference techniques is the replacement of grid-based approximation of spatial derivatives with a technique applicable to a random collection of interpolation points. In order to appreciate this difference, consider the following brief sketch of the general steps associated with any explicit numerical method:

1. Discretize information in time and space so that it is only known at a discrete number of positions and times. Start the cycle with the information at all positions known at time t. This may be either time zero at which all information was initially defined, or some later time to which the information has been advanced.

2. Based on the values of the stresses at each discretized location at time t, construct an approximation to the stress divergence. The acceleration of each discretized location at time t can then be computed from the law of conservation of momentum (the equation of motion).

3. Use the accelerations at time t to compute the new velocities and finally the new positions at time $t + \Delta t$, where Δt is the incremental time step. Calculate the new strain rates from the spatial derivatives of the new velocities, and the new strains from the new positions.

4. Use the new strains and strain rates to calculate the new stresses at time $t + \Delta t$. All information has now been advanced to the new time, and the cycle can repeat.

During this process, spatial derivatives of stress are required to compute accelerations, and spatial derivatives of velocity are required to compute strain rates. It is in this step that SPH differs from standard grid-based Lagrangian methods. The grid-based methods assume a fixed connectivity between particles (neighbors are assumed to remain neighbors) in order to construct approximations for spatial derivatives, while SPH uses the Kernel approximation, which is based on randomly distributed interpolation points with no assumptions about which points are neighbors. SPH is thus applicable to large deformations while grid-based Lagrangian techniques are not. Details are provided below.

2.1 Kernel Approximation

Consideration of standard Lagrangian finite-difference techniques shows that the major purpose of the spatial grid is to provide a basis for the construction of approximations to spatial derivatives. As shown in Figure 2.1 , the grid provides a

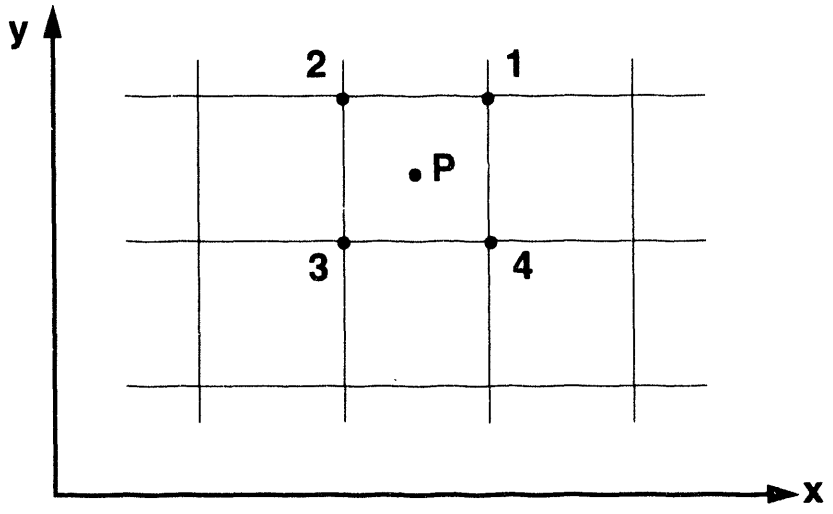


Figure 2.1 Standard finite-difference grid.

framework which specifies the relationship between the various nodes and allows algorithms for estimating spatial derivatives to be implemented. For instance, a standard two-dimensional finite-difference approximation[10] for the partial derivative with respect to x at point P is

$$\left.\frac{\partial \psi}{\partial x}\right|_P \approx \left(\oint_{C_{1234}} \psi dy \right) / A_{1234} \approx \frac{(\psi_1 - \psi_3)(y_2 - y_4) + (\psi_2 - \psi_4)(y_1 - y_3)}{2A_{1234}} , \quad (2.1)$$

where the approximation is obtained by applying Green's theorem in the plane and taking the line integral around the quadrilateral surrounding the point P, with A_{1234} the area of the quadrilateral. The problem with this technique is that it assumes that the same four nodes will always surround point P. Also, as the grid distorts and the quadrilateral becomes less square, the approximation loses accuracy.

The smoothed particle technique involves replacing grid-based approximations such as Eq. (2.1) with algorithms applicable to an arbitrary collection of interpolation points. The basis of the method is the kernel estimate, which starts from the identity

$$f(x) = \int f(x') \, \delta(x - x') \, dx' , \quad (2.2)$$

where f is a vector function of the three-dimensional position vector x, dx' is a volume, and $\delta(x-x')$ is the Dirac delta function. If $\delta(x-x')$ is replaced by a kernel function $W(x-x', h)$ where h is known as the smoothing length, the result is the kernel estimate

$$f(x) \approx \int f(x')\, W(x-x', h)\, dx' . \tag{2.3}$$

Various possibilities exist for the choice of $W(x-x')$. The requirements that are usually placed on the kernel function are 1) it reduces to the delta function,

$$\lim_{h \to 0} W(x-x', h) = \delta(x-x') , \tag{2.4}$$

2) it is normalized,

$$\int W(x, h)\, dx = 1 , \tag{2.5}$$

and 3) it has compact support (is zero everywhere but on a finite domain),

$$W(x, h) = 0 \quad \text{for} \quad |x| \geq 2h . \tag{2.6}$$

The approximation for spatial derivatives is obtained by substituting $\nabla \bullet f(x)$ for $f(x)$ in Eq. (2.3),

$$\nabla \bullet f(x) \approx \int \nabla \bullet f(x')\, W(x-x', h)\, dx' . \tag{2.7}$$

The divergence in the integral is taken with respect to the primed coordinate system. Now

$$\nabla \bullet f(x')\, W(x-x', h) = \nabla \bullet (f(x')\, W(x-x', h)) - f(x') \bullet \nabla W(x-x', h) , \tag{2.8}$$

so that

$$\nabla \bullet f(x) \approx \int \nabla \bullet (f(x')\, W(x-x', h))\, dx' - \int f(x') \bullet \nabla W(x-x', h)\, dx' . \tag{2.9}$$

The first term on the right side of the equation can be converted by means of the divergence theorem into an integral over the surface of the domain of integration

$$\int \nabla \bullet (f(x')\, W(x-x', h))\, dx' = \int_S f(x')\, W(x-x', h) \bullet \hat{n}\, dS = 0 . \tag{2.10}$$

The fact that the surface integral is zero follows from Eq. (2.6), as long as the region of integration is further than $2h$ from the boundary of the material. If it is not, modifications should be made to account for boundary conditions, although this is

rarely done due to the fact that the unmodified method works for free surfaces. Thus,

$$\nabla \bullet f(x) \approx -\int f(x') \bullet \nabla W(x-x', h)\, dx' \ . \tag{2.11}$$

Again, all gradients in the above integrals are taken with respect to the primed coordinate system. The kernel approximation thus allows spatial gradients to be determined from the values of the function and the derivative of the kernel, rather than the derivatives of the function itself. A final step is to convert from continuous volume integrals to sums over discrete interpolation points. Note

$$\int \phi(x')\, dx' = \int \left(\frac{\phi(x')}{\rho} \right) \rho\, dx' \ . \tag{2.12}$$

If $\rho dx'$ is interpreted as the mass associated with the interpolation point and the integral is approximated by finite sums using the value of the integrand at the interpolation point, then

$$\int \phi(x')\, dx' \approx \sum_{J=1}^{N} \frac{\phi(x^J)}{\rho^J} m^J \quad , \tag{2.13}$$

where the superscript indicates the value of the quantity at interpolation point J, and the sum is over N interpolation points. Thus, equations (2.3) and (2.11) become

$$f(x) \approx \sum_{J=1}^{N} \frac{m^J}{\rho^J} f(x^J)\, W(x-x^J, h) \tag{2.14}$$

and

$$\nabla \bullet f(x) \approx -\sum_{J=1}^{N} \frac{m^J}{\rho^J} f(x^J) \bullet \nabla W(x-x^J, h) \ , \tag{2.15}$$

where the gradient ∇W in Eq. (2.15) is with respect to x^J. The above equations provide continuous approximations to a function and its spatial gradient based on an arbitrary set of discrete interpolation points at which the function is known. It is clear from the above that SPH 'particles' should be thought of as interpolation points rather than as interacting mass elements. No connectivity or spatial relation of the points is assumed. The sum is over the entire set of points, although only those within the range of the kernel function (typically $2h$) will contribute. In order to avoid an N^2 algorithm in which the distance between all particles is tested to determine which particles contribute to the sums, a search algorithm is used to find

neighbors. Appendix A describes a newly developed algorithm whose execution time is roughly $N \log N$, and which is extremely efficient in terms of memory utilization.

Equation (2.15) provides a straightforward approximation for any spatial gradients which may be required in implementing the smoothed particle algorithm. However, variants of this relation are sometimes used which start either from the identity

$$\nabla \bullet \boldsymbol{f}(\boldsymbol{x}) = \frac{1}{\rho}\left[\nabla \bullet (\rho \boldsymbol{f}(\boldsymbol{x})) - \boldsymbol{f}(\boldsymbol{x}) \bullet \nabla \rho\right] \tag{2.16}$$

or

$$\nabla \bullet \boldsymbol{f}(\boldsymbol{x}) = \rho\left[\nabla \bullet \left(\frac{\boldsymbol{f}(\boldsymbol{x})}{\rho}\right) + \frac{\boldsymbol{f}(\boldsymbol{x})}{\rho^2} \bullet \nabla \rho\right] . \tag{2.17}$$

These relations may be substituted in the integral in Eq. (2.7), and a manipulation similar to that leading to Eq. (2.15) may be applied to each term. The only additional step is that each of the expressions multiplying $\nabla \rho$ in the second term on the right-hand sides of Eqs. (2.16) and (2.17) is brought outside the integral and evaluated at the point at which the gradient is being evaluated[8]. The results for the divergence of $\boldsymbol{f}$ at the position $\boldsymbol{x}^I$ of the I^{th} particle are

$$\nabla \bullet \boldsymbol{f}(\boldsymbol{x}^I) \approx \frac{1}{\rho^I} \sum_{J=1}^{N} m^J (\boldsymbol{f}(\boldsymbol{x}^I) - \boldsymbol{f}(\boldsymbol{x}^J)) \bullet \nabla W(\boldsymbol{x} - \boldsymbol{x}^J, h) , \tag{2.18}$$

based on Eq. (2.16), and

$$\nabla \bullet \boldsymbol{f}(\boldsymbol{x}^I) \approx -\rho^I \sum_{J=1}^{N} m^J \left(\frac{\boldsymbol{f}(\boldsymbol{x}^I)}{(\rho^I)^2} + \frac{\boldsymbol{f}(\boldsymbol{x}^J)}{(\rho^J)^2}\right) \bullet \nabla W(\boldsymbol{x} - \boldsymbol{x}^J, h) , \tag{2.19}$$

based on Eq. (2.17). The rationale behind the use of these relations will be discussed in the next section.

2.2 Numerical Algorithm

The only spatial gradients which must be evaluated in a standard Lagrangian finite-difference numerical algorithm are the stress divergence in the equations of motion

$$\boldsymbol{a} = \frac{1}{\rho} \nabla \bullet \sigma , \tag{2.20}$$

where σ is the Cauchy stress tensor, and the velocity gradients in the strain-rate tensor,

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right), \tag{2.21}$$

and in the vorticity tensor

$$\omega_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i}\right). \tag{2.22}$$

When indicial, rather than direct, notation is used the subscripts i and j refer to the spatial components in the x_1, x_2, and x_3 directions. Straightforward application of Eq. (2.15) to the equation of motion gives the acceleration of the I^{th} particle,

$$a_i^I = -\sum_{J=1}^{N} m^J \sum_{j=1}^{3} \frac{\sigma_{ij}^J}{\rho^I \rho^J} \frac{\partial W}{\partial x^J{}_j}. \tag{2.23}$$

Alternatively, Eq. (2.19) may be used, with the result

$$a_i^I = -\sum_{J=1}^{N} m^J \sum_{j=1}^{3} \left[\left(\frac{\sigma_{ij}}{\rho^2}\right)^I + \left(\frac{\sigma_{ij}}{\rho^2}\right)^J\right] \frac{\partial W}{\partial x^J{}_j}. \tag{2.24}$$

The latter relation has the advantage that the force on particle I due to particle J is the same as that on particle J due to particle I, so that strict momentum conservation is guaranteed. For the velocity gradient, an analogue of Eq. (2.15) yields

$$\left(\frac{\partial V_i}{\partial x_j}\right)^I = -\sum_{J=1}^{N} \frac{m^J}{\rho^J} V_i^J \frac{\partial W}{\partial x^J{}_j} \tag{2.25}$$

However, application of Eq. (2.18) yields

$$\left(\frac{\partial V_i}{\partial x_j}\right)^I = \frac{1}{\rho^I} \sum_{J=1}^{N} m^J \left(V_i^I - V_i^J\right) \frac{\partial W}{\partial x^J{}_j}. \tag{2.26}$$

The latter relation has the advantage that the contribution to the strain rate tensor from particles I and J is zero if their relative velocity is zero. In general, there are only minor differences between the regular forms, Eqs. (2.23) and (2.25), and the symmetric forms, Eqs. (2.24) and (2.26), except in special cases which can occur at boundaries.

Various schemes exist for advancing the solution in time, but a simple centered-difference scheme for the approximation of time derivatives will result in a numerical algorithm which differs from a standard Lagrangian finite-difference technique only in the approximations to the spatial derivatives described above. Since the equations of motion involve no time derivatives, all quantities are evaluated at the same time. Consequently, the equation of motion, Eq. (2.24), provides accelerations at time t^n from stresses and positions known at time t^n. The new velocities and positions can then be obtained from

$$V_i^{I,n+\frac{1}{2}} = V_i^{I,n-\frac{1}{2}} + \frac{1}{2}\left(\Delta t^{n+\frac{1}{2}} + \Delta t^{n-\frac{1}{2}}\right) a_i^{I,n} \tag{2.27}$$

and

$$x_i^{I,n+1} = x_i^{I,n} + \Delta t^{n+\frac{1}{2}} V_i^{I,n+\frac{1}{2}} \quad . \tag{2.28}$$

The superscripts involving n denote the time at which the quantity is evaluated. The centered temporal difference scheme results in accelerations and positions being evaluated at integral times, while velocities are evaluated at half-integral times. Equations (2.27) and (2.28) allow the new positions at time t^{n+1} to be determined from accelerations at time t^n. The strain rates can then be determined from the velocity gradient. Proper time centering requires that the rates be evaluated at time $t^{n+\frac{1}{2}}$, so positions and densities at that time must be used in the kernel approximation for the velocity gradient. Quantities at half-integral times can be evaluated from expressions of the form

$$x_i^{I,n+\frac{1}{2}} = \frac{1}{2}\left(x_i^{I,n+1} + x_i^{I,n}\right) . \tag{2.29}$$

The new density at time t^{n+1} can be determined directly from the kernel approximation, Eq. (2.14), resulting in

$$\rho(x) = \sum_{J=1}^{N} m^J W(x - x^J, h), \tag{2.30}$$

where all positions are evaluated at time t^{n+1}. However, Eq. (2.30) results in undesirable effects at boundaries. Figure 2.2 shows a graphical representation of

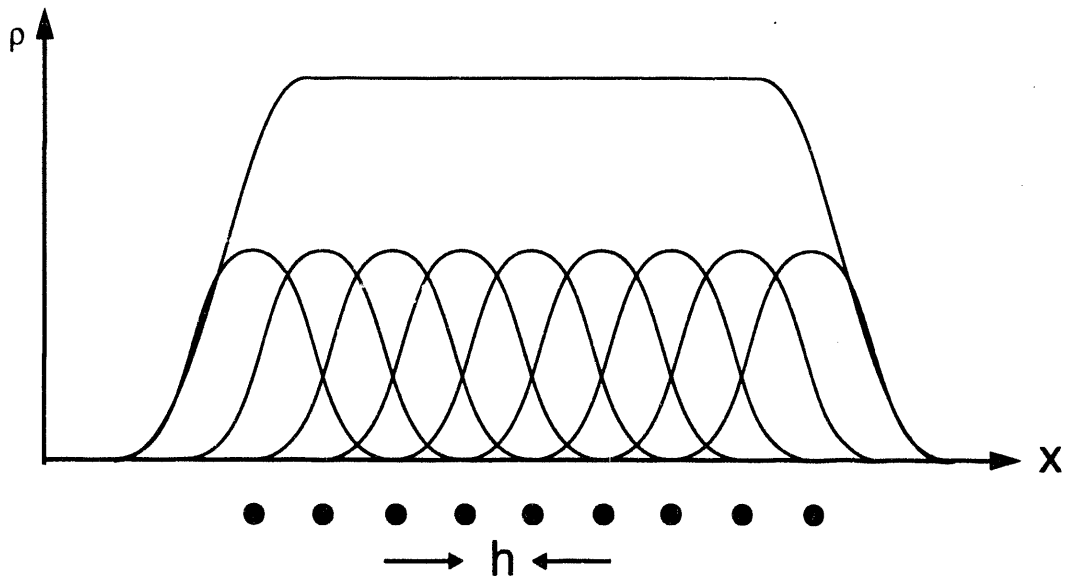


Figure 2.2 Kernel density from equally spaced mass points

the kernel density calculated from a set of equal masses with spacing h. The contribution from each mass point is represented by a curve which is an image of the kernel function. Thus, each point's mass seems to be spread in space over a distance $2h$, which is in fact the basis of the 'smoothed particle' terminology. However, it is more valid to think of the particles as interpolation points rather than interacting masses. The density at any position is the sum of the contributions from all points, with only those within a distance $2h$ providing a non-zero contribution. For equal masses with an equal spacing of h, the density in the interior is independent of position. However, the boundary particle sees a contribution from points on only one side, rather than from both sides as in the interior, and the calculated density is less. Equation (2.30) thus results in boundary anomalies unless special boundary treatments are implemented. An alternative method is to solve the continuity equation

$$\frac{\dot{\rho}}{\rho} = -\nabla \bullet \boldsymbol{V} = -\sum_{j=1}^{3} \frac{\partial V_j}{\partial x_j}. \tag{2.31}$$

Using Eq. (2.26) in a centered finite-difference temporal formulation of Eq. (2.31) results in

$$\rho^{I,n+1} = \rho^{I,n} - \Delta t^{n+\frac{1}{2}} \left\{ \sum_{J=1}^{N} m^J \sum_{j=1}^{3} (V_j^I - V_j^J) \frac{\partial W}{\partial x^J{}_j} \right\}^{n+\frac{1}{2}}. \tag{2.32}$$

This relation allows the initial density to be set to the desired value at each point, with density changes resulting only from relative motion between points. In practice, Eqs. (2.30) and (2.32) give identical results at interior points. The new densities at time t^{n+1} and the strain rates at time $t^{n+\frac{1}{2}}$ allow the equation of state to be evaluated at time t^{n+1}. Material strength is clearly included in the above formulation, since all components of the stress and strain rate tensors are treated. In fact, the equation of state can be completely arbitrary, especially when internal state variables are associated with each point. Given the Lagrangian nature of the points, tracking arbitrary internal state variables is a trivial process.

2.3 Artificial Viscosity

Since the differential equations expressing conservation of mass, momentum, and energy apply only to regions of continuous flow, an artificial viscosity[11] is required to smooth shocks into continuous steady waves and prevent the formation of mathematical discontinuities. The artificial viscous stress used in standard Lagrangian finite-difference methods[10] is

$$Q = \rho b_1^2 \left(\frac{\dot{\rho}}{\rho} \right)^2 + \rho b_2 C \left(\frac{\dot{\rho}}{\rho} \right), \tag{2.33}$$

where b_1 and b_2 are constants with dimensions of length, and C is the sound speed. To produce a shock width consistent with the numerical resolution, the dimensional constants are scaled by a characteristic resolution length, l, which in SPH should be related to the smoothing length, h. The viscous stress can then be written in the form

$$Q = \rho B_1^2 l^2 \left(\frac{\dot{\rho}}{\rho} \right)^2 + \rho B_2 l C \left(\frac{\dot{\rho}}{\rho} \right), \tag{2.34}$$

hereafter referred to as the finite-difference artificial viscosity, where B_1 and B_2 are dimensionless coefficients, which in standard finite-difference methods usually have the values $B_1 = 2$, $B_2 = 0.1$. Typically, the artificial viscosity is used only in compression, and Q is set to zero on expansion. However, in certain cases, such

as rarefaction shocks, it is necessary to turn the artificial viscosity on in expansion. A form of Eq. (2.34) which has the correct sign both in compression and expansion is

$$Q = \rho l\left(\frac{\dot{\rho}}{\rho}\right)\left(B_1^2 l\left|\frac{\dot{\rho}}{\rho}\right| + B_2 C\right). \tag{2.35}$$

The artificial viscosity is a mean stress which can be stored at each point and added to the stress tensor to form a total stress for use in the equations of motion. The equation of motion thus becomes

$$a_i^I = -\sum_{J=1}^{N} m^J \sum_{j=1}^{3}\left[\left(\frac{\sigma_{ij} - Q\delta_{ij}}{\rho^2}\right)^I + \left(\frac{\sigma_{ij} - Q\delta_{ij}}{\rho^2}\right)^J\right]\frac{\partial W}{\partial x^J_j}\,, \tag{2.36}$$

where $Q\delta_{ij}$ is subtracted from σ_{ij} since Q is positive in compression while σ_{ij} is positive in tension.

Previous SPH investigators have reported that the above form of the artificial viscosity is inadequate to prevent oscillations on the scale of the particle spacing, and an alternative SPH artificial viscosity formulation[12] is used which takes the form of an interparticle stress given by

$$Q^{IJ} = \rho^2 \Pi^{IJ}, \tag{2.37}$$

where

$$\Pi^{IJ} = \frac{\alpha C^{IJ}\mu^{IJ} - \beta(\mu^{IJ})^2}{\rho^{IJ}}\,, \tag{2.38}$$

$$\mu^{IJ} = \frac{h\sum_{j=1}^{3}(V_j^I - V_j^J)(x_j^I - x_j^J)}{(|x^I - x^J|)^2 + \varepsilon h^2}, \tag{2.39}$$

$$C^{IJ} = \frac{1}{2}(C^I + C^J)\,, \tag{2.40}$$

and

$$\rho^{IJ} = \frac{1}{2}(\rho^I + \rho^J) . \tag{2.41}$$

The dimensions of Q^{IJ} are stress with the same sign convention as σ_{ij} (negative in compression), while Π^{IJ} has dimensions of stress divided by ρ^2, and μ^{IJ} has dimensions of velocity. Normally, Eq. (2.38) only applies if $\mu^{IJ} < 0$, so that particles I and J are approaching. Otherwise, $\Pi^{IJ} = 0$. A modification which provides the correct sign for Π^{IJ} regardless of the sign of μ^{IJ} and thus allows the viscosity to be used when particles are separating, is

$$\Pi^{IJ} = \frac{\mu^{IJ}(\alpha C^{IJ} + \beta|\mu^{IJ}|)}{\rho^{IJ}} . \tag{2.42}$$

With this form of the viscosity, since Q^{IJ} cannot be written as a viscous stress to be added to the normal equation of state stress of each material element, the equation of motion becomes

$$a_i^I = -\sum_{J=1}^{N} m^J \sum_{j=1}^{3} \left[\left(\frac{\sigma_{ij}}{\rho^2} \right)^I + \left(\frac{\sigma_{ij}}{\rho^2} \right)^J + \Pi^{IJ}\delta_{ij} \right] \frac{\partial W}{\partial x^J{}_j} . \tag{2.43}$$

Although the differences between the two forms seem minor, especially since μ^{IJ} would appear to be closely related to the volumetric strain rate, they have quite different properties, as will be shown later.

2.4 Stability Criterion

Explicit numerical solution methods such as the one described here must restrict the time step to a maximum value for the solution to remain stable. The CFL (Courant, Friedrichs, Lewy)[13] criterion essentially reduces to

$$\Delta t \leq \frac{\Delta x}{C} , \tag{2.44}$$

where Δx is the smallest resolution length in the calculation. A stability criterion used for finite-difference methods[10] takes the form

$$\Delta t \leq \frac{k\Delta x}{B_2 C + 2B_1^2|\dot{\rho}/\rho|\Delta x + \sqrt{(B_2 C + 2B_1^2|\dot{\rho}/\rho|\Delta x)^2 + C^2}} , \tag{2.45}$$

where k is a safety factor usually equal to 0.9. This form includes modifications due to the artificial viscosity in the finite-difference form of Eq. (2.35). A stability criterion for the full SPH formulation including the viscosity in the SPH form of Eq. (2.38) has not been done, but a comparison of the coefficients in the two forms suggests that α corresponds to B_2, while β corresponds to B_1^2, at least on a dimensional basis.

2.5 Internal Energy

Evolution of the internal energy is described by the relation

$$\rho \dot{E} = \sigma \bullet \dot{\varepsilon} + \dot{E}_Q \tag{2.46}$$

where E is the specific internal energy, σ is the stress tensor, $\dot{\varepsilon}$ is the strain rate tensor, given by Eq. (2.21), and $\dot{E}_Q$ is the contribution from the viscous stress. In finite-difference form this becomes

$$E^{n+1} = E^n + \left(\frac{\sigma \bullet \dot{\varepsilon} + \dot{E}_Q}{\rho / (\Delta t)} \right)^{n+1/2} \tag{2.47}$$

Since the components of σ and $\dot{\varepsilon}$ are known at each particle I, the term $\sigma \bullet \dot{\varepsilon}$ can be evaluated independently at each particle without the involvement of kernel sums. If the finite-difference artificial viscosity is used, then

$$\dot{E}_Q = -Q\boldsymbol{I} \bullet \dot{\varepsilon}, \tag{2.48}$$

where I is the unit tensor, since Q is simply an additive stress. However, if the SPH viscosity is used, then[8]

$$\dot{E}_Q = \frac{1}{2} \sum_{J=1}^{N} m^J \Pi^{IJ} \sum_{j=1}^{3} (V_j^I - V_j^J) \frac{\partial W}{\partial x^J{}_j} . \tag{2.49}$$

For an energy-independent equation of state, Eq. (2.47) may be evaluated directly once the stress has been determined. For an energy-dependent equation of state, simultaneous solution of the equations of state and energy evolution is frequently possible. Determining the new stress and internal energy at all points completes the calculations for a given cycle, allowing the time to be incremented, and the process to begin again.

2.6 Kernel Function

In order to implement the SPH algorithm, the kernel function must be specified. Although numerous possibilities exist for the kernel function, one of the most widely used is the cubic b-spline kernel

$$W = \begin{cases} \frac{C}{h^D}\left[1 - \frac{3}{2}z^2 + \frac{3}{4}z^3\right] & z < 1 \\ \frac{C}{4h^D}[2 - z]^3 & 1 \le z < 2 \\ 0 & z \ge 2 \end{cases} \tag{2.50}$$

where

$$z = \frac{u}{h}, \tag{2.51}$$

u is the interparticle distance,

$$u = |x - x'|, \tag{2.52}$$

D is the dimension of the problem (1, 2, or 3), and the constant C is given by

$$C = \begin{cases} \frac{2}{3} & D = 1 \\ \frac{10}{7\pi} & D = 2 \\ \frac{1}{\pi} & D = 3 \end{cases} \tag{2.53}$$

The derivative of the kernel function is

$$\frac{\partial W(x - x')}{\partial x_j} = \frac{(x_j - x'_j)}{u}\frac{\partial W}{\partial u} = -\frac{\partial W(x - x')}{\partial x'_j}, \tag{2.54}$$

where

$$\frac{\partial W}{\partial u} = \begin{cases} \frac{3C}{h^{D+1}}\left[-z + \frac{3}{4}z^2\right] & z < 1 \\ \frac{-3C}{4h^{D+1}}[2 - z]^2 & 1 \le z < 2 \\ 0 & z \ge 2 \end{cases} \tag{2.55}$$

2.7 Variable Smoothing Length

In large deformation SPH calculations, it is clear that particle spacings may change dramatically from the initial configuration. In particular, if large expansions occur, particles may become separated by more than twice the smoothing length, h. In this case, the particles will no longer interact, and the calculation will break down. Thus, it is necessary to implement an algorithm to vary the smoothing length as strain accumulates. As indicated in Figure 2.3, the smoothing length may be

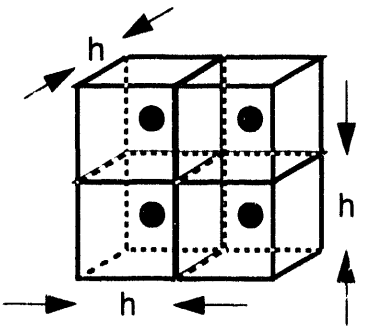


Figure 2.3 Particle volume

related to the particle size, or the volume occupied by the particle. If the particles are equally spaced with an inter-particle distance of h, the mass of the particle is related to its density and volume by

$$m = \rho h^D. \tag{2.56}$$

Thus, as the density of a fixed mass particle varies, one approach to implementing a variable smoothing length is

$$h \propto \left(\frac{m}{\rho}\right)^{1/D}. \tag{2.57}$$

This approach will not avoid difficulties in every situation, as illustrated in Figure 2.4. Here a constant volume stretching is taking place, so that the particle

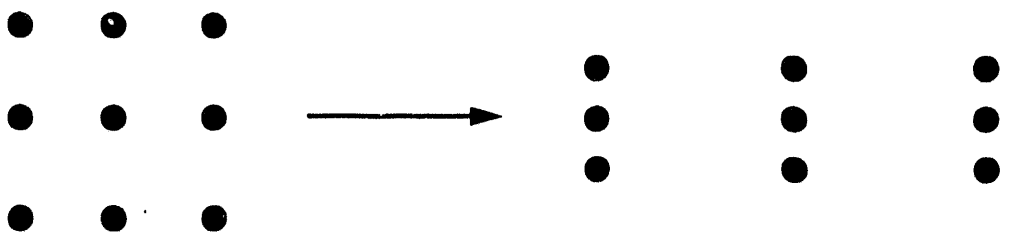


Figure 2.4 Constant volume stretching deformation

separation is increasing in one direction and decreasing in the other, while the density remains constant. However, the variable smoothing length algorithm described below will remain essentially unchanged if Eq. (2.57) is replaced by a dependence on a different strain measure.

In the general case in which the smoothing length varies from particle to particle, symmetry and thus conservation of momentum is maintained if the smoothing length used for interactions between particles I and J is taken to be

$$h^{IJ} = \frac{(h^I + h^J)}{2}. \tag{2.58}$$

Variable smoothing is easily implemented in the equation of motion, since only quantities at time n are involved, and complete information at this time is available. With variable smoothing, Eq. (2.24) becomes

$$a_i^{I,n} = -\left\{ \sum_{J=1}^{N} m^J \sum_{j=1}^{3} \left[\left(\frac{\sigma_{ij}}{\rho^2} \right)^I + \left(\frac{\sigma_{ij}}{\rho^2} \right)^J \right] \frac{\partial W(x^I - x^J, h^{IJ})}{\partial x^J{}_j} \right\}^n, \tag{2.59}$$

where

$$h^{I,n} = \left(\frac{m^I}{\rho^{I,n}} \right)^{1/D}. \tag{2.60}$$

Calculation of the new velocity gradient and density is less straightforward. If the smoothing length is constant in time, density is evolved according to the continuity equation, which in centered finite-difference form becomes

$$\left(\frac{\dot{\rho}}{\rho} \right)^{I,n+\frac{1}{2}} = \frac{(\rho^{I,n+1} - \rho^{I,n})/\Delta t^{n+\frac{1}{2}}}{\rho^{I,n+\frac{1}{2}}} = -\nabla \bullet V = -\sum_{j=1}^{3} \left(\frac{\partial V_j}{\partial x_j} \right)^{I,n+\frac{1}{2}}, \tag{2.61}$$

where the velocity gradient is given by

$$\left(\frac{\partial V_i}{\partial x_j} \right)^{I,n+\frac{1}{2}} = \frac{1}{\rho^{I,n+\frac{1}{2}}} \left\{ \sum_{J=1}^{N} m^J (V_i^I - V_i^J) \frac{\partial W(x^I - x^J, h^{IJ})}{\partial x^J{}_j} \right\}^{n+\frac{1}{2}}. \tag{2.62}$$

Inserting Eq. (2.62) in Eq. (2.61) and multiplying both sides by $\rho^{I,n+\frac{1}{2}}$ yields

$$\rho^{I,n+1} = \rho^{I,n} - \Delta t^{n+\frac{1}{2}} \left\{ \sum_{J=1}^{N} m^J \sum_{j=1}^{3} (V_j^I - V_j^J) \frac{\partial W(x^I - x^J, h^{IJ})}{\partial x^J_j} \right\}^{n+\frac{1}{2}} . \quad (2.63)$$

Note that the expression for the velocity gradient requires the density at the new time, since

$$\rho^{I,n+\frac{1}{2}} = \frac{1}{2}(\rho^{I,n+1} + \rho^{I,n}) . \quad (2.64)$$

However, the cancellation of the $\rho^{I,n+\frac{1}{2}}$ term allows the new density, which depends on the velocity gradient, to be calculated explicitly, rather than implicitly, which would be the case if terms on the right side of Eq. (2.63) contained the new density. In practice, the calculational procedure involves determination of the partial sums

$$\varphi_{ij}^{I,n+\frac{1}{2}} = \left\{ \sum_{J=1}^{N} m^J (V_i^I - V_i^J) \frac{\partial W(x^I - x^J, h^{IJ})}{\partial x^J_j} \right\}^{n+\frac{1}{2}}, \quad (2.65)$$

which then allows the new density to be determined from

$$\rho^{I,n+1} = \rho^{I,n} - \Delta t^{n+\frac{1}{2}} \sum_{i=1}^{3} \varphi_{ii}^{I,n+\frac{1}{2}}, \quad (2.66)$$

while the velocity gradient is given by

$$\left(\frac{\partial V_i}{\partial x_j} \right)^{I,n+\frac{1}{2}} = \varphi_{ij}^{I,n+\frac{1}{2}} / \rho^{I,n+\frac{1}{2}} . \quad (2.67)$$

Unfortunately, in the general variable smoothing case, the smoothing length varies in time, so that the partial sums defined in Eq. (2.65) require the smoothing length at time $n + 1/2$, which is given by

$$h^{I,n+\frac{1}{2}} = \left(\frac{m^I}{\rho^{I,n+\frac{1}{2}}} \right)^{1/D} \tag{2.68}$$

in the present case of a density-based variable smoothing length, or in the more general case depends on some other strain measure. In either case, Eqs. (2.65) to (2.67) become highly implicit.

Several solution methods exist for this implicit set of equations. The most accurate and also the most time consuming would be to iterate, using the current value of h to calculate the new density, determining an improved value of h from Eq. (2.68), and recalculating the new density until convergence is achieved. A more expedient solution would simply be to use the old value of h, which basically assumes that the above iteration converges sufficiently in a single step. An alternate method would be to estimate the required value of h at $n + 1/2$ by the following procedure. Assume that the density increment does not change significantly from time $n - 1/2$ to time n, so that

$$\delta\rho^n = \rho^{n+\frac{1}{2}} - \rho^{n-\frac{1}{2}} \approx \delta\rho^{n-\frac{1}{2}} = \rho^n - \rho^{n-1} \quad . \tag{2.69}$$

An estimated value of the density can then be found from

$$\rho_{est}^{n+\frac{1}{2}} = \frac{1}{2}(3\rho^n - \rho^{n-1}) \ , \tag{2.70}$$

allowing the value of h at $n + 1/2$ required in Eqs. (2.65) to (2.67) to be estimated from

$$h_{est}^{I,n+\frac{1}{2}} = \left(\frac{m^I}{\rho_{est}^{I,n+\frac{1}{2}}} \right)^{1/D} \ . \tag{2.71}$$

3. Weighted Residual Derivation for Smoothed Particle Hydrodynamics

Here, we derive the equations for smoothed particle hydrodynamics (SPH) using a weighted residual approach. This derivation follows the approach used in a displacement based finite element method. Trial and test functions of the same class as the displacement are used in the formulation. A spherical basis function is used to interpolate the displacements and velocity.

The weighted residual derivation presented here generates a consistent mass matrix for the SPH technique. When this consistent mass matrix is lumped, then the current method reduces to the classical formulation for SPH. The weighted residual derivation also generates the terms required for prescribed boundary tractions. These traction terms can be reduced to the same form as derived by Campbell. [14]

3.1 Displacement and Strain Approximation

In SPH, the displacement at a point is approximated using kernel estimates. The displacement kernel estimate starts from the identity

$$\underline{u}(\underline{r}) = \int \underline{u}(\underline{r}')\,\delta(\underline{r}-\underline{r}')\,d\underline{r}', \tag{3.1}$$

where $\underline{r}$ is the three-dimensional position vector and $\delta(\underline{r}-\underline{r}')$ is the Dirac delta function. If $\delta(\underline{r}-\underline{r}')$ is replaced by a kernel function $W(\underline{r}-\underline{r}', h)$ where h is a smoothing length, then a kernel approximation for the displacement is

$$\underline{u}(\underline{r}) \approx \int_{\Omega} \underline{u}(\underline{r}')\,W(\underline{r}-\underline{r}', h)\,d\underline{r}'. \tag{3.2}$$

Two constraints on the form of the kernel function W are that it should reduce to the delta function as h goes to zero,

$$\lim_{h\to 0} W(\underline{r}-\underline{r}', h) = \delta(\underline{r}-\underline{r}'), \tag{3.3}$$

and that

$$\int_{\Omega} W(\underline{r}, h)\,dV = 1. \tag{3.4}$$

In addition, it is desirable to have compact support for W, i.e.

$$W(\underline{r}, h) = 0 \quad \text{for} \quad |\underline{r}| \geq 2h. \tag{3.5}$$

3.2 Polynomial Kernel Function

A polynomial kernel function W is introduced for an SPH particle of the following form:

$$W(\underline{r}, h) = \begin{bmatrix} \frac{1.5}{0.7\pi h^2}\left(\frac{2}{3} - z + \frac{z^2}{2}\right) & (0 < z < 1) \\ \frac{0.25}{0.7\pi h^2}(2-z)^3 & (1 < z < 2) \\ 0 & (2 < z) \end{bmatrix}, \tag{3.6}$$

where $z = \frac{|\underline{r}|}{h}$. In the following derivation, the shorthand notation: $W^I = W(\underline{r} - \underline{r}^I, h)$, $W^{IJ} = W(\underline{r}^{IJ}, h) = W(\underline{r}^I - \underline{r}^J, h)$, and $\underline{u}^J = \underline{u}(\underline{r}^J)$ will be used. The superscript J refers to a neighboring particle of particle I.

The kernel function used in Equation (3.6) was selected because it is used in classical SPH methods and it satisfies the constraints expressed in Equations (3.3) and (3.4). The displacements, velocity and acceleration are interpolated using the kernel basis function W in Equation (3.6).

3.3 Volume Weighted Sum Integral Approximation

The volume integral in Equation (3.2) is approximated by a volume weighted sum at particle I

$$\underline{u}(\underline{r}^I) = \sum_{J \in h_i} \underline{u}(\underline{r}^J)\, W(\underline{r}^{IJ}, h)\, \Delta V^J, \tag{3.7}$$

where ΔV^J is the volume of particle J, and the sum is over all particles within the smoothing length h_i of particle I. The volume of a particle is computed from its mass, m, and the density, ρ, at the particle

$$\Delta V^J = \frac{m^J}{\rho^J}. \tag{3.8}$$

For a uniform initial grid, the mass associated with a node can be computed from the total number of particles within a volume of known mass. The velocity and acceleration can be computed using the same volume-weighted sum approximation

$$\ddot{\underline{u}}^I = \sum_{J \in h_j} \ddot{\underline{u}}^J\, W^{IJ}\, \Delta V^J. \tag{3.9}$$

3.4 Velocity gradient

In a continuum, the gradient of the velocity is needed to determine the stretch. The spatial gradient is computed using a kernel approximation by substituting $\nabla \underline{u}(\underline{r})$ for $\underline{u}(\underline{r})$ in Equation(3.2)

$$\nabla \underline{u} \approx \int \nabla \underline{u}(\underline{r}') W(\underline{r} - \underline{r}', h)\, dV' \quad . \tag{3.10}$$

Green's theorem and Equation (3.5) can be used to transform Equation (3.10) to

$$\nabla \underline{u}(\underline{r}) \approx -\int_{\Omega} \underline{u}(\underline{r}') \nabla W(\underline{r} - \underline{r}', h)\, dV' \quad . \tag{3.11}$$

Using a volume weighted sum to replace the volume integral, the velocity gradient at particle *I* is approximated in SPH by

$$\nabla \underline{u}^I = \sum_{J \in h_I} \left(\underline{u}^J - \underline{u}^I \right) \nabla W^{IJ} \, \Delta V^J , \tag{3.12}$$

where $\left(\underline{u}^J - \underline{u}^I \right)$ is used to insure a symmetric system and $\nabla W^{IJ} = \frac{\partial}{\partial r_i} W(r_I - r_J)\, \underline{i}$.

3.5 Weak Form of Linear Momentum Balance

The differential form of the linear momentum balance, can be expressed in a weak form by multiplying by a weighting function, $\delta \underline{u}$,

$$\int_{\Omega} \delta \underline{u} \left(\nabla \bullet \underline{\sigma} + \rho \ddot{\underline{u}} + \underline{f}_b \right) dV = 0 . \tag{3.13}$$

Assume, for now, that the body force, f_b, is zero, so that

$$\int_{\Omega} \delta \underline{u} \left(\nabla \bullet \underline{\sigma} + \rho \ddot{\underline{u}} \right) dV = 0 . \tag{3.14}$$

Integration by parts of Equation (3.14) gives

$$\int_{\Omega} \delta \underline{u}\, \rho\, \ddot{\underline{u}}\, dV - \int_{\Omega} \underline{\sigma} \bullet \nabla \delta \underline{u}\, dV + \int_{\partial \Omega} \underline{t}\, \delta \underline{u}\, dS = 0 , \tag{3.15}$$

where $\underline{t} = \underline{\sigma} \cdot \underline{n}$ is the traction on surface with normal $\underline{n}$. To obtain a weighted residual statement for the discrete system, substitute Equation (3.7) into Equation (3.15),

$$R_u = \int_\Omega \rho \ddot{\underline{u}} \left(\Sigma \delta \underline{u}^I W^I \Delta V^I\right) dV \tag{3.16}$$

$$-\int_\Omega \underline{\sigma} \bullet \left(\Sigma \delta u^I \nabla W^J \Delta V^J\right) dV + \int_{\partial\Omega} \underline{t} \left(\Sigma \delta \underline{u}^I W^J \Delta V^J\right) dS \cdot$$

The residual at each node, *I*, can be minimized by imposing

$$\frac{\partial R_u}{\partial \delta \underline{u}} = 0. \tag{3.17}$$

For node *I*, Equation (3.17) gives

$$\int \rho \ddot{\underline{u}} W^I \Delta V^I dV - \int \underline{\sigma} \nabla W^J \Delta V^I dV + \int \underline{t} W^I \Delta V^I dS = 0 \quad . \tag{3.18}$$

Note that the integral in Equation (3.18) is over the total volume; however, $W^I = 0$ and $\frac{\partial W^I}{\partial \underline{r}} = 0$ for $|\underline{r}| > h$, so that Equation (3.18) can be approximated using a volume weighted sum

$$\int_\Omega \rho \ddot{\underline{u}} W^I \Delta V^I dV \approx \sum_J \rho^J \ddot{\underline{u}}^J W^{IJ} \Delta V^I \Delta V^J , \tag{3.19}$$

$$\int_\Omega \underline{\sigma} \frac{\partial W^I}{\partial \underline{r}} \Delta V^I dV \approx \sum_J \underline{\sigma}^J \nabla W^{IJ} \Delta V^I \Delta V^J , \tag{3.20}$$

where ΔV^J is the volume contribution of node *J*

$$\Delta V^J = \frac{m^J}{\rho^J}. \tag{3.21}$$

The mass m^J at a node is fixed and the volume contribution to the integral due to node *J* changes as the density changes.

Likewise, the integral along the boundary with tractions can be approximated by

$$\int_{\partial\Omega} \underline{t} W^I \Delta V^I dS \approx \sum_J \underline{t} W\left(\underline{r}^I - \underline{r}^B\right) \Delta V^I \Delta S^J , \tag{3.22}$$

where r^B is the shortest distance from the boundary to the particle *I*, and ΔS^J is the surface intercept along the prescribed boundary. Campbell [14] computes ΔS based on the intersection of a sphere with a plane as

$$\Delta S^J = C(\alpha)\left(h^2 - \left|\underline{r}^J - \underline{r}^B\right|^2\right)^{\frac{\alpha-1}{2}}, \tag{3.23}$$

where α = 1,2,3 is the dimension of the problem and $C(\alpha)$ = (1,2,π). Substituting Equations (3.19), (3.20) and (3.22) into Equation (3.18) gives a total of N equations with each node I having an equation of the form

$$\sum_{J \in h_I} \rho^J W^{IJ} \Delta V^I \Delta V^J \ddot{\underline{u}} = \sum_{J \in h_I} \underline{\sigma}^J \nabla W^{IJ} \Delta V^I \Delta V^J \tag{3.24}$$

$$- \sum_{J \in h_I} \underline{t} W^{IJ} \Delta V^I \Delta S^J .$$

The interacting particles $J \in h_I$, i.e. those particles that are within the smoothing length h_I of particle I, are determined by a particle sort routine (described later). Equation (3.24) can be assembled to give a mass matrix and force vector

$$\boldsymbol{M}\,\ddot{\boldsymbol{u}} = \boldsymbol{F}^{\text{ext}} - \boldsymbol{F}^{\text{int}} . \tag{3.25}$$

For dynamic problems, Equation (3.25) may be integrated forward in time using a central difference or other time integrating scheme.

In the classical SPH techniques, the mass matrix is lumped. In the current derivation of the SPH equations using a weak form, the particle mass matrix in Equation (3.25) leads to a consistent mass matrix. Numerical tests need to be performed to determine whether a lumped or a consistent mass matrix will give more accurate results. A consistent mass has the disadvantage that it will require a matrix inversion for each time step.

The individual components of a consistent mass matrix are given by

$$M^{IJ} = \rho^J W^{IJ} \Delta V^I \Delta V^J . \tag{3.26}$$

For a lumped mass matrix, the initial particle mass may be determined from the density at the particle and the initial volume of the particle (both are supplied as input to SPH).

The components of the internal force vector are given by

$$m^I \frac{d\underline{v}}{dt} = \boldsymbol{F}^I = \sum_{J \in h_I} \underline{\sigma}^J \nabla W^{IJ} \Delta V^I \Delta V^J . \tag{3.27}$$

Equation (3.27) can be made symmetric by adding the constant σ^I to the sum at node I

$$\underline{F}^I = \sum_{J \in h_I}^{K} \left(\underline{\sigma}^J + \underline{\sigma}^I \right) \nabla W^{IJ} \, \Delta V^I \, \Delta V^J . \tag{3.28}$$

If the smoothing length used for particle I and J are equal, $h^I = h^J$, then a symmetric system of equations will result since $W^{IJ} = W^{JI}$ and $\frac{\partial W^{IJ}}{\partial \underline{r}} = \frac{\partial W^{JI}}{\partial \underline{r}}$. In addition, the system will be banded since local support is insured by $W^{IJ} = 0$ and $\frac{\partial W^{IJ}}{\partial \underline{r}} = 0$ when $r^{IJ} > h$.

3.6 Summary of SPH Equations of Motion

In the above section, the SPH equations were derived using an approach that parallels the derivation of the classical displacement based finite element method. The derivation starts with an approximating function for the displacements using a kernel sum. A weighted residual form of the momentum balance condition, using a weighting function of the same class as the displacement function, was minimized to form a system of equations that can be integrated through time. A key to obtaining the classical SPH equations is the use of a volume weighted sum to perform the spatial integration.

From this perspective, the SPH method can be viewed as a special case of the finite element method, where the connectivity of the element is constructed by a search for the nearest neighbors. In addition to generating a consistent mass matrix, the derivation suggests several areas for further work.

The accuracy of the approximating function will depend on the accuracy of the kernel sum. Replacing the volume weighted sum used in the spatial integration with a more accurate numerical integration may improve the method. Just as with successful finite element methods, the SPH method must pass the equivalent of a patch test and be proven to be stable and convergent.

4. Tests of the SPH Algorithm

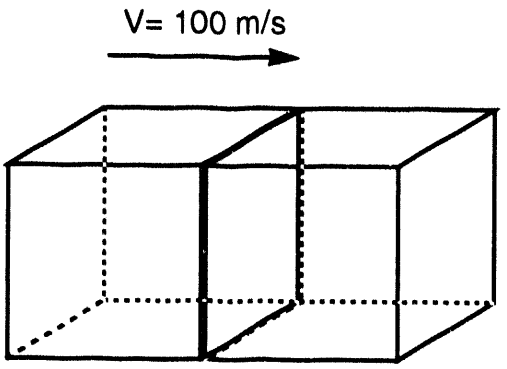
The SPH algorithm has been implemented in a test code with one-, two-, and three-dimensional options in rectangular coordinates. Tests started with very basic investigations into the adequacy of the algorithm, and proceeded to examinations of the ability of the method to solve simple wave propagation problems and complex multi-dimensional impacts. Of particular interest in the current evaluation of the SPH method is its adequacy under conditions of tensile stress, fracture, and fragmentation. Before attempting to implement sophisticated fracture and fragmentation models into SPH, it seems prudent to investigate the behavior of the method under tensile conditions if no fracture models have been implemented, so that there is no limit to the magnitude of the tensile stresses generated by the EOS (Equation of State). In other words, determine whether bodies will hold together if the physical input parameters to the EOS prohibit fracture. Several tests were designed to address this issue.

4.1 Gradient Approximation

Since the crux of the method, and the major difference from standard finite-difference algorithms, is the kernel approximation for spatial gradients, the first question was whether the correct gradients were obtained. That is, apply a constant stress gradient and see if the correct acceleration is obtained, or apply a constant velocity gradient and see if the correct strain rate is obtained. Results were obtained for equal masses with a spacing of h, either on a line in 1D, a square lattice in 2D, or a cubic lattice in 3D. The following results apply only to this particular equally-spaced configuration, and can be verified analytically. In 1D, the exact result is obtained for the gradient. However, in 2D, the gradient is high by a factor of 1.0131, and in 3D it is high by 1.02004. These errors result from the normalization constant C in Eq. (2.53) being determined by a continuous integral over the domain of the kernel function, as in Eq. (2.5), rather than for the discrete lattices of the test problem. Given the various other choices for particle spacing, such as hexagonal close-packed or random, as well as various choices for the kernel function itself, it is clear that the kernel approximation is indeed an approximation, and that numerically exact answers to simple test problems will only be obtained in very special cases. The adequacy of the kernel approximation in more general cases will be addressed in later test problems.

4.2 Low Velocity 1D Impact - Bodies in Contact

Figure 4.1 illustrates a simple 1D shock problem having a known analytic solution. A 1 cm thick block of material impacts an identical block at 100 m/s. This is a low stress problem with an impact stress of 8 kbar and less than one percent strain, so that a constant, rather than variable, smoothing length is appropriate, and the linear regime of the EOS is tested. The materials are initially in contact, and 201 particles are placed along a one-dimensional line with an interparticle spacing of 0.01 cm, equal to the smoothing length, h. The initial velocities are 100 m/s for



Material Properties

Density, ρ_0	2.7 g/cm^3
C_0	5.38 km/s
S	1.337
Gruneisen Parameter, Γ	0.
Poisson's Ratio, ν	0.1
Yield Strength, Y	2 kbar

Figure 4.1 Symmetric impact with initial contact

particles 1-100, 50 m/s for particle 101, and 0 for particles 102-201. An elastic-perfectly-plastic equation of state was used with material constants representative of aluminum. The pressure is given by the Mie-Gruneisen equation,

$$P = P_H + \rho\Gamma (E - E_H) , \tag{4.1}$$

where Γ is the Gruneisen parameter, P_H is the pressure on the Hugoniot, given by

$$P_H = \frac{\rho C_0^2 \eta}{(1 - S\eta)^2} , \tag{4.2}$$

and E_H is the energy on the Hugoniot, given by

$$E_H = \frac{P_H}{2\rho_0} . \tag{4.3}$$

The material properties are shown in Figure 4.1. The Gruneisen parameter was chosen to be zero to remove the energy dependence. The 2 kbar yield strength and the constant value of 0.1 for Poisson's ratio were chosen to increase the separation

of the elastic precursor and the plastic wave, rather than because these are the most appropriate values for aluminum. The finite-difference form of the artificial viscosity was used. Figure 4.2 shows the SPH stress profiles compared to the

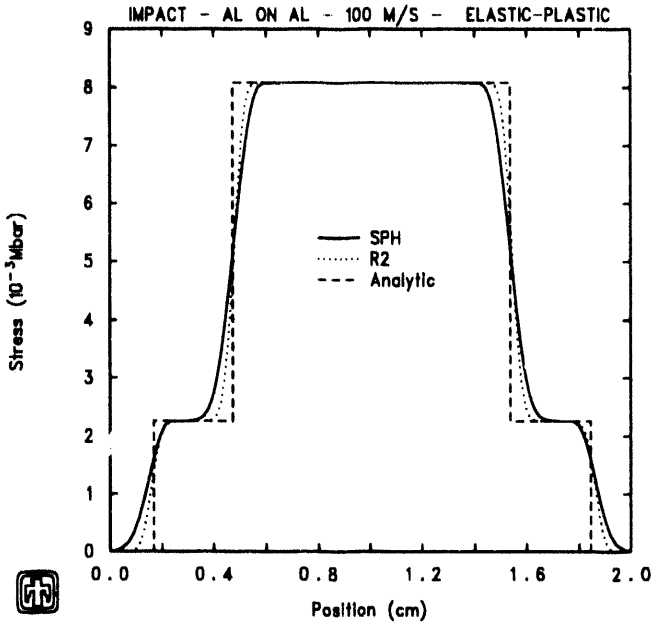


Figure 4.2 Contact impact stress profiles

analytic solution and to a solution generated by TOODY[10], a standard Lagrangian finite-difference wavecode. The time is just prior to reflection from the free surfaces of the shocks generated in the impactor and target.The comparison shows that the method is capable of handling shocks using the standard form of the artificial viscosity, and the appearance of an elastic precursor and a plastic wave shows that material strength is included correctly. Figure 4.3 shows particle velocity profiles after reflection of the shocks from the free surfaces, so that rarefactions are proceeding back into the materials. The agreement shows that the method correctly handles wave reflections from free surfaces without any special boundary treatments. The agreement is especially good considering that a complex elastic-plastic wave interaction process occurs at the interface, and the interface velocity is not equal to twice the particle velocity behind the shock, as it would be for hydrodynamic materials. This calculation shows that for equally spaced particles in 1D, the method is capable of reproducing standard finite-difference results, with waves which are only slightly more dispersed.

4.3 Artificial Viscosity and Time Step

The above impact problem is also appropriate for evaluating the different forms of the artificial viscosity and their interaction with the stability criterion. Calculations show that either form of the artificial viscosity is capable of spreading and smoothing shock fronts. The standard values $B_1 = 2$, $B_2 = 0.1$, are adequate for

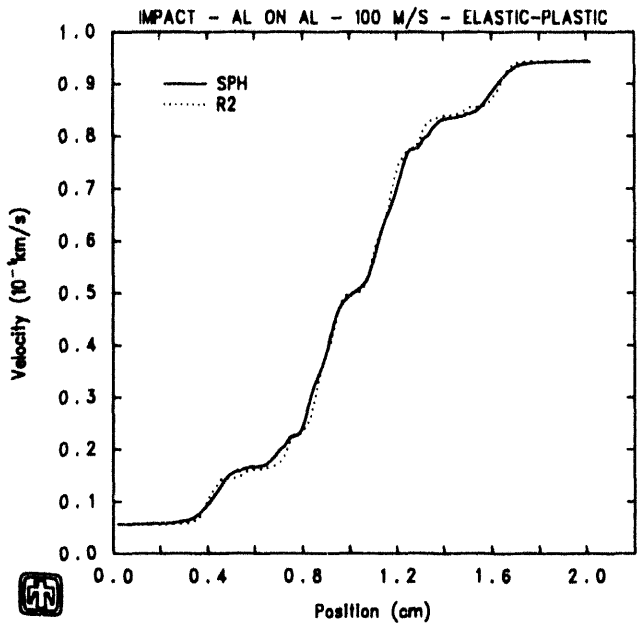


Figure 4.3 Contact impact velocity profiles

the finite-difference viscosity if the scaling length is taken to be $2h$. Standard values of α and β for the SPH viscosity have apparently yet to be determined. However, calculations indicate that the linear term, α, needs to be on the order of 1 or greater, while the quadratic term, β, has very little effect on the solution if varied in the range from 0.5 to 5. Acceptable ranges seem to run from $\alpha = 1$, $\beta = 2$, to $\alpha = \beta = 2.5$. The fact that the linear and quadratic coefficients are of the same order is in contrast to the finite-difference case in which the linear coefficient is carefully kept at least an order of magnitude smaller than the quadratic to avoid distorting the solution.

The finite-difference time step criterion, Eq. (2.45), works well for the SPH algorithm with the finite-difference form of the viscosity, but with the important difference that Δx can be taken to be the smoothing length, h, rather than the minimum distance between particles, as would be expected by analogy with finite-difference methods. This means that calculations with exactly the same particle spacing but twice the smoothing length will run stably with twice as large a time step. Calculations run with the finite-difference form of the viscosity, $\Delta x = h$, and a safety factor, k, of 0.9 show no stability problems.

As pointed out previously, dimensional comparisons of the finite-difference and SPH forms of the viscosity indicate that α corresponds to B_2, while β corresponds to B_1^2. However, if these values are substituted into the stability criterion, the large numerical value required for α results in a severe limitation on the time step

generated by Eq. (2.45). Test calculations indicate that stability is obtained if Δx is set equal to h, B_1 is set equal to β, and $\alpha/3$ is substituted for B_2 when the SPH form of the viscosity is used.

4.4 Low Velocity 1D Impact - Separated Bodies

The next test involved the same problem, except that an extra gap of 0.01 cm was inserted between the particles representing the impactor and target. The nominal spacing between particles was already 0.01cm, so that the spacing in the gap was 0.02 cm, or $2h$, and there was initially no interaction between the impactor and target. For the impact velocity of 100 km/s, it takes 1 microsecond to close the gap from 0.02cm to 0.01cm. The situation thus represents an impact between two initially separated bodies, such as would be encountered when debris from an initial event impacts on secondary bodies. This case also tests the algorithms when particles are not perfectly spaced. Figure 4.4 shows results of this test at a time of

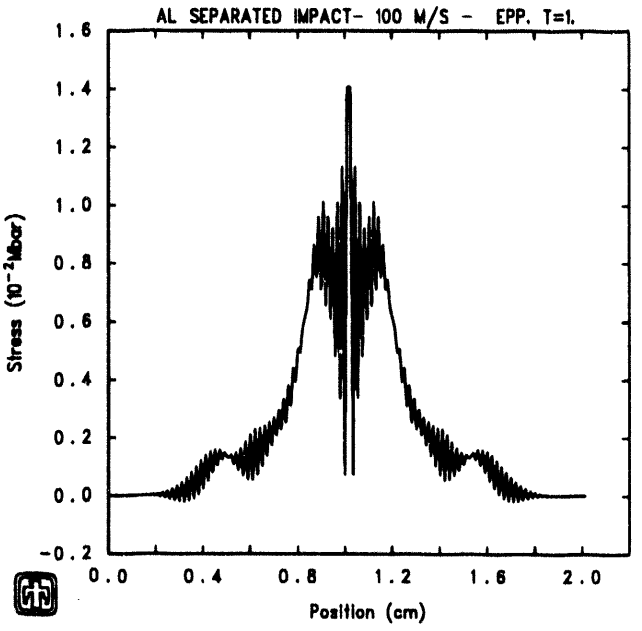


Figure 4.4 Separated impact stress profiles-t=1 microsecond

1 microsecond, while Figure 4.5 shows results at 2 microseconds. In contrast to the results of the previous test, the method does not handle the separated impact problem well, in that it fails to properly equilibrate the large stress fluctuations, sometimes referred to as noise, generated at the interface. Various unsuccessful attempts were made to remove the noise by varying the time step criterion, the artificial viscosity coefficients, and the smoothing length. Further, replacing the finite-difference viscosity with the SPH form failed to eliminate the noise.

While the extreme difference between the performance of the SPH algorithm on the contact and separated impact calculations may at first seem surprising, there

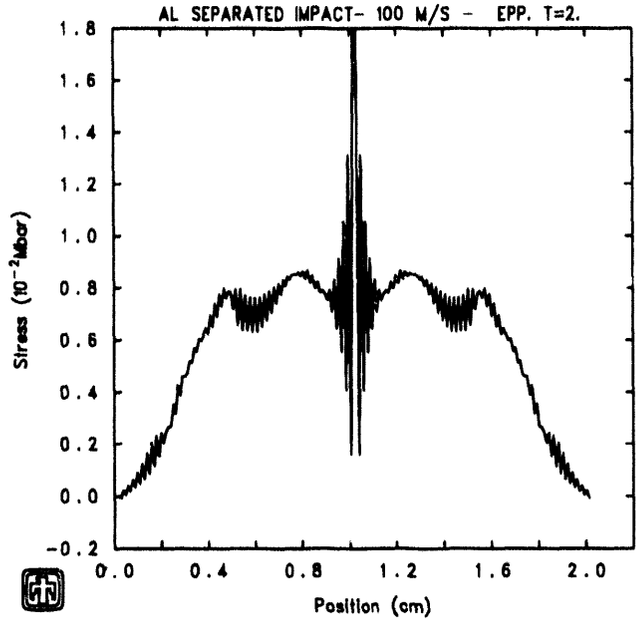


Figure 4.5 Separated impact stress profiles-t=2 microseconds

is in fact a difference between the physical situations being represented. The 'smoothed particle' nature of the method, as illustrated in Figure 2.2, means that boundaries and interfaces are not sharp. Just as there is a contribution to the kernel density which extends beyond the particles at the extremes of the body, there is also a contribution to the force that a given particle exerts on other particles which extends beyond boundary particles. The 'stress field' of a single particle is an image of the derivative of the kernel function, and thus, for the b-spline kernel, is zero at $2h$, and increases quadratically as the distance to the particle decreases. There is in effect a force between the particles at the boundaries of the impactor and target which is initially much smaller than the force between interior particles. In a sense, there is a very weak material in the gap which results in a large impedance mismatch between the aluminum bodies and the 'gap material.' If this physical situation is represented in a finite-difference wavecode calculation by adding a single zone of low density gas between the impacting bodies, the result shown in Figure 4.6 is obtained. The large impedance mismatch creates an inherently noisy situation, with large stress fluctuations, or ringing, at the interface. However, in the finite-difference solution, the ringing is damped out with time.

In contrast, the large stress fluctuations at the interface are essentially frozen in place in the smoothed particle calculation. This is non-physical, since any stress gradient should produce propagating waves which reduce the gradient. The reason this does not happen in the smoothed particle calculation is indicated in Figure 4.7 , which shows a close look at the stress field in the neighborhood of the impact interface. The points are the stresses at the particle positions, taken from the SPH calculation shown in Fig. (2.11), while the line is the kernel approximation

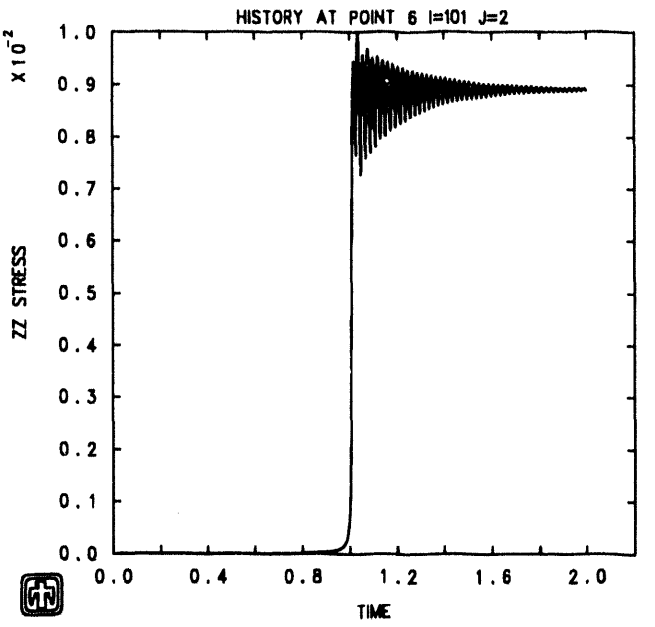


Figure 4.6 Finite-difference interfacial stress history for air-filled gap impact

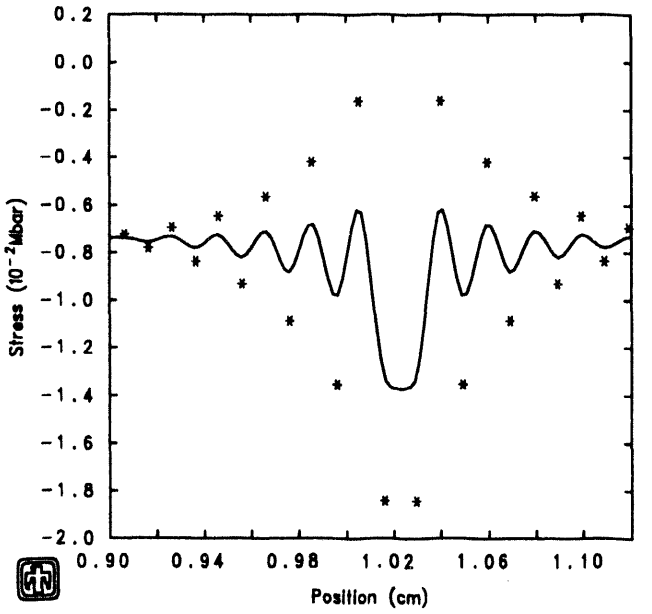


Figure 4.7 Stress values and kernel fit

to these discrete values calculated from a relation of the form of Eq. (2.14), using the discrete stress points as the values of the function f. A different sign convention is used in this figure than in the previous ones, so that stress is negative rather than positive in compression. The figure shows that there are large fluctuations in the

discrete stresses at the points, and the kernel approximation, which is after all just a polynomial fit to the data points, is trying to follow the oscillations. The problem is that the fitted function turns out to have its maxima and minima at the locations of the particles, so that the stress gradients, and thus the accelerations, are zero at the particle positions. This result is shown even more clearly in Figure 4.8, in

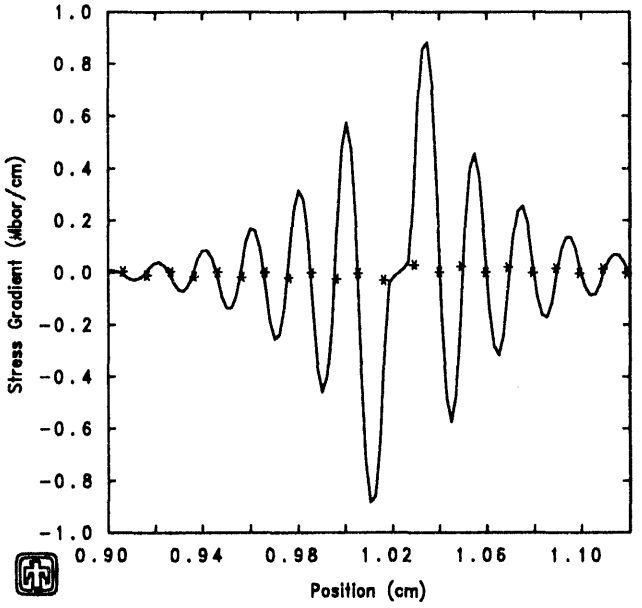


Figure 4.8 Stress gradient and kernel fit

which the points are the accelerations (proportional to the stress gradient) as calculated from Eq. (2.24), and the continuous line is the fit given by Eq. (2.23). Thus, the line in Figure 4.8 is the derivative of the line in Figure 4.7. Even though there are large stress gradients in the kernel fit to the stress, the gradients are essentially zero at the particles, which are the only locations at which the stress gradient is applied. There are thus no forces to accelerate the particles and reduce the stress fluctuations.

The situation described above results from the fact that the smoothed particle method does not constitute a staggered centered difference algorithm for spatial derivatives like, for instance, the von Nuemann-Richtmyer[11] (VNR) scheme. The differential equation of motion in one dimension is

$$\ddot{x} = \frac{1}{\rho}\frac{\partial \sigma}{\partial x} \qquad (4.4)$$

Consider the solution of this equation using the VNR finite-difference method as compared to the smoothed particle method. As illustrated in Figure 4.9, the finite-difference scheme has positions, velocities, and accelerations located at the

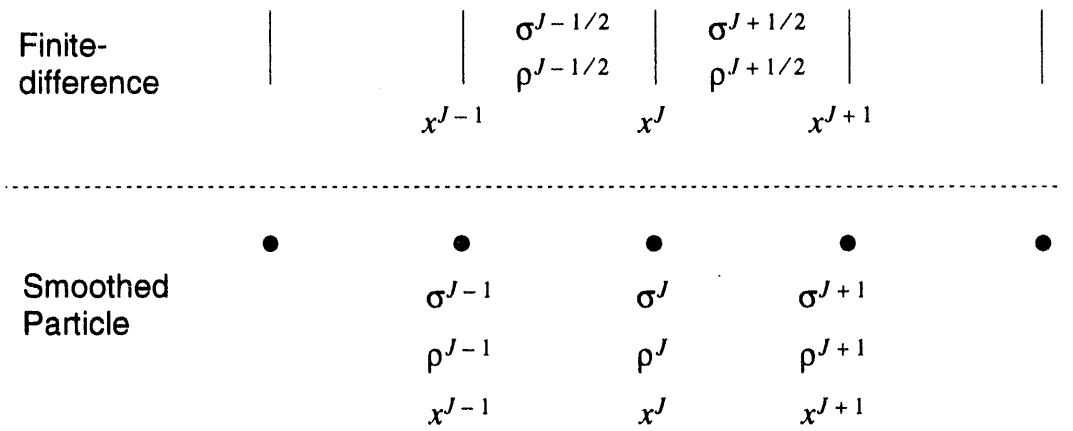


Figure 4.9 Comparison of finite-difference and smoothed particle schemes

nodes, represented by the vertical lines. All other quantities are located at the mesh centers between the nodes. In the smoothed particle method, all quantities are located at the particle positions. Assume that in both cases, the uniform spacing is h and all densities are equal to ρ. The finite-difference relation for the acceleration of node J is then[15]

$$\ddot{x}^J = \frac{1}{\rho}\left(\frac{\sigma^{J+1/2} - \sigma^{J-1/2}}{h}\right) \tag{4.5}$$

while the smoothed particle relation reduces to

$$\ddot{x}^J = \frac{1}{\rho}\left(\frac{\sigma^{J+1} - \sigma^{J-1}}{2h}\right). \tag{4.6}$$

Although both these relations use quantities centered about x^J, the finite difference relation uses the stresses in the two meshes directly adjacent to the node. These are consecutive meshes which are separated by h. However, in the smoothed particle method the stresses are located at the particle positions rather than staggered in space between the particles. Equation (4.6) uses stresses at particles on either side of the node, which are separated by $2h$, and does not use the stress at the node itself. The equation shows that, in SPH, no acceleration will be produced at a given particle if the stresses at the two neighbor particles are equal, regardless of the value of the stress at the particle itself. This situation can be produced by a stress which alternates between two values from particle to particle, so that all even particles have one value of the stress, while all odd particles have a different value. SPH will thus not equilibrate a stress field which fluctuates from particle to particle, since no particle motion will be produced to alter the strain field.

The above analysis can be extended to show that the odd-even problem is independent of the form of the kernel function and the size of the smoothing length compared to the particle spacing. Although the derivation strictly applies only to the special case in which all particles are evenly spaced, it is clear from the above figures that it is the odd-even problem which allows noise to remain frozen into SPH solutions. Rapidly varying (particle-to-particle) fields result in an oscillatory function being generated by the kernel approximation which has its maxima and minima at the particle positions, producing negligible gradients at the particles even if they are not equally spaced. The fundamental cause of the problem is that all information is located at the particle positions.

4.5 Alternating Velocities

In the VNR finite-difference method, an alternating stress causes node accelerations which reduce the stress fluctuations. Nodes adjacent to a mesh with a high stress move apart, reducing the density and thus the stress in the mesh. Nodes adjacent to a mesh with a low stress move together, increasing the density and thus the stress in the mesh. Thus, an alternating stress causes alternate nodes to move in opposite directions. In contrast, the previous test problem shows an example of an SPH calculation in which an alternating stress field produces no accelerations.

As a further check on the SPH method, a test was performed in which alternating particles had velocities in opposite directions. A one-dimensional calculation was done in which the particles were initially uniformly spaced with equal masses and densities (2.7 g/cc), just as in the contact impact problem. However, the initial velocities of the particles alternated from particle to particle between equal magnitude positive and negative values. At a later time, particles are thus closer to the neighbor on one side and further from the neighbor on the other side, as seen in Figure 4.10. The discrete points in the figure show the density calculated from the velocity divergence, Eq. (2.32), for each particle as a function of its position. The line is the continuous density field produced by the kernel approximation, Eq. (2.30). The figure shows that the kernel approximation produces an oscillating density field with high densities between particles that are close together and low densities between those that are farther apart. However, equal values of the density are obtained at every particle location. The average value of the continuous density field is equal to the initial density, but the value of the density at each particle location is larger than the initial density. Since the densities at each particle are the same, the stresses are the same at each particle, as shown in Figure 4.11. However, in correspondence with the density, the continuous stress field has high stresses between particles that are close together and low stresses between those that are farther apart. In effect, the uniform stress at each particle is fit with a curve which oscillates between the points, and the stress gradient at each particle is non-zero, as Figure 4.12 shows. The data points and lines in Figure 4.11 and Figure 4.12 represent data points and kernel fits as described for Figure 4.7 and Figure 4.8.

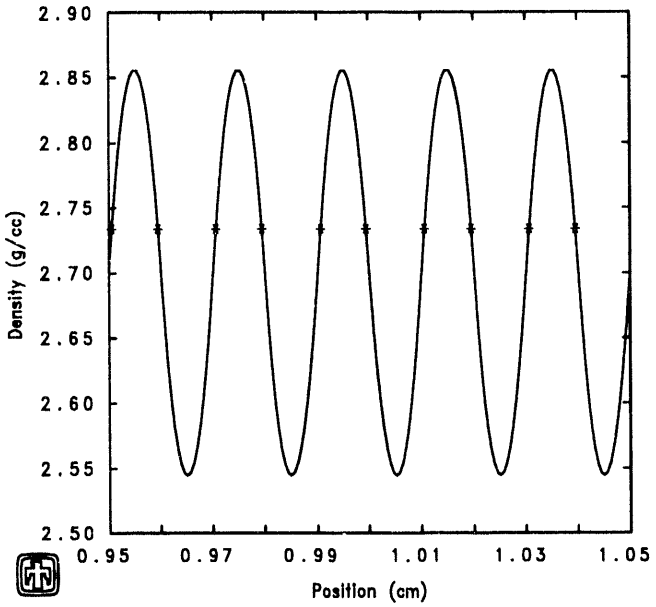


Figure 4.10 Density field from alternating velocities

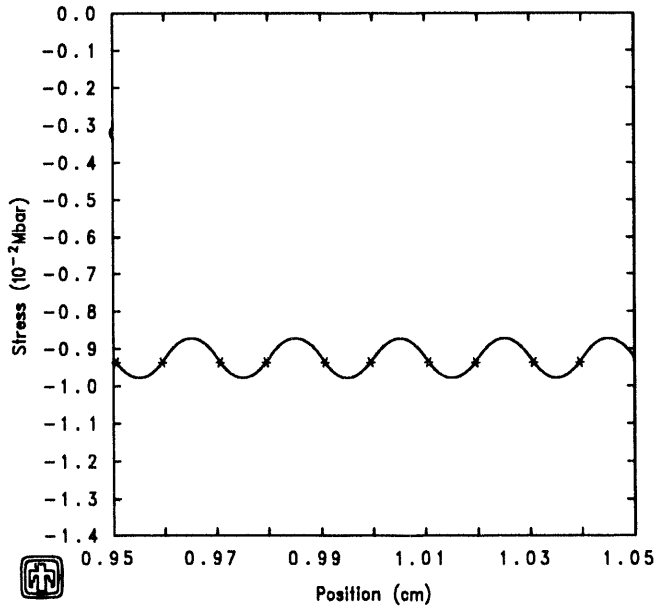


Figure 4.11 Stress field from alternating velocities

The previous test problem shows an example in which the smoothed particle method produces zero accelerations from a stress which fluctuates from particle to particle, while the current test problem shows an example in which non-zero

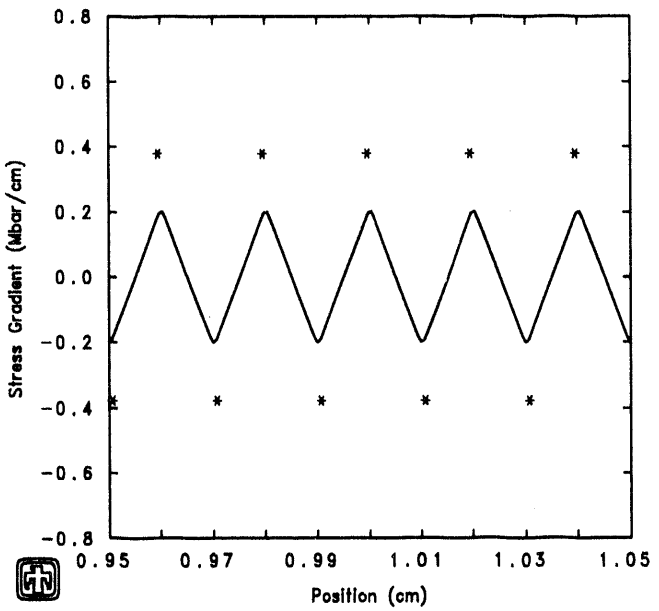


Figure 4.12 Stress gradient from alternating velocities

accelerations result from a stress which is the same at every particle. These results are the opposite of the finite-difference situation. The general form[15] of Eq. (4.5),

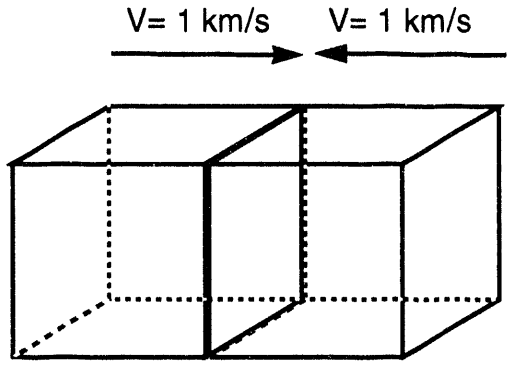
$$\ddot{x}^J = 2\left(\frac{\sigma^{J+1/2} - \sigma^{J-1/2}}{\rho^{J+1/2}(x^{J+1} - x^J) + \rho^{J-1/2}(x^J - x^{J-1})}\right), \tag{4.7}$$

shows that in the finite-difference method, if neighboring stresses are different, an acceleration will be produced, while if they are the same, no acceleration will be produced.

While there is no reason to insist that smoothed particle methods reproduce finite-difference behavior, in the above cases the smoothed particle method seems to be producing results that contradict expected physical behavior. The figures show that the kernel approximation is really just a curve fit to discrete data. As is well known, when the order of a curve fit is too high, a straight line can be fit with a sinusoidally varying curve, as is the case in Figure 4.10 and Figure 4.11. In the above tests the behavior of the SPH algorithm seems to be controlled more by the vagaries of the mathematics of curve fitting than by the physical laws of conservation of mass, momentum, and energy. Such behavior manifests itself not only in simple test cases, but also in the extremely noisy solution to the separated impact problem and, as will be seen, in more complex calculations as well.

4.6 High Velocity 1D Impact

The previous impact test calculations were low velocity tests in the linear regime with less than one percent strain. Figure 4.1 shows a 2 km/s copper on copper



Material Properties

Density, ρ_0	8.94 g/cm^3
C_0	3.447 km/s
S	1.489
Gruneisen Parameter, Γ	1.994
Yield Strength, Y	0

Figure 4.13 High velocity impact

contact impact in which the analytic solution for the stress is 441.28 kbar and the strain is 20 percent. Figure 4.14 shows the results of this calculation using a constant smoothing length; the dots are the SPH results and the solid line is the TOODY[10] calculation. Here again, either the finite-difference or the SPH form of the viscosity is adequate to smooth the shock in the SPH calculation. Since an energy dependent equation of state is being used, some noise is generated in the stress profile by either form of viscosity at the impact interface. An interface effect would also be present in a VNR finite-difference calculation and is caused by excess energy being deposited by the artificial viscosity in the region near the interface where the sharp impact shock has not propagated far enough to become a smooth steady wave. The excess energy would result in a hot, low density region, but the stress would be uniform. As discussed previously, in the SPH algorithm the stress fluctuations at the interface are left fixed in place rather than being equilibrated. The interesting point to note in this calculation is that while the TOODY calculation reproduces the analytic pressure of 441.28 kbar, the inset in

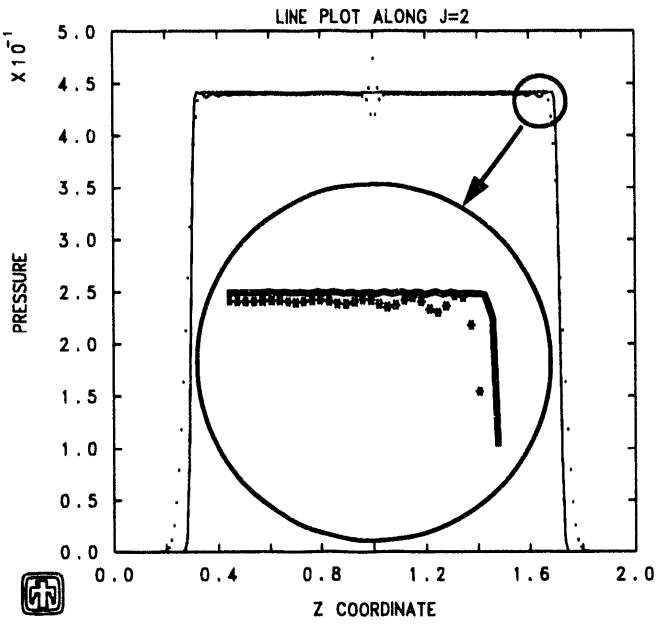


Figure 4.14 High velocity impact with constant smoothing length

the figure shows that the SPH and TOODY results do not agree, and in fact the constant smoothing length SPH solution is about 0.5% low at 439 kbar. Figure 4.15

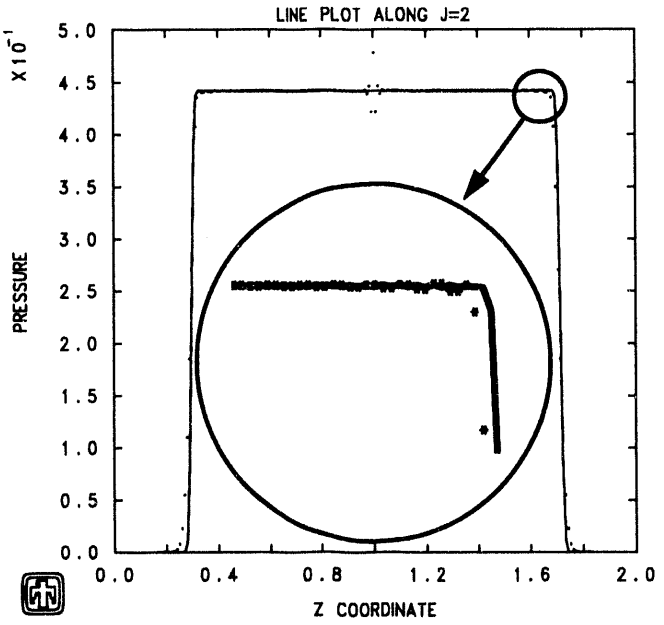


Figure 4.15 High velocity impact with variable smoothing length

shows the result with variable smoothing, in which the analytic solution is obtained with SPH, as indicated by the inset which shows that SPH and TOODY agree.

Thus, when strains are large constant smoothing is inadequate and variable smoothing is required to obtain the exact solution.

When this high velocity impact is repeated with a gap at the impact interface rather than with the materials in contact, the result shown in Figure 4.16 is obtained,

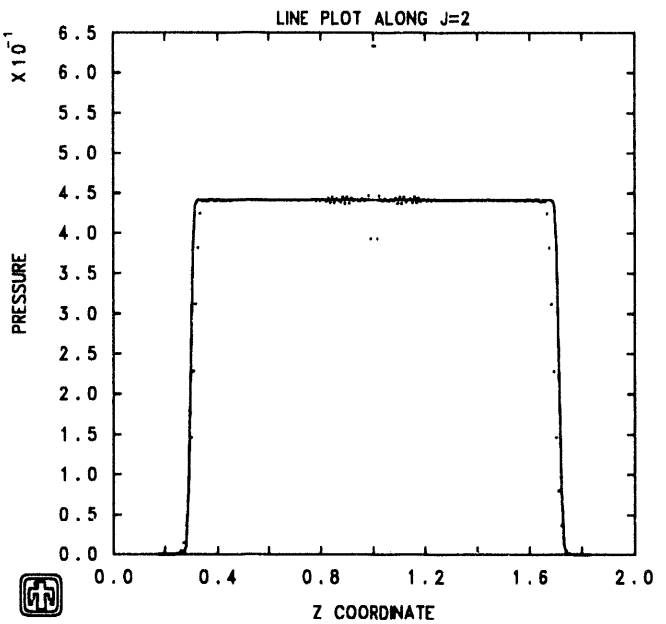
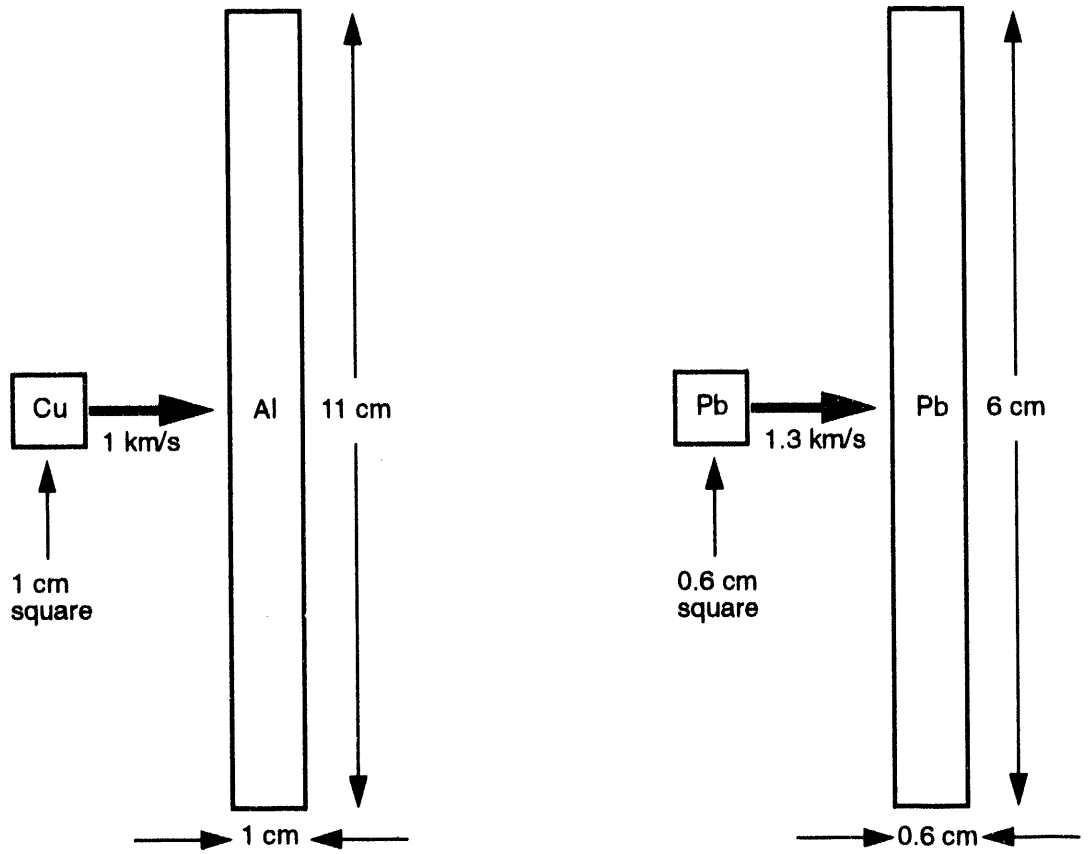


Figure 4.16 High velocity impact with separated interface

where again the dots are the SPH result and the line is the finite-difference result. There is much less difference between the contact and separated impacts at the higher velocity than at the lower velocity, as a comparison of the low velocity contact and separated impacts, Figure 4.2 and Figure 4.5, with the high velocity contact and separated impacts, Figure 4.15 and Figure 4.16, clearly shows. Although excess noise is generated and left fixed in place at the interface at the high velocity, the amplitude of the fluctuations is smaller and the solution is more acceptable. At the low velocity the gap takes many time steps to close, while at the higher velocity the gap is closed in a few time steps and the effect is smaller. Clearly, if relative impact velocities are large enough that boundary particles move from a separation of greater than $2h$ to a separation equal to the in-material particle spacing within a single time step, gaps will have no effect.

4.7 Plate Penetration

Figure 4.17 shows the configuration for two plate penetration calculations which examine the behavior of SPH in the multi-dimensional high deformation regime. The configurations are similar, with a square projectile impacting a plate whose thickness is equal to the projectile dimension, and whose length is an order of magnitude larger. In one case, a heavy material, copper, impacts a light plate,



Material Properties

	Al	Cu	Pb
Density, ρ_0 (g/cm^3)	2.7	8.94	11.35
C_0 (km/s)	5.38	3.447	2.05
S	1.337	1.489	1.46
Gruneisen Parameter, Γ	2.	1.994	2.8
Poisson's Ratio, ν	0.333	.345	.44
Yield Strength, Y (kbar)	3	2	1
Fracture Strength	∞	∞	∞

Figure 4.17 Plate penetration configurations

aluminum, and in the other the impact is symmetric, with both materials being lead. The impact velocities are 1 km/s and 1.3 km/s, respectively. Rectangular geometry is assumed in both cases. All materials are represented as elastic-perfectly plastic with finite yield strengths, but with no fracture model included, so that the fracture strength is infinite. Consequently, the deviatoric stresses are limited by the yield strength, but there is no limit to the magnitude of mean tensile stress (negative

pressure) which the material can generate. Figure 4.18 shows a sequence of snapshots of particle positions at 0, 20, 40, and 60 microseconds for the lead on lead impact, and seems to indicate that SPH is perfectly capable of treating this impact. However, a closer look at the solution and an examination of the effect of variations in some of the calculational parameters raises further questions.

Comparisons of the effect of resolution and type of artificial viscosity were made for both configurations. All calculations had uniform zoning with particles placed on a rectangular lattice, and all used a density-based variable smoothing length with the initial smoothing length, h, equal to twice the particle spacing. Since the kernel function goes to zero at $2h$, interactions occurred between all particles within a radius of four spacings. The low resolution calculations had 10 particles through the plate thickness for the copper on aluminum impact and 12 for the lead on lead, while the high resolution calculations had 20 and 24 particles through the plate thickness, respectively. Calculations using the finite-difference form of the viscosity used $B_1 = 2$, $B_2 = 0.1$, while those using the SPH form of the viscosity used the values $\alpha = \beta = 2.5$. Figure 4.19 shows the results 100 microseconds after impact for the copper on aluminum case, while Figure 4.20 shows results at the same time for the lead on lead impact. In these figures, calculations in the left column use the finite-difference viscosity (denoted in the figure by WV), those in the right column use the SPH viscosity (denoted by MV), the top row is coarse resolution (denoted COARSE), and the bottom row is high resolution (denoted FINE). Thus, all combinations of the two viscosities and resolutions are shown.

In contrast to the one-dimensional case, the form of the viscosity has a major effect on the solution in these two-dimensional calculations. A common feature in all cases is that in the lead on lead case the impact stress is spread over a large area of the target, while the copper projectile remains relatively intact, confining the damage to a small region of the aluminum plate. However, the finite-difference viscosity calculations show the projectiles penetrating the plates, while in the SPH viscosity calculations the plates bend and deform, almost being dragged into the crater by the projectile, but do not separate. There is also a much greater effect of resolution when the SPH viscosity is being used than with the finite-difference viscosity. Unfortunately, the correct solution to this problem is unknown. However, since the equation of state limits the deviatoric stress but not the tensile stress, it seems plausible that the correct mathematical solution to the problem would be the formation of a bubble coming out the back of the plate that would stretch and thin, but not break. This process could only be expected to continue in the calculations until the bubble thickness becomes equal to the particle size, at which time particles would lose contact as illustrated in Figure 2.4. The bubble thickness is approximately equal everywhere in the lead on lead case, while it is much thicker directly behind the copper target in the other case, with the contacts between this plug and the rest of the plate thinning and breaking much earlier. However, the particles seem more inclined to separate and clump than spread evenly in the

finite-difference viscosity case, while the SPH viscosity seems to add an inordinate amount of artificial shear strength to the plate.

An examination of the stress fields, as in Figure 4.21, which is at a time of 15 microseconds in the high resolution finite-difference viscosity lead on lead calculation, reveals that the locked-in noise seen in 1D is also present in higher dimensions. The color of each particle in this figure is representative of the magnitude of its pressure in megabars, as indicated by the scale on the right. There is a general background of plus and minus stress with a magnitude of roughly 20 kbar, but also a few isolated particles with stress magnitudes of 70 to 80 kbar. These plus and minus pressure spikes occur on particles that are usually grouped together, and were initially on the impact interface. The reason given previously that these pressure spikes do not equilibrate is verified in Figure 4.22 , which is for the high resolution finite-difference viscosity calculation of the copper on aluminum impact at 25 microseconds. Here the colors in the top half of the figure show the actual stresses at each particle. The bottom half is a reflection of the top half, except that here the color represents the kernel approximation to the stress field as calculated from Eq. (2.14). As described previously, the kernel fit to the data is insensitive to particle-to-particle fluctuations, smoothing the stress spikes into a much more uniform stress field at the location of each particle. Since the particle accelerations are based on this smoothed stress field, the spikes remain fixed and do not equilibrate.

The reason for the differences between the finite-difference and SPH forms of the viscosity is illustrated in Figure 4.23, which is for the high resolution SPH viscosity calculation of the lead on lead impact at 15 microseconds. There are locked-in stress spikes present with this form of the viscosity, but the distribution of particles is much more ordered than in Figure 4.21. The particles have in fact formed filaments resembling long polymer chains, and the stress spikes appear in ordered rows. This polymerization of the particles indicates that the SPH form of the viscosity inhibits relative particle motion and generates an artificial shear strength.

Possibly the most important point to note in the plate penetration calculations is that they exhibit breakup and separation of the bodies even though fracture is prohibited. The calculations indicate that particle flow tends to be uneven in tensile regions, which causes clumping, separation, and eventual artificial numerical fracture due to the fact that communication is lost between particles as their separation exceeds the smoothing length. Particles can lose contact even if a variable smoothing length based on density is used, since the density does not change appreciably in the clumping and separation process. Unfortunately, the evidence that fracture is numerical rather than physical is not completely unambiguous, since physical intuition would seem to say that real materials should break up under such conditions. The calculations thus seem to be behaving correctly, regardless of the fact that fracture has been prohibited, since very little intuition exists about materials with infinite fracture strengths. In addition, since there is a finite yield strength, it could be argued that breakup should occur due to

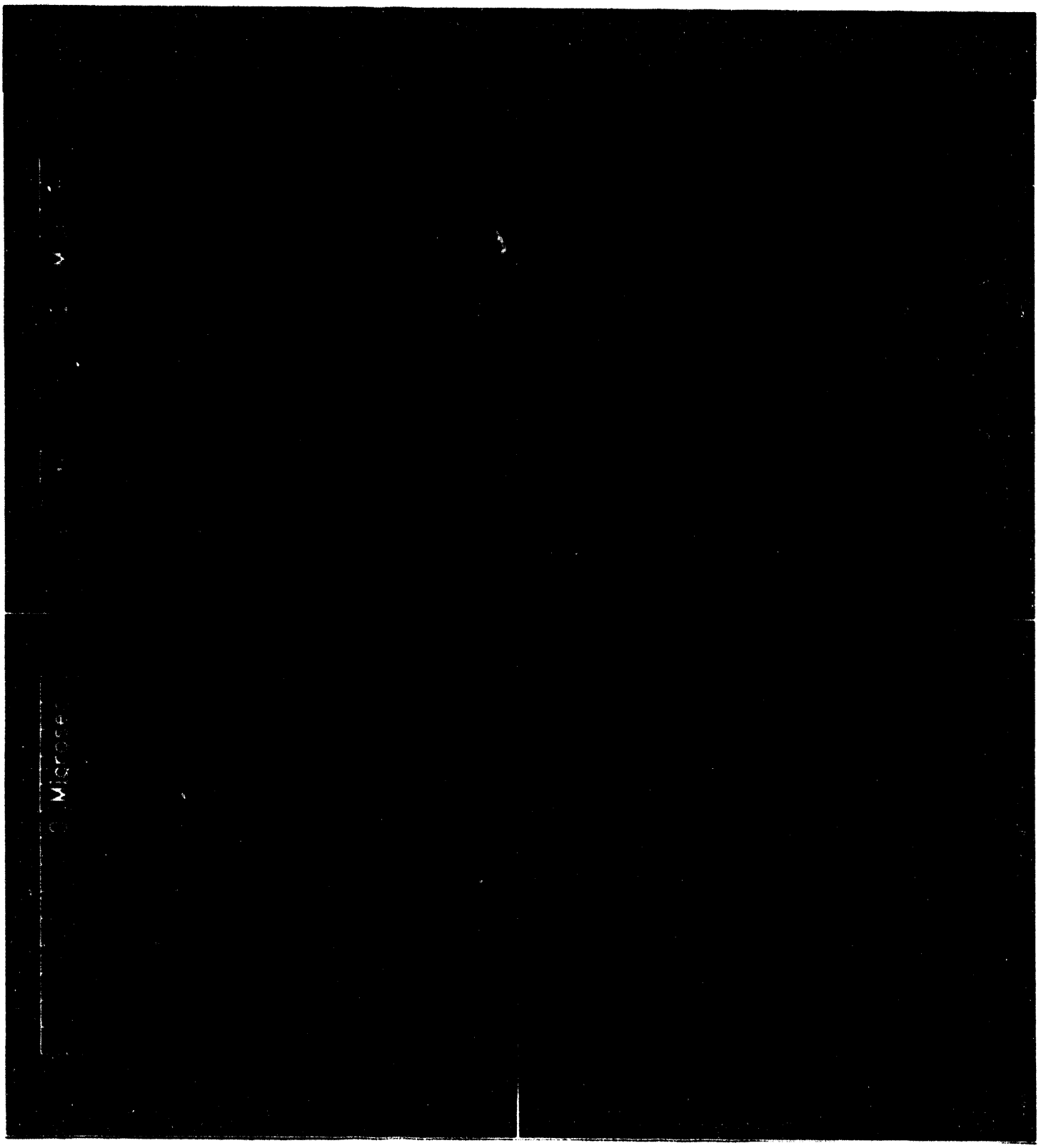


Figure 4.18 Lead on lead impact

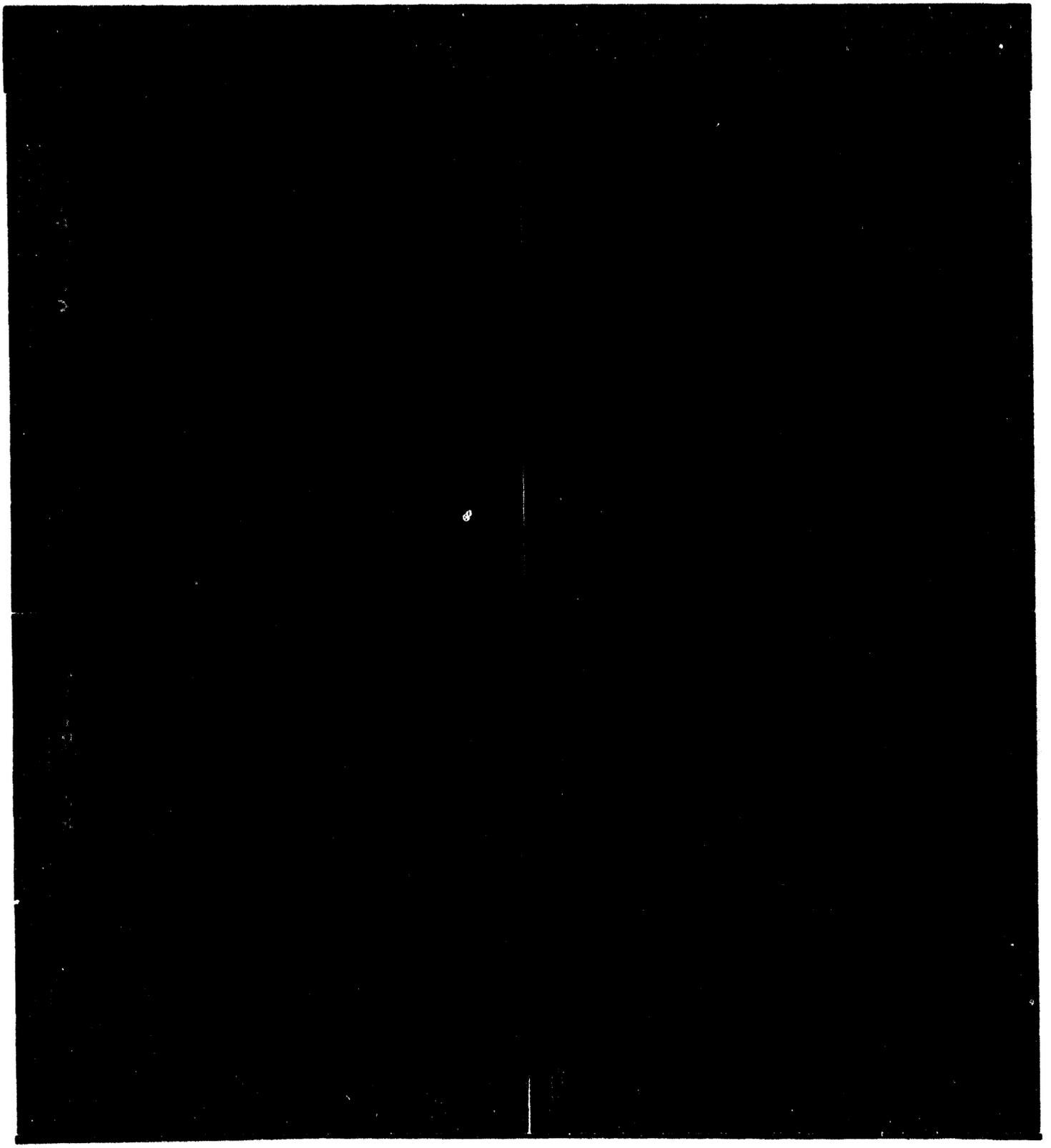


Figure 4.19 Viscosity and resolution comparison for copper on aluminum

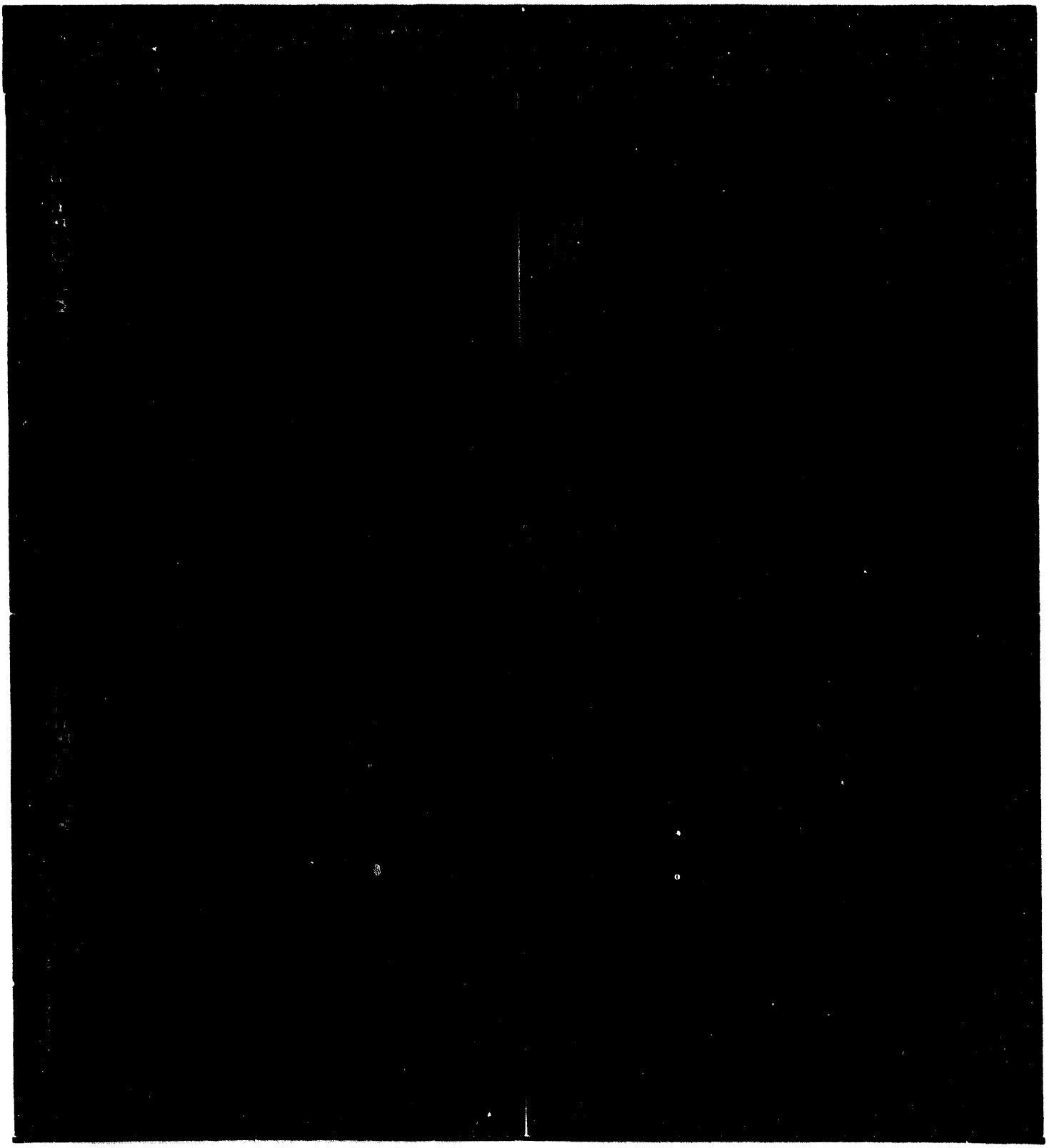


Figure 4.20 Viscosity and resolution comparison for lead on lead

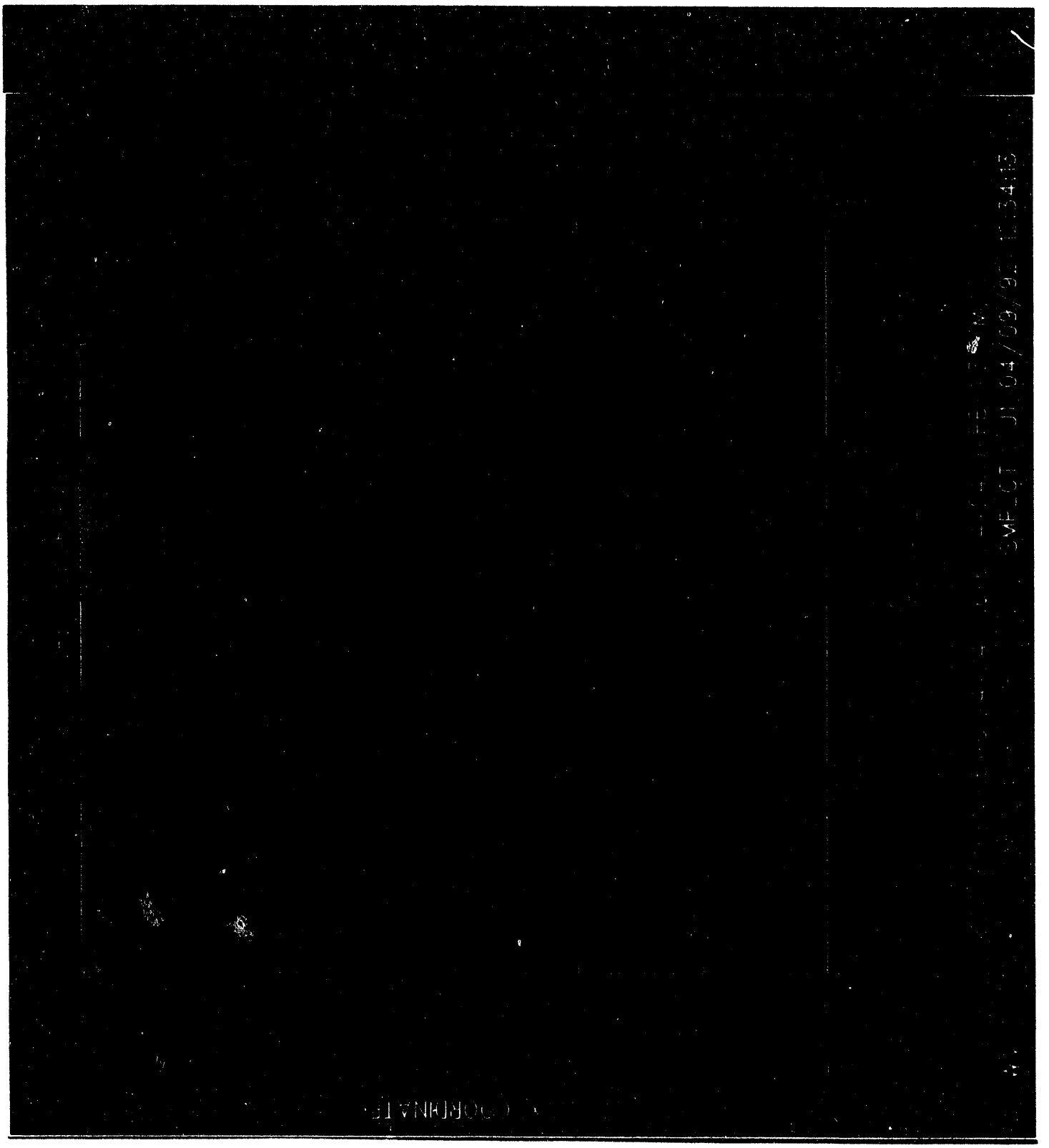


Figure 4.21 Pressure at 15 microseconds for lead on lead

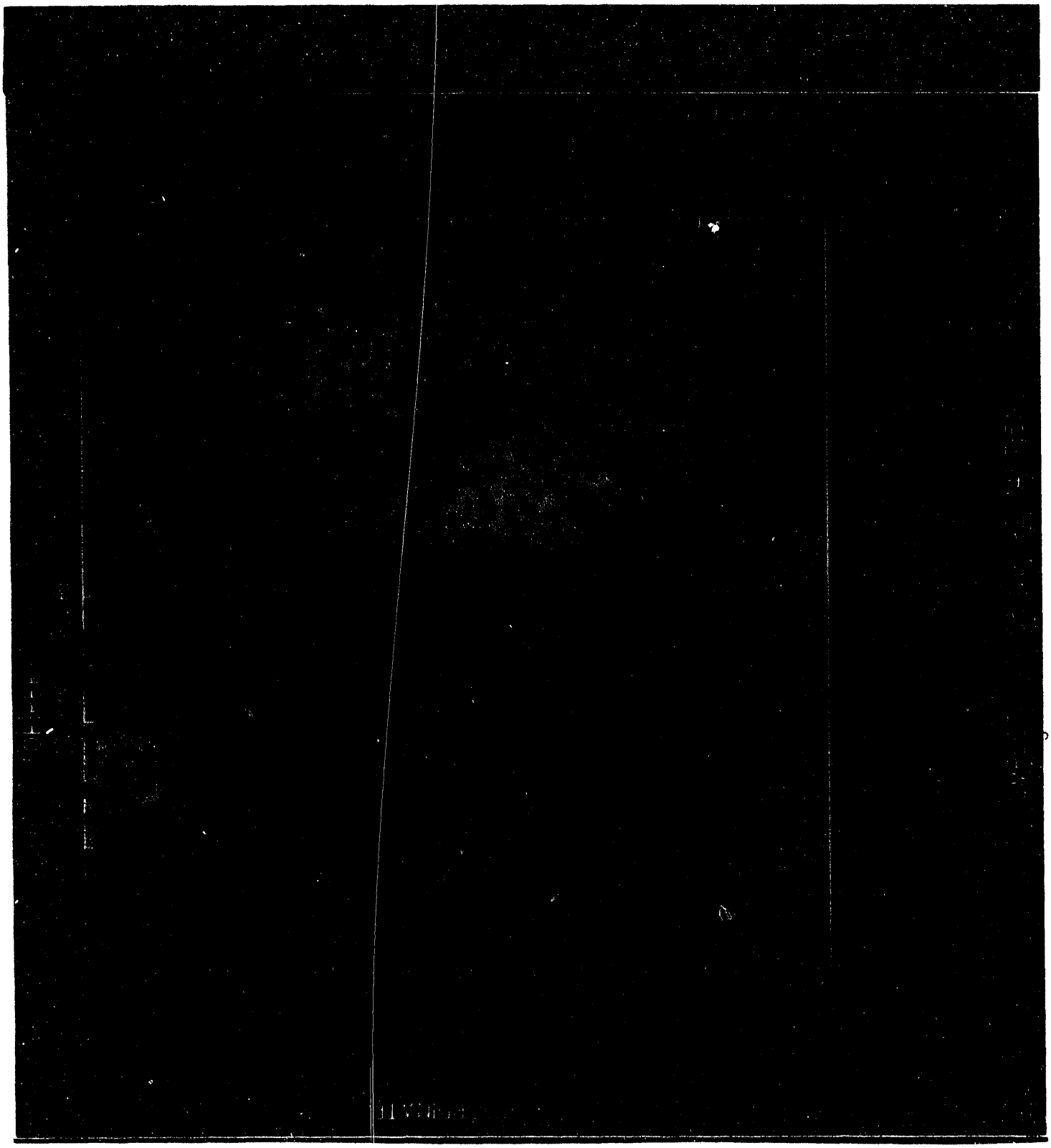


Figure 4.22 Pressure at 25 microseconds for copper on aluminum

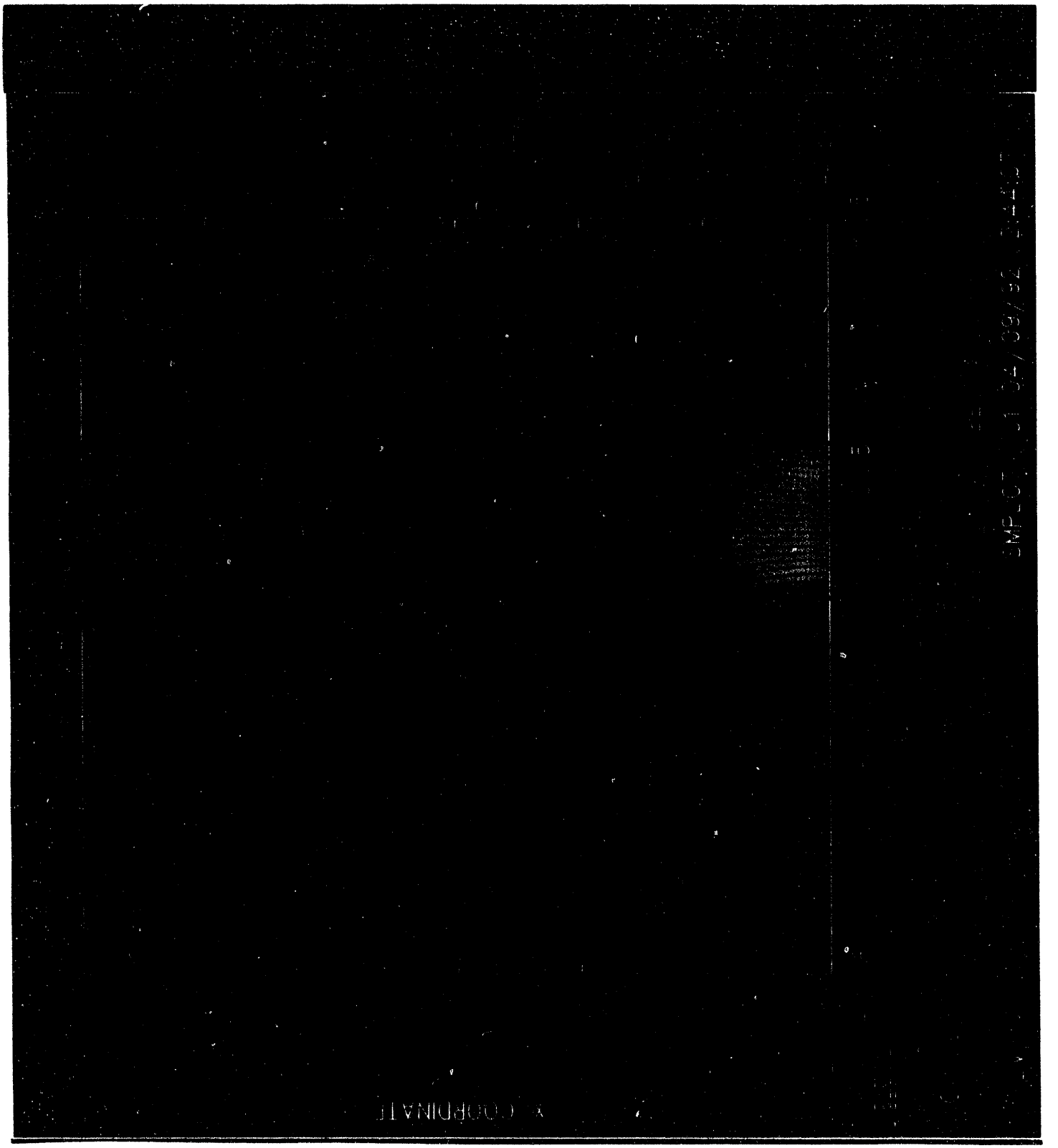
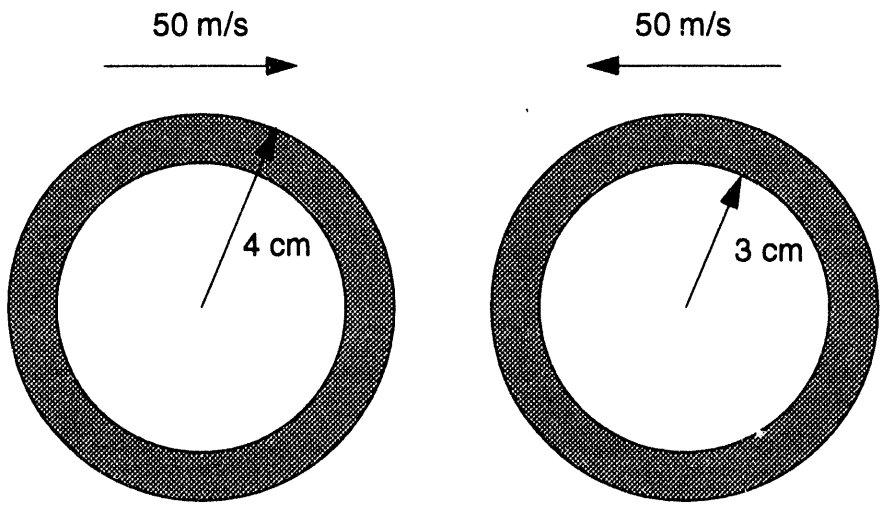


Figure 4.23 Pressure at 15 microseconds for lead on lead with SPH viscosity

shear fracture. Therefore, the next test concerns a situation where physical intuition indicates that breakup should not occur in order to unambiguously evaluate the behavior of the algorithm.

4.8 Tennis Ball Impact

The plate penetration test illustrated some important aspects of the SPH algorithm, but did not clearly indicate whether or not it was behaving correctly in tension. The current test, which is the impact of a spherical rubber shell on a rigid wall at velocities achievable in a game of tennis, is designed to resolve the question. Tennis balls do not normally shatter on impact, and standard finite-difference wavecodes easily produce elastic impacts and rebounds, even though the deformation of the ball is quite large.



Material Properties

Density	1.01 g/cm^3
Hugoniot	$U_s=C_0+S\,U_p$
C_0	.852 km/s
S	1.865
Gruneisen Parameter	1.5
Poisson's Ratio	.4
Yield Strength	∞
Fracture Strength	∞

Figure 4.24 Description of tennis ball impact problem

The tennis ball impact configuration is illustrated in Figure 4.24. A ball with material properties characteristic of rubber impacts a rigid wall (same as the symmetric

impact of two balls) at 50 m/s. Again, rectangular coordinates are used so that infinitely long tubes, rather than spherical balls, are actually being represented.

Figure 4.25 shows calculational results for this problem using the Lagrangian finite-difference wavecode TOODY[10]. The frames from left to right, top to bottom, show the configuration at the instant of impact, then 500, 1000, and 1500 microseconds after impact. The balls impact, distort, and rebound. Flexural oscillations in the shape of the ball persist after the impact, but the balls remain intact and do not fracture or fragment, in accord with physical intuition. The calculation was also done using the finite-element code PRONTO[9] with identical results.

SPH calculations of the impact used three different initial particle packings: radial as shown in Figure 4.26 , rectangular as shown in Figure 4.27 , and hexagonal as shown in Figure 4.28 . All used 10 particles through the radius of the ball, which is the same number of zones used in the TOODY calculation.

Results of the SPH calculations using the radial grid and the finite-difference viscosity are shown at 100, 400, 700, and 1000 microseconds in Figure 4.29 . In regions of the ball where strains are such that the interparticle distance is increasing and tension is developing, the particles are clumping, separating, and eventually fracturing. This is happening in spite of the use of a density-based variable smoothing length, since density changes are small, even though individual components of the strain tensor are large. The fracture is entirely numerical, since the fracture and yield strengths are much larger than the approximately one-half kilobar maximum stress in the problem. Results for the rectangular and hexagonal packings are similar, although the details of the pattern of breakup vary. Figure 4.30 shows the results using the SPH viscosity in the standard form in which the viscous stress is zero if the particles are separating. Although numerical fracture still occurs, the ball holds together better, in accord with the results of the plate penetration calculations.

In order to further evaluate the effect of the different viscosities in holding the ball together, radially gridded calculations were run in which all EOS stresses were zeroed, and the only forces in the calculations were due to the artificial viscosities. Figure 4.31 shows the results using the finite-difference viscosity. As particles approach and the density increases, a mean stress is developed which accelerates the particles which then move away in relatively independent directions. Figure 4.32 shows the results using the SPH viscosity. In this case, particle interpenetration is completely inhibited, and there is a strong group nature to the particle flow. Clearly, the SPH viscosity acts to inhibit relative particle motion. This is emphasized in Figure 4.33 and Figure 4.34, which show, from left to right and top to bottom, plots of the no EOS calculation, and the regular EOS calculations for the rectangular, radial, and hexagonal packings. The plots show close-ups of the top half of the ball, with each particle colored based on its velocity in the original direction of motion, according to the scale on the right. All times are at 1000

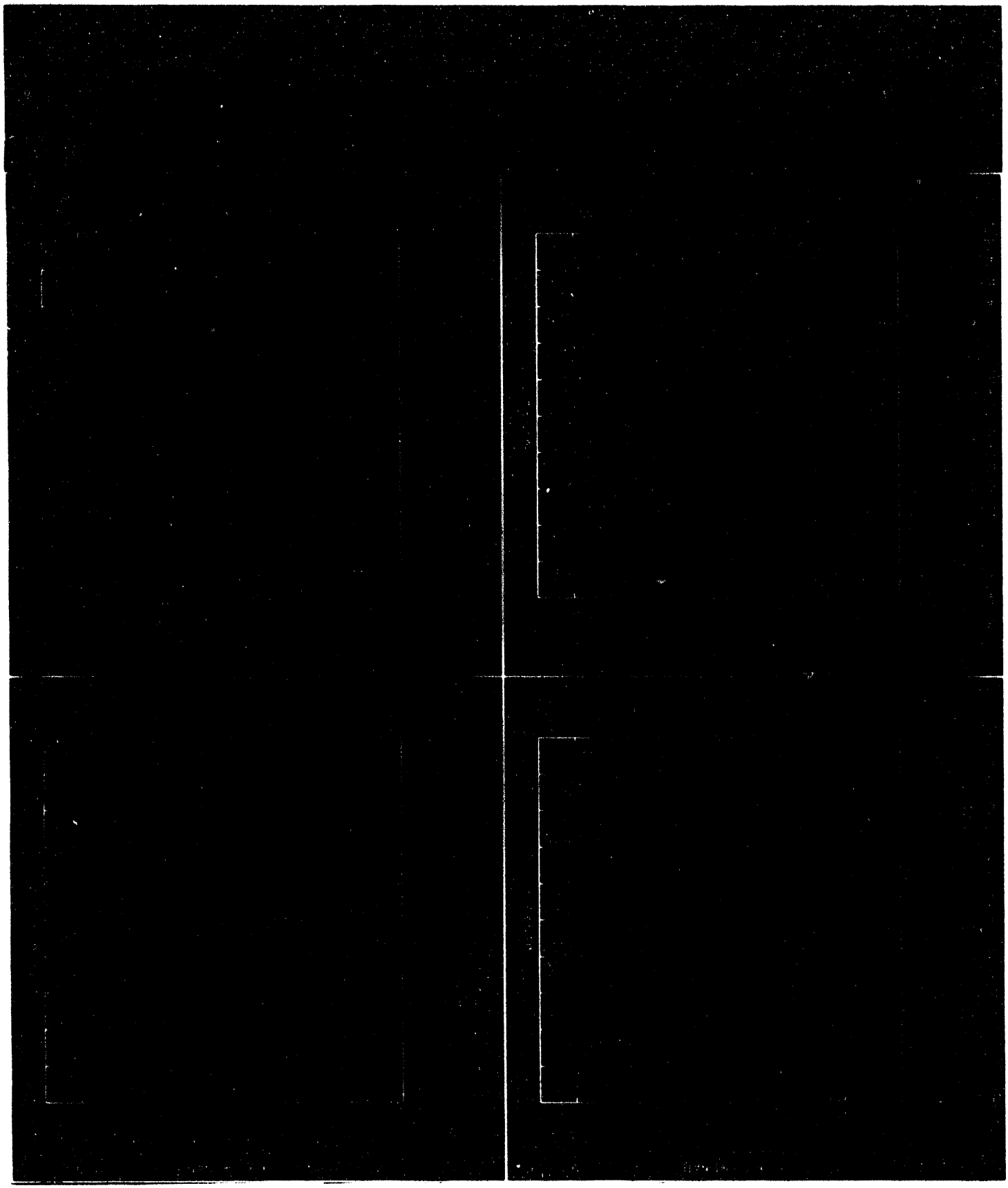


Figure 4.25 TOODY results for tennis ball impact

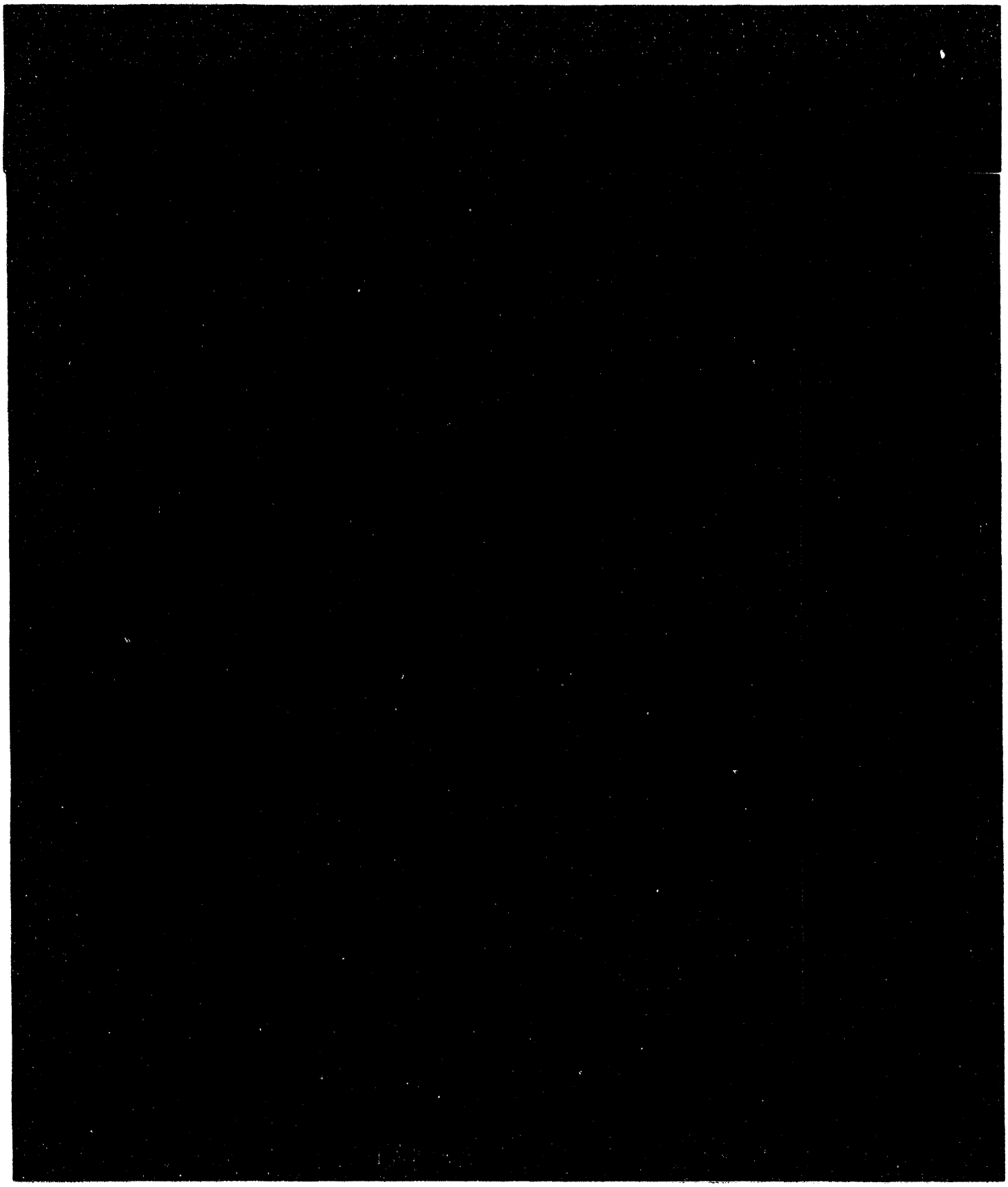


Figure 4.26 Radial SPH packing for tennis ball impact

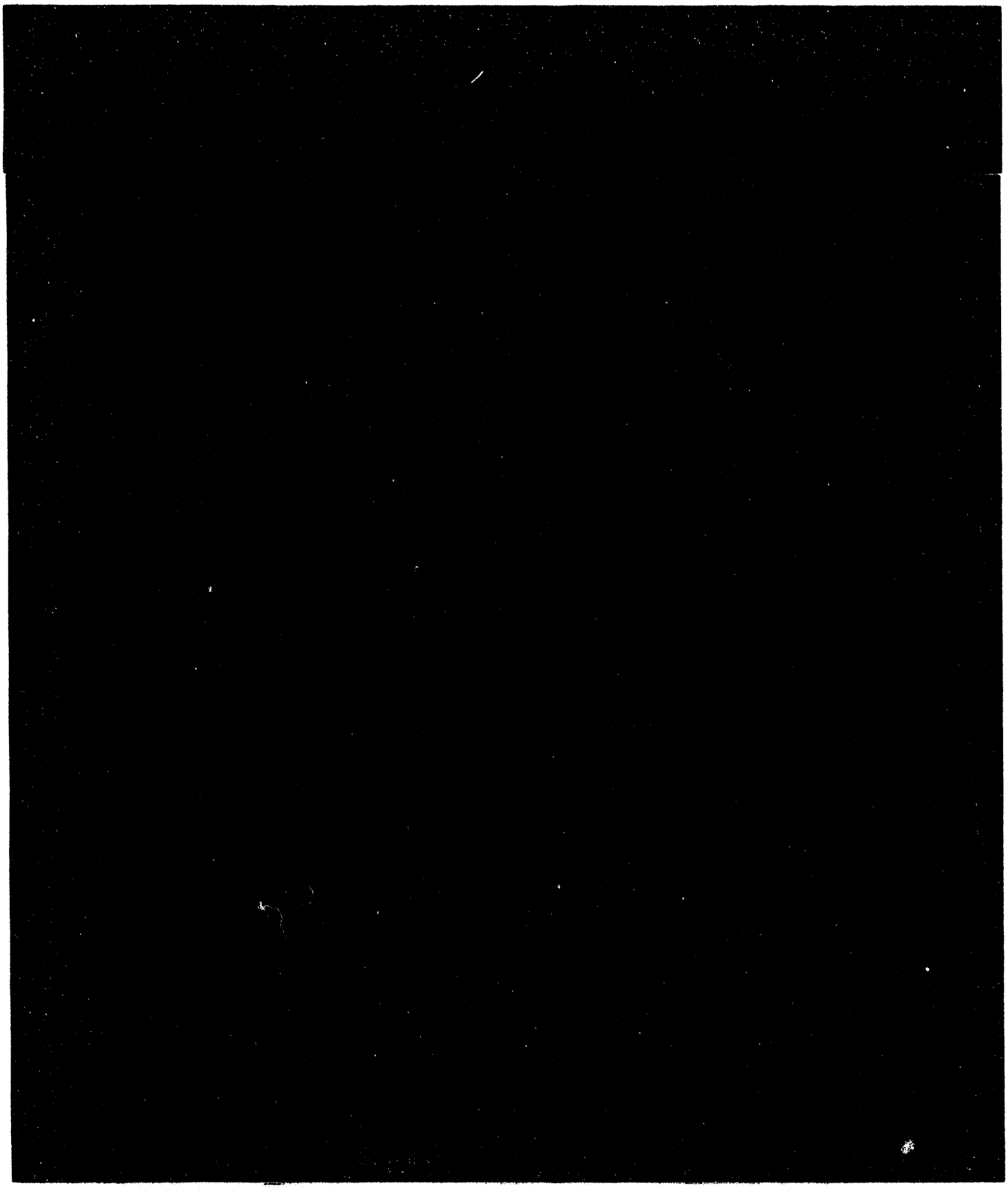


Figure 4.27 Rectangular SPH packing for tennis ball impact

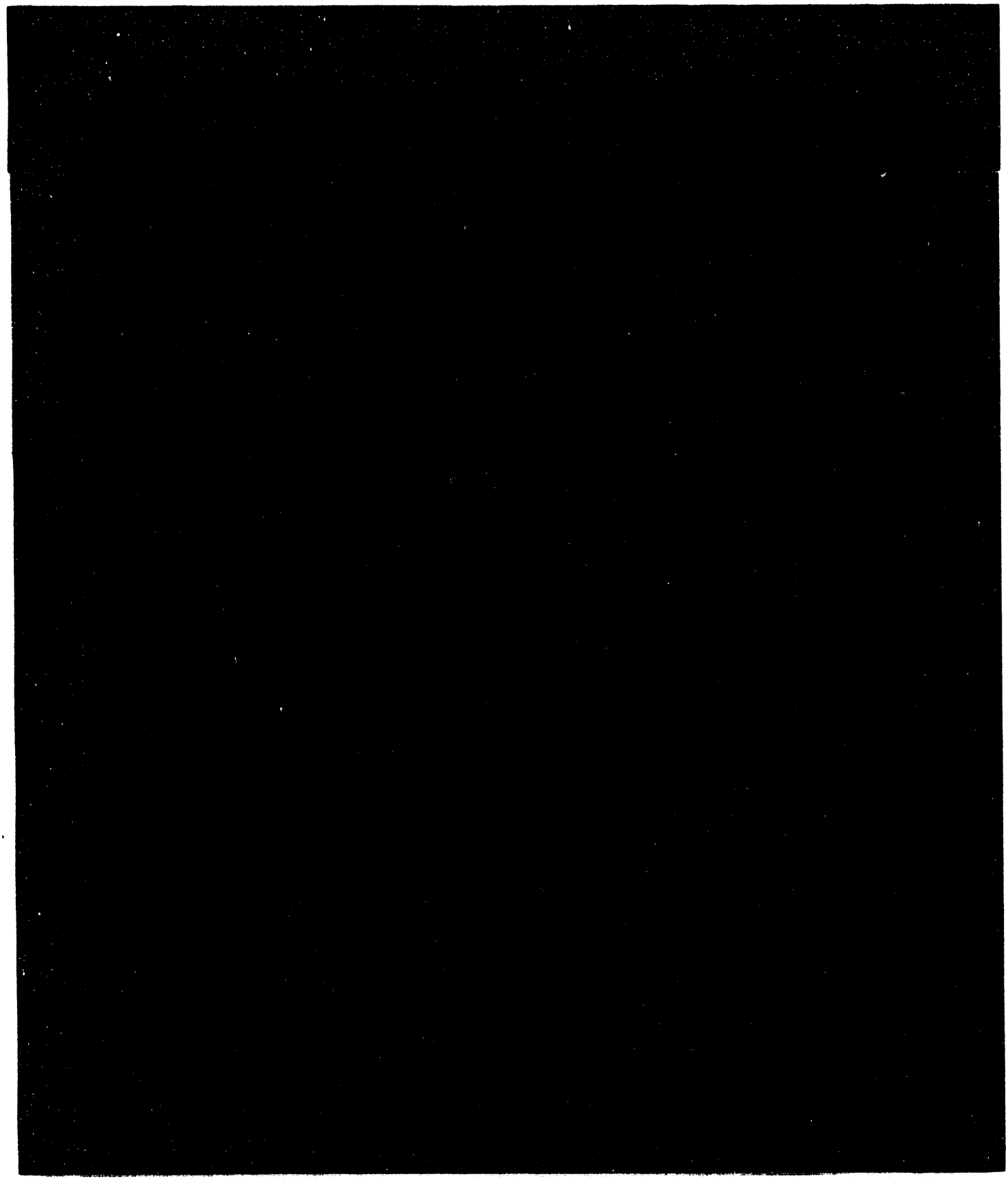


Figure 4.28 Hexagonal SPH packing for tennis ball impact

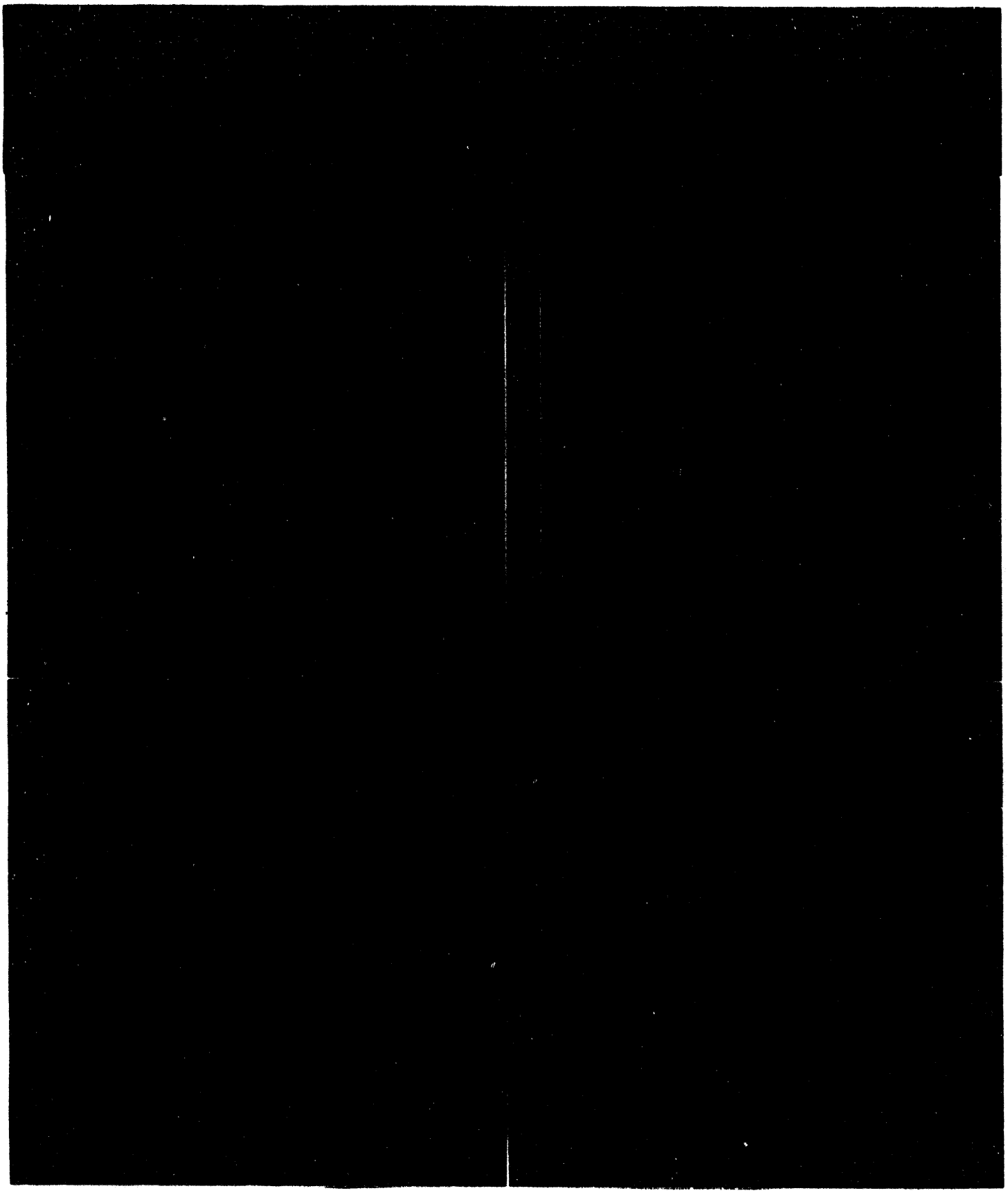


Figure 4.29 SPH results with radial packing and finite-difference viscosity

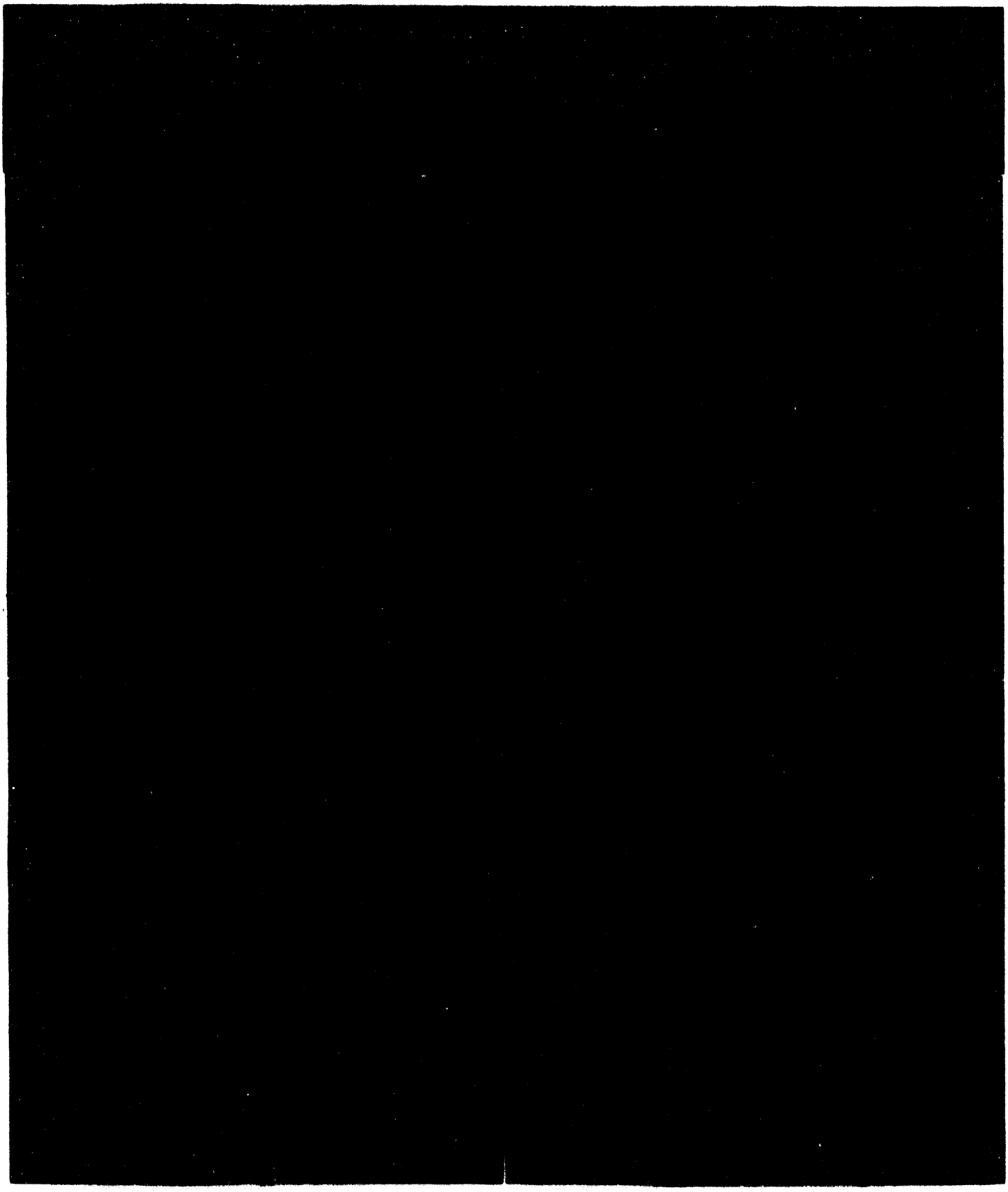


Figure 4.30 SPH results with radial packing and SPH viscosity

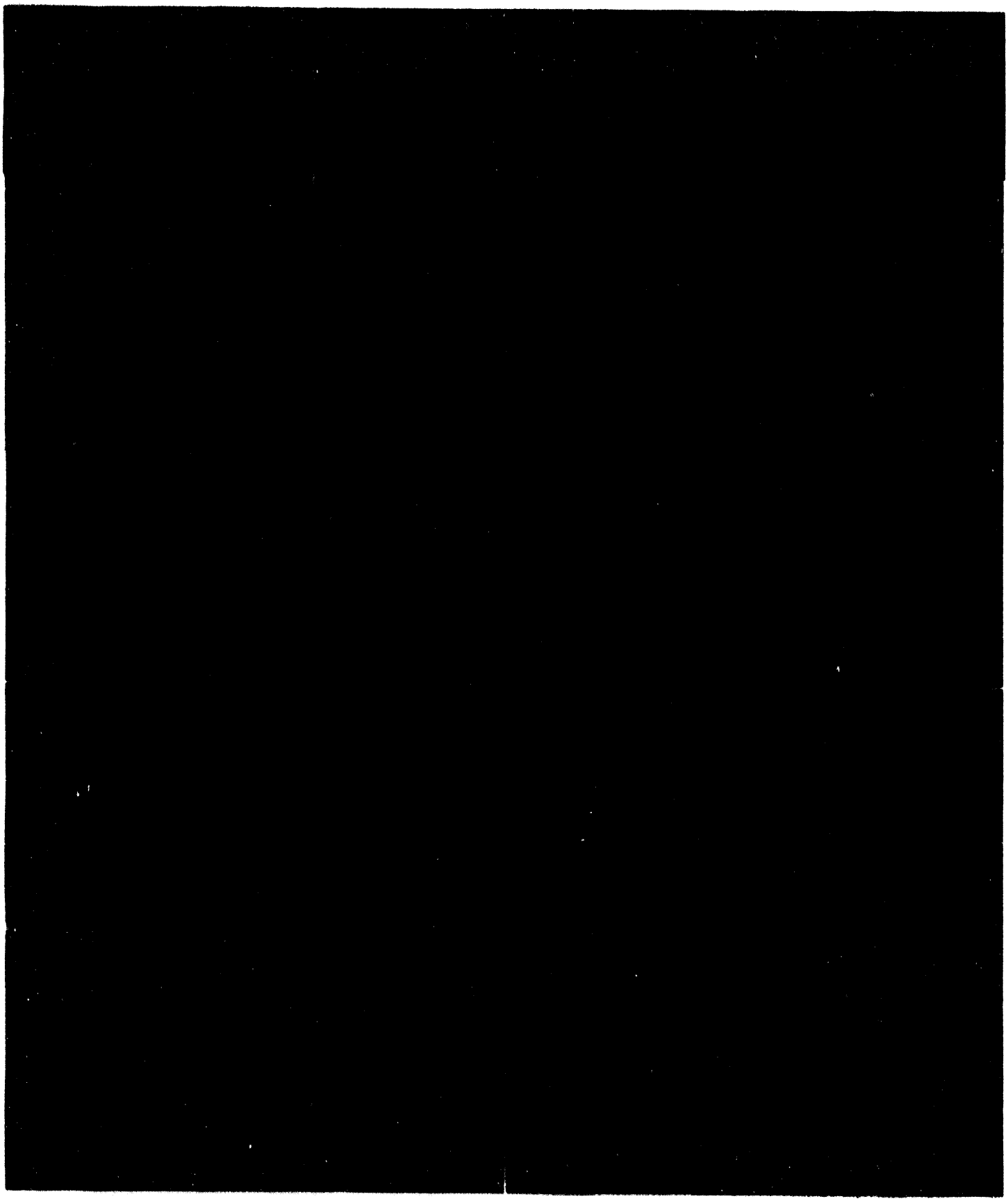


Figure 4.31 Finite-difference viscosity only

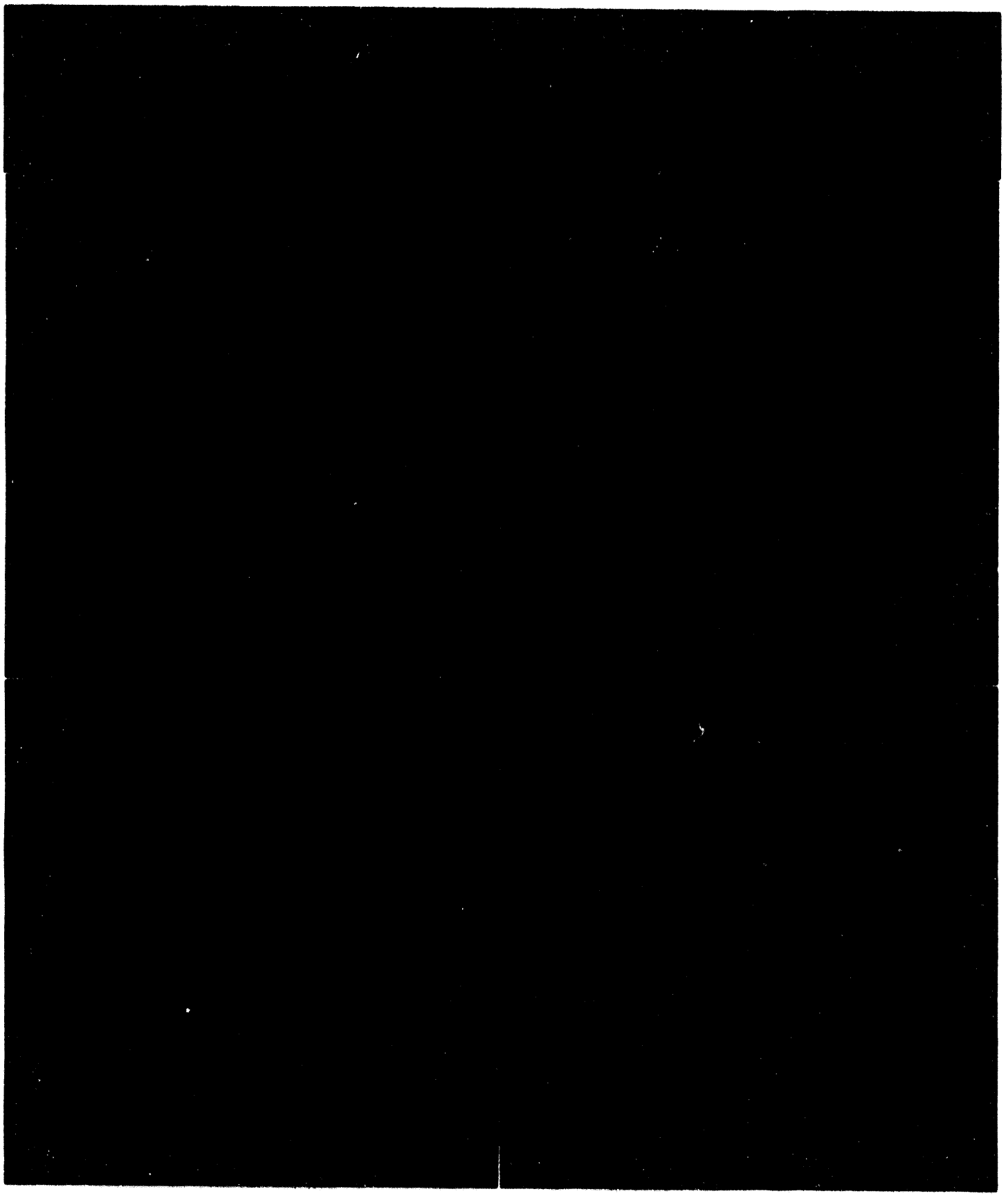


Figure 4.32 SPH viscosity only

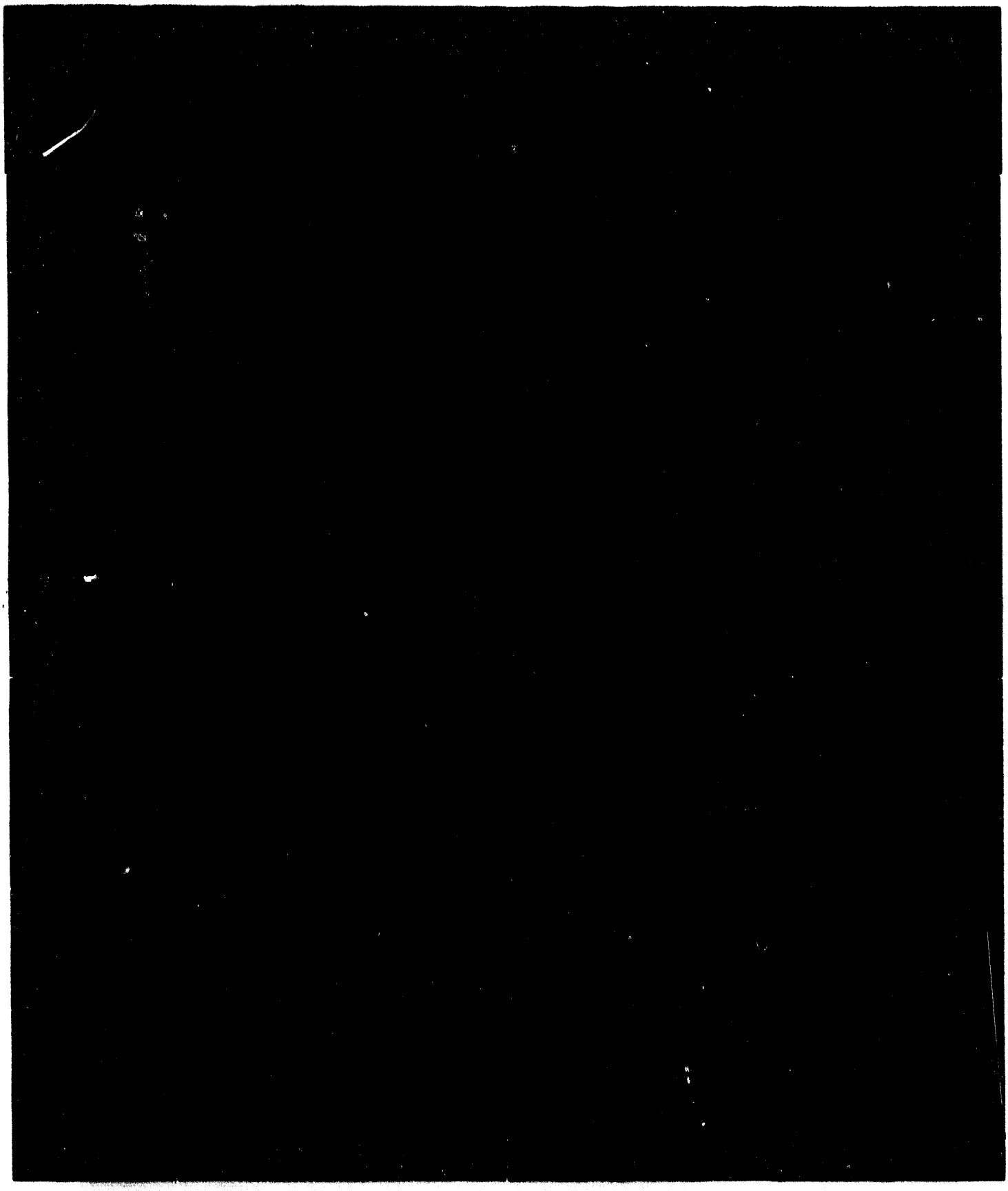


Figure 4.33 Particle velocity with finite-difference viscosity

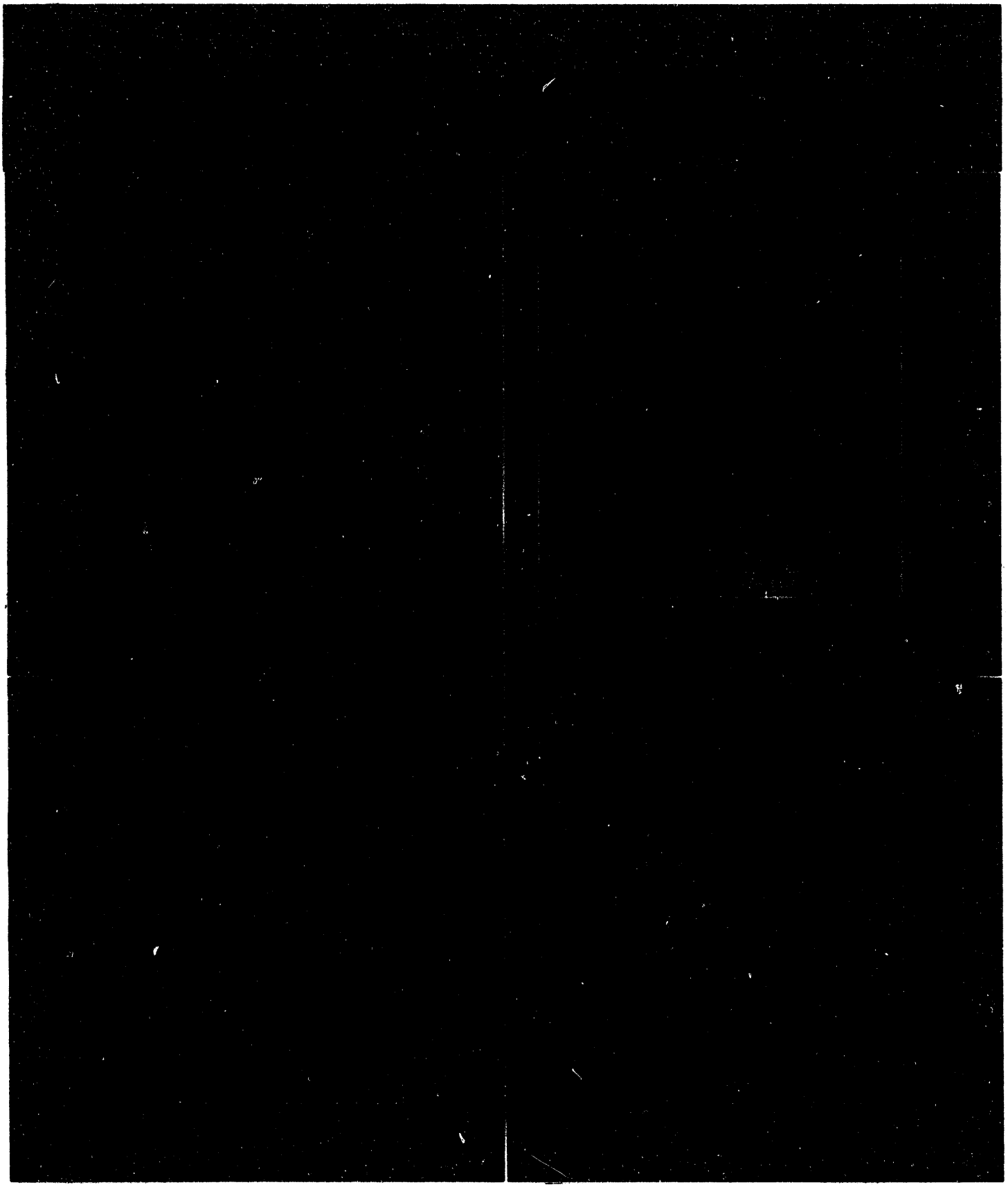


Figure 4.34 Particle velocity with SPH viscosity

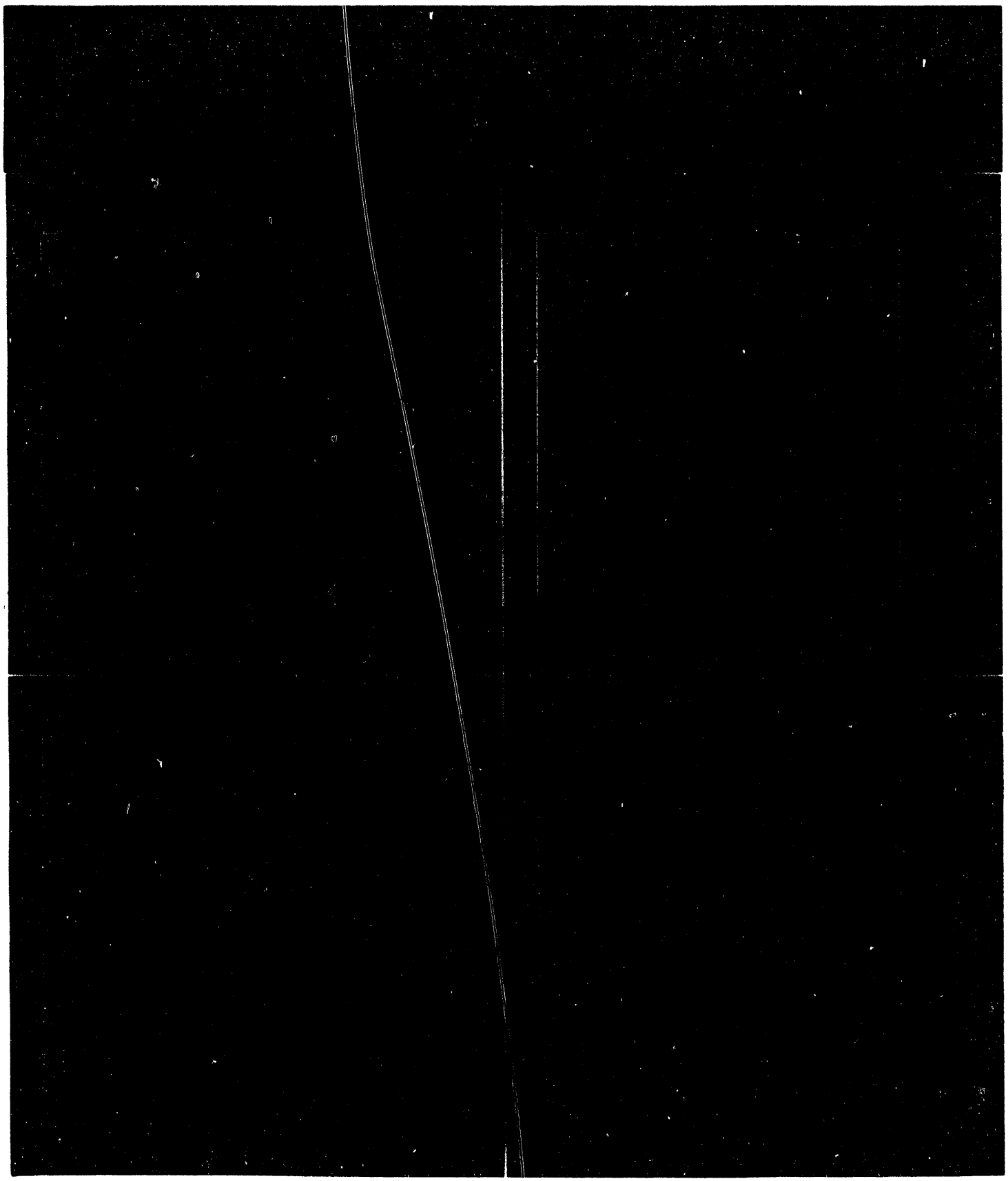


Figure 4.35 Radial packing with SPH viscosity on in expansion

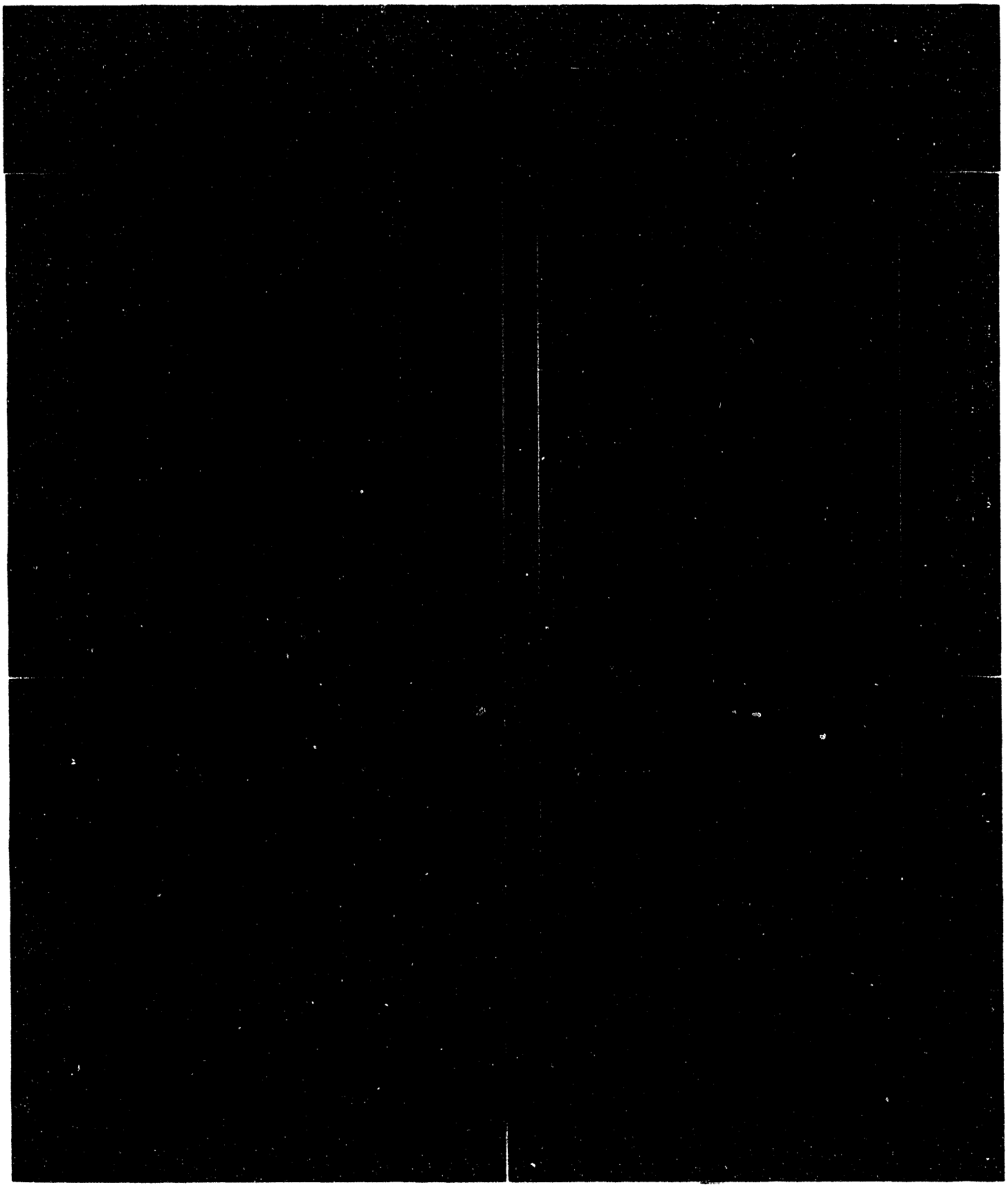


Figure 4.36 Rectangular packing with SPH viscosity on in expansion

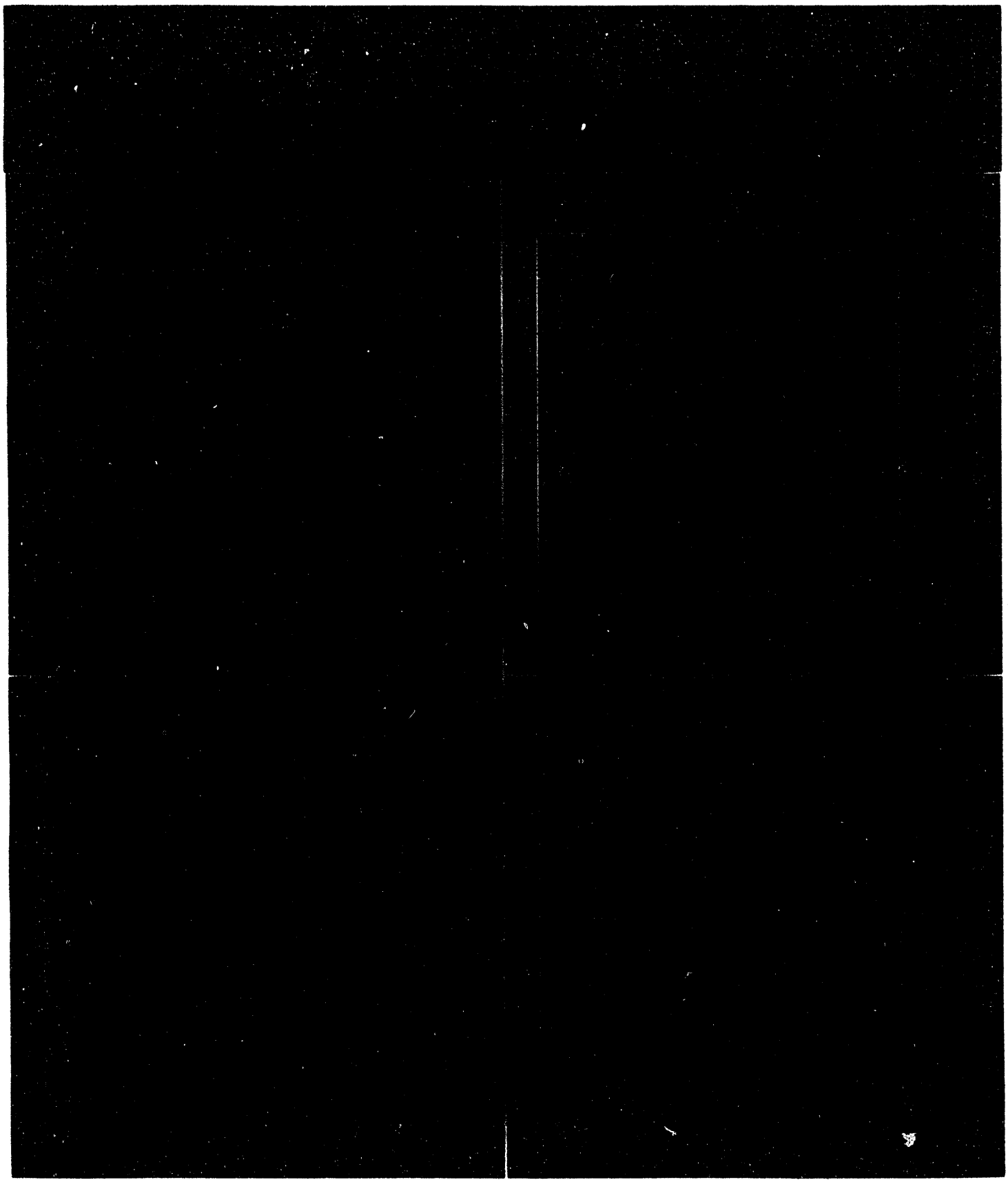


Figure 4.37 Hexagonal packing with SPH viscosity on in expansion

microseconds. Figure 4.33 is for the finite-difference viscosity case, and there is a great deal of particle-to-particle fluctuation. As described previously, such fluctuations in the velocity and stress fields are not equilibrated due to the inability of the SPH gradient operator to detect variations on the scale of the particle spacing. Figure 4.34 shows the SPH viscosity case, and the velocity fields are much smoother, with the inhibiting effect of this form of the viscosity on relative particle motion clearly demonstrated.

Figure 4.35 shows the radially-gridded SPH results if the SPH viscosity is also turned on when particles are separating. Here the plots at time 0, 500, 1000, and 1500 microseconds show that the ball remains basically intact. It is disturbing that the viscosity has such a large effect on the solution, and it seems clear that the SPH viscosity has a different function than simply spreading discontinuous shocks into smooth steady waves. However, even turning the SPH viscosity on at all times does not keep the ball completely intact with the rectangular and hexagonal griddings, as shown in Figure 4.36 and Figure 4.37, respectively.

The tennis ball problem clearly demonstrates that artificial numerical fracture can occur in SPH calculations. As shown in the plate impact problem, the SPH form of the viscosity produces an undesirable artificial shear strength which can alter solutions. However, it also acts to partially mitigate numerical fracture, in that it inhibits relative particle motion. In the next section it is shown that numerical fracture is actually a result of a numerical instability in the SPH algorithm.

4.9 Tensile Instability Tests

The numerical fracture seen in the tennis ball impact calculations can be understood by examining simple test configurations which illustrate that the SPH method is unstable in tension. Consider a one-dimensional string of equally spaced particles, far enough away from the boundaries that signals from the ends do not reach the middle during the time of the calculation. Although it does not matter to the qualitative behavior of the calculation, the material properties are representative of hydrodynamic (zero yield strength) aluminum, and the particle spacing is 0.1 cm. All viscosities are turned off. Let the initial conditions for the problem be that velocities are zero, and the initial density is a fraction, f, of the ambient density of the material. If f is greater than 1, a constant compressive stress exists in the material, while if it is less than 1, a constant tensile stress exists. Put a very small perturbation velocity on one particle. If the stress is compressive, the perturbation creates small amplitude noise which propagates through the particles but becomes no larger than the initial perturbation. If the stress is tensile, the perturbation grows exponentially.

Figure 4.38 shows velocity histories of the perturbed particle for various values of the initial density, resulting in tensile stresses from 40 kilobars to less than a kilobar. The initial velocity of the particle is 10^{-10}, and if the stress is compressive, it does not exceed this value. If the stress is tensile, the amplitude grows by eight orders of magnitude. The instability apparently occurs for any tensile stress amplitude, although the growth rate is lower for smaller values of the tensile stress. The growth does not continue indefinitely, however, which makes the instability difficult to detect. When the displacement of the particle becomes large enough that compressive stresses are generated in any region, growth of the perturbation stops. The stress in the perturbed region fluctuates about zero, and contact is lost between particles whose separation exceeds the smoothing length, creating a numerical fracture. Figure 4.39 shows the stress profile after separation has occurred for an initial density 0.95 of ambient. Rather than the uniform 40 kbar tensile stress which should be present, the perturbation has grown catastrophically, creating a fracture and reducing the stress in the region of the fracture to particle-to-particle noise which averages zero. Figure 4.40 shows the position of each particle, and the point of separation between the particles is clear. However, evidence of the instability is not dramatic if only particle positions are examined, and it is not obvious that the separations in the plate penetration calculations, for instance, are actually due to a numerical instability.

More dramatic evidence of the instability can be seen in a two-dimensional analog of the above calculation. Figure 4.41 shows a two-dimensional array of particles placed on a rectangular lattice. The initial densities of the particles are set to create either a compressive or tensile stress, as described above. The positions of the three rows of particles next to the boundary are fixed so that release waves do not propagate in from the boundary. Give the center particle a perturbation velocity of 10^{-10} as before. The perturbation growth mimics that shown in Figure 4.38. If the

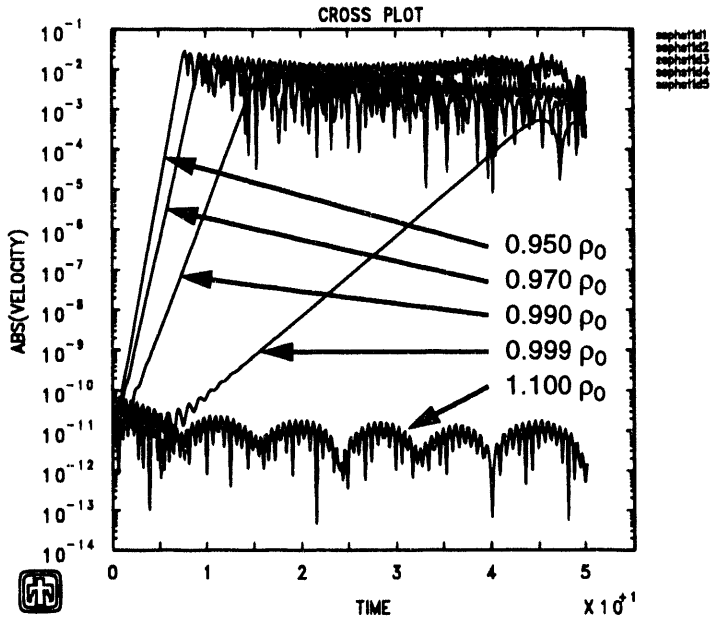


Figure 4.38 Perturbation amplitudes

initial stress is compressive, the perturbation is stable, and no detectable change occurs in the particle positions. If the initial stress is tensile, the rectangular lattice 'fractures' into the configuration shown in Figure 4.42.

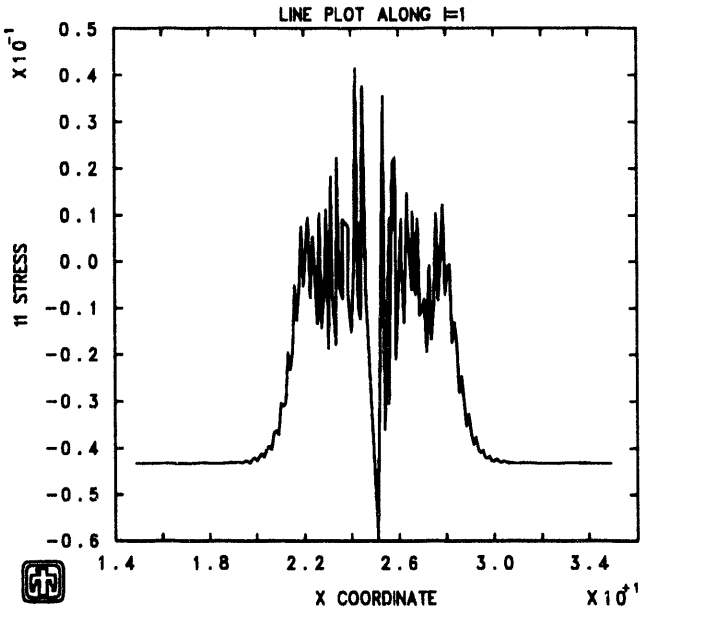


Figure 4.39 Stress profile after growth of instability.

Naturally, the instability of the method can be detected in situations other than these simple test problems which have all viscosities turned off. For instance, the

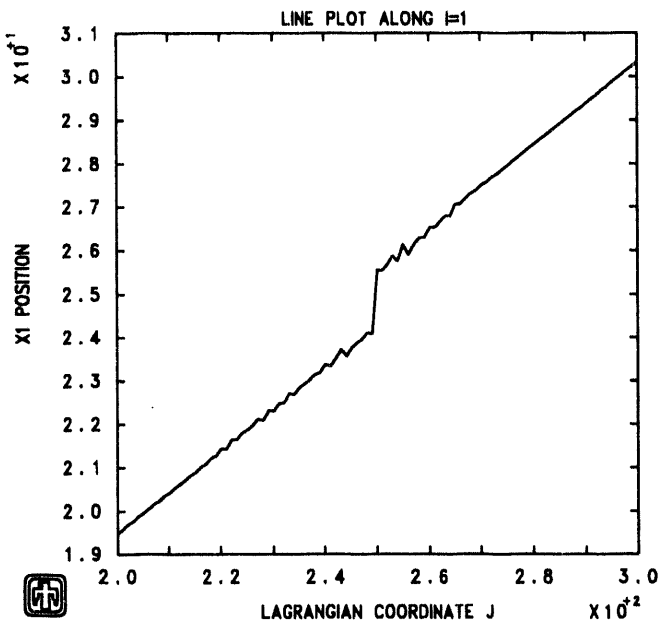


Figure 4.40 Particle positions after growth of instability.

low velocity contact impact problem illustrated in Figure 4.2 goes unstable if the calculation is continued far enough. The shocks generated at the impact interface travel to the free surfaces, and reflect as rarefactions which travel back toward the impact interface. The calculation exhibits no difficulties until the rarefactions meet, forming an unstable tensile region. Growth of the instability starts at the impact interface and continues until a stress profile similar to that in Figure 4.39 is generated, creating a numerical fracture. Of course, artificial viscosity is required in such a calculation, and has an effect on the growth of the instability. As shown above, the basic SPH algorithm is unstable for any tensile stress amplitude, although the growth rate is lower for smaller values of the tensile stress. Inclusion of either form of the viscosity mitigates the growth of the instability, and can stabilize the method if the tensile stress is small enough. The SPH viscosity is more effective at stabilizing the method, although the existence or value of a threshold tensile stress for instability growth has not yet been established as a function of the amount or type of viscosity. In the next section, however, a rigorous SPH stability analysis is performed which explains the results seen in the above test problems.

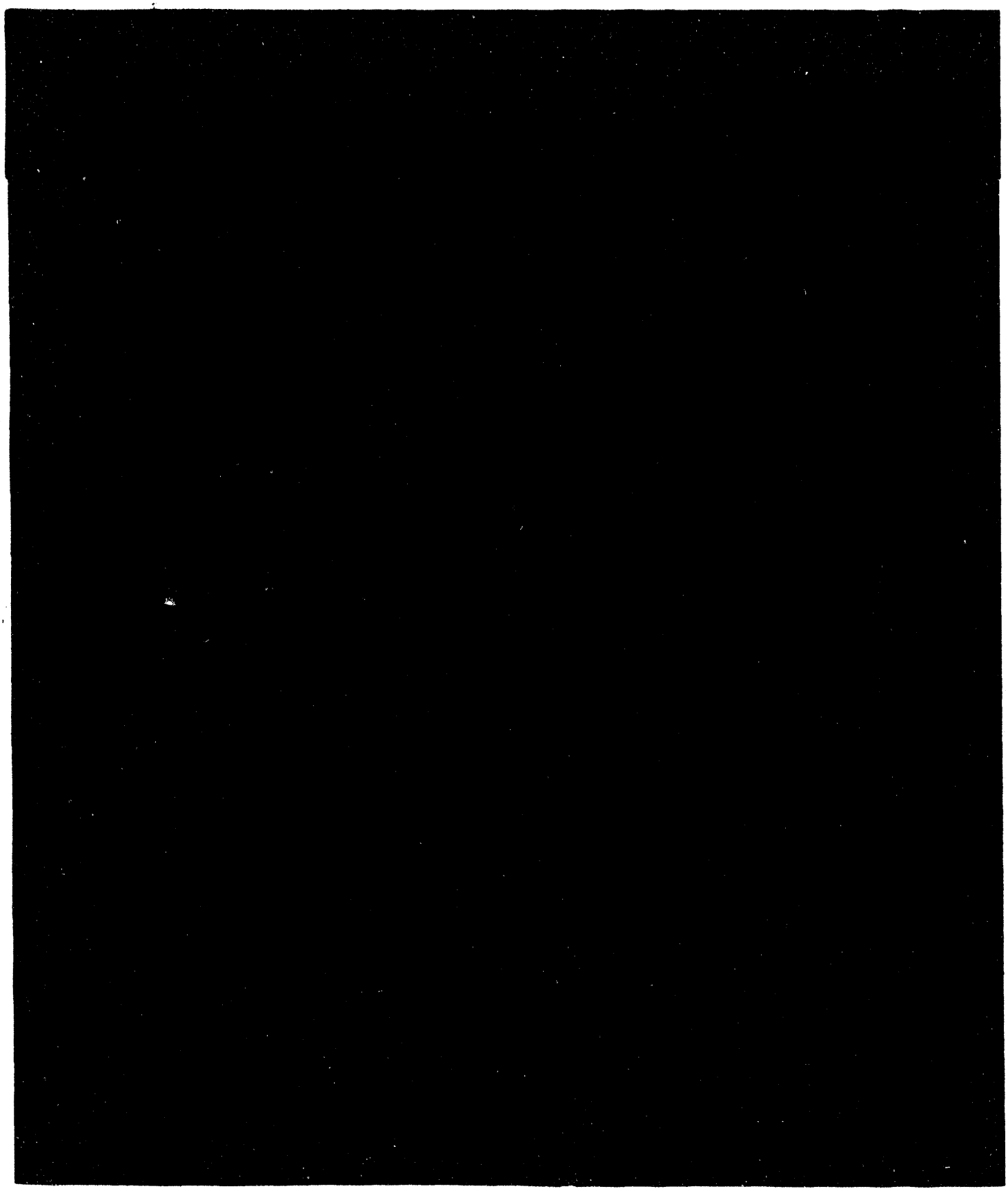


Figure 4.41 Initial rectangular lattice for stability test

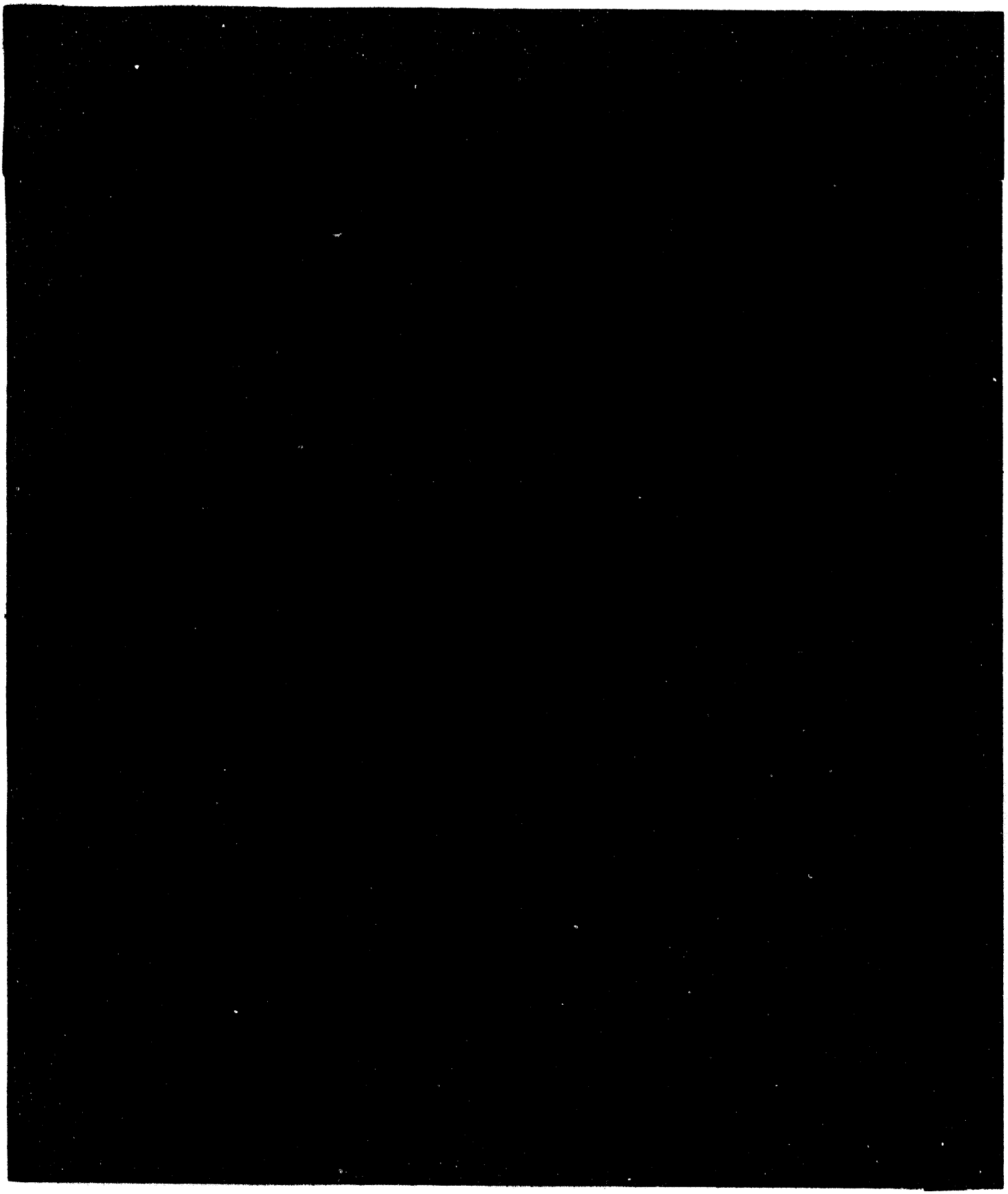


Figure 4.42 Rectangular lattice after instability growth

5. Stability Analysis

The tests of the SPH method detailed in the previous section demonstrate problems in tension. The presence of an instability is indicated, but it is not possible to fix the problem without knowing its exact cause. A rigorous analysis of the SPH algorithm which unambiguously determines the stability of the method has been sought for a number of years, but a stability analysis is complicated by the lack of connectivity between particles. In the present section an equivalent one-dimensional difference scheme is obtained which is amenable to stability analysis while still capturing the essence of the instability. A rigorous stability analysis of this scheme identifies the criterion for stability or instability in terms of the stress state and the second derivative of the kernel function. The analysis explains the observation that the method is unstable in tension while apparently stable in compression, but shows that it is possible to construct kernel functions which are stable in tension and unstable in compression. The instability is shown to result not from the numerical time integration algorithm, but rather from an effective stress with a negative modulus being produced by the interaction between the constitutive relation and the kernel function. That is, changes in the effective stress act to amplify, rather than reduce, perturbations in the strain. The analysis and the stability criterion provide insight into possible methods for removing the instability.

5.1 One-Dimensional SPH Equations

In one dimension, conservation of mass, Eq. (2.32) becomes

$$\dot{\rho}^I = -\sum_{J=1}^{N} m^J (\dot{x}^I - \dot{x}^J) \frac{\partial W(u^{IJ})}{\partial x^J}, \tag{5.1}$$

while conservation of momentum, Eq. (2.43), becomes

$$\ddot{x}^I = -\sum_{J=1}^{N} m^J \left(\frac{\sigma^J}{\rho^I \rho^J} + \Pi^{IJ} \right) \frac{\partial W(u^{IJ})}{\partial x^J}, \tag{5.2}$$

where $\ddot{x}$ is acceleration, $\dot{x}$ is velocity,

$$u^{IJ} = |x^I - x^J|, \tag{5.3}$$

and for simplicity, the unsymmetrized conservation of momentum form has been used, which is derived from Eq. (2.23) rather than Eq. (2.24). The dependence of the kernel function on the smoothing length has not been explicitly included in these equations.

5.2 Simplified One-Dimensional SPH Equations

The most critical step in performing a stability analysis of the SPH equations is the reduction of the general forms involving sums over neighbor particles to simpler forms which are amenable to stability analysis but still retain the stability properties of the general equations. All simplifying assumptions which have been made for purposes of the analysis have been computationally verified to satisfy this requirement by direct substitution into the test code. In fact, calculations suggest that the simplified equations are more stable than the general forms.

If it is assumed that the smoothing length is equal to the initial interparticle distance, then nearest neighbors contribute to the particle sums, while next-nearest neighbors located a distance $2h$ away do not, since both the kernel function and its derivative are zero for $u^{IJ} \geq 2h$. Although the analysis will later be generalized to include an arbitrary number of neighbors, for the present only nearest neighbors are considered. Equations (5.1) and (5.2) then reduce to

$$\dot{\rho}^I = -m\left[\,(\dot{x}^I - \dot{x}^{I+1})\,W'(u^{I,I+1}) - (\dot{x}^I - \dot{x}^{I-1})\,W'(u^{I,I-1})\,\right], \tag{5.4}$$

and

$$\dot{x}^I = -m\left[\left(\frac{\sigma^{I+1}}{\rho^I\rho^{I+1}} + \Pi^{I,I+1}\right)W'(u^{I,I+1}) - \left(\frac{\sigma^{I-1}}{\rho^I\rho^{I-1}} + \Pi^{I,I-1}\right)W'(u^{I,I-1})\right], \tag{5.5}$$

where it has been assumed that the particles have been numbered in order of increasing position, x, so that $I-1$ is the index of the nearest neighbor in the negative direction, while $I+1$ is the index of the nearest neighbor in the positive direction. In writing these equations use has been made of the fact that in one dimension the derivative of the kernel function reduces to

$$\frac{\partial W(u^{IJ})}{\partial x^J} = \begin{cases} W'(u^{IJ}) & x^J > x^I \\ -W'(u^{IJ}) & x^J < x^I \end{cases}, \tag{5.6}$$

where the prime on W refers to the derivative with respect to the argument. In one dimension, W has dimensions of length^{-1}, while the mass, m, should be interpreted

as mass per unit area, with the cross-sectional area numerically equal to one. The dimensions of W' are thus length^{-2}.

While it is possible to perform a stability analysis on Eqs. (5.4) and (5.5), the algebra becomes quite tedious, and clarity dictates further simplification of the equations before proceeding. If the densities in the denominators of Eq. (5.5) are taken to be constant for small perturbations, then

$$\ddot{x}^I = -\frac{m}{\rho^2}\left[(\sigma^{I+1} + Q^{I,I+1})\, W'(u^{I,I+1}) - (\sigma^{I-1} + Q^{I,I-1})\, W'(u^{I,I-1}) \right], \quad (5.7)$$

where

$$Q^{I,I+1} = \rho^2 \Pi^{I,I+1}. \quad (5.8)$$

Time derivatives will be approximated by centered-difference expressions, so that an equation of the form

$$\frac{df}{dt} = g \quad (5.9)$$

will become

$$\frac{f^{n+\frac{1}{2}} - f^{n-\frac{1}{2}}}{\Delta t} = g^n, \quad (5.10)$$

where the superscripts involving n denote the time at which the quantity is evaluated.

The one dimensional SPH equation of motion thus becomes

$$\dot{x}^{I,n+\frac{1}{2}} - \dot{x}^{I,n-\frac{1}{2}} = -\frac{m\Delta t}{\rho^2}\left(T^{I+1} W'^{I+\frac{1}{2}} - T^{I-1} W'^{I-\frac{1}{2}} \right)^n, \quad (5.11)$$

where

$$W'^{I+\frac{1}{2}} = W'(u^{I,I+1}), \quad (5.12)$$

the total stress, T, is given by

$$T^{I+1} = \sigma^{I+1} + Q^{I+\frac{1}{2}}, \tag{5.13}$$

and the linear term in Eq. (2.38) reduces in one dimension to

$$Q^{I+\frac{1}{2}} = Q^{I,I+1} = \rho^2 \Pi^{I,I+1} = \alpha' \rho c \, (\dot{x}^{I+1} - \dot{x}^{I}) = \alpha \, (\dot{x}^{I+1} - \dot{x}^{I}) \, , \tag{5.14}$$

where α' is the original dimensionless coefficient from Eq. (2.38), and additional constants have been included in the dimensional constant α.

The only density dependence in Eq. (5.11) comes from the dependence of the stress, σ, on density. A linear dependence of stress on strain (Hooke's Law) is described by the relation

$$\sigma = -K\eta = \rho_0 c^2 \left(1 - \frac{\rho_0}{\rho} \right), \tag{5.15}$$

where $K = \rho_0 c^2$ is the bulk modulus, η is the volume strain, c is the sound speed, and stress is taken negative in compression. Rather than using the continuity equation (conservation of mass) to find the strain, it can be noted that in one dimension, the volume associated with an interpolation point is determined by the positions half-way between the point and its neighbors to the left and right. The density can thus be determined from

$$\rho^I = \frac{2m}{x^{I+1} - x^{I-1}} \, . \tag{5.16}$$

Therefore, the stress can be expressed in terms of the particle positions by

$$\sigma^I = -K \left[1 - \frac{\rho_0}{2m} (x^{I+1} - x^{I-1}) \right]. \tag{5.17}$$

Calculations using this simplified relation show that it produces solutions consistent with those provided by the continuity equation, and in fact possesses smaller growth rates in unstable regimes.The only variables in Eq. (5.11) are thus velocity and position, which are related by

$$x^{I,n+1} = x^{I,n} + \Delta t \dot{x}^{I,n+\frac{1}{2}}. \tag{5.18}$$

Equations (5.11) and (5.18) constitute the simplified SPH equations to be analyzed for stability.

5.3 Stability Analysis Sketch

The SPH equations will be subjected to a stability analysis of a type which was popularized, beginning in the 1940's, by John von Neumann (1903-1957); the basic idea can be traced back to Jean Baptiste Joseph Fourier (1768-1830). A von Neumann stability analysis consists of the following steps[16]:

1. Obtain the linearized equations of first variation, which describe the propagation of small perturbations in the original equations. This is done by applying perturbations of the form

$$x \rightarrow x + \delta x \tag{5.19}$$

to each of the variables, then subtracting the unperturbed equations, keeping only terms which are first order in δx. All unperturbed quantities in the equations are frozen; the coefficients of the perturbations are considered constants. The resulting equations are known as the perturbation propagation equations.

2. Perform a Fourier analysis of the perturbation propagation equations, which involves assuming a separation of variables solution of the form

$$\delta x(X, t) = \delta x(t)\, e^{ikX}, \tag{5.20}$$

where X is the Lagrangian coordinate and k is the wavenumber of the perturbation.

3. For the resulting system of equations, find the amplification matrix $\underset{\sim}{A}$, defined by

$$\underset{\sim}{U}^{n+1} = \underset{\sim}{A}\,\underset{\sim}{U}^{n}, \tag{5.21}$$

where $\underset{\sim}{U}^{n+1}$ is the vector of values at the new time step, and $\underset{\sim}{U}^{n}$ is the vector of values at the old time step. The eigenvalues of $\underset{\sim}{A}$, which depend on the wavenumber of the perturbation, determine the stability of the system of equations. If the largest eigenvalue exceeds unity, the amplitude of the value vector is exponential in time.

5.4 Instability Criterion

The details of the stability analysis are given in Appendix B. The shortest wavelength, λ_{min}, capable of being resolved by the discrete system is twice the

particle spacing. The results show that at λ_{min}, a sufficient condition for unstable growth is

$$W''T > 0 \tag{5.22}$$

where W'' is the second derivative of W with respect to its argument, and is thus the slope of W'. The convention which is used throughout is that the stress T is negative in compression and positive in tension. The instability condition is independent of the artificial viscosity and the form of the kernel function. There are no stress or strain thresholds for the onset of the instability. The condition involves only the sign of the product of the total stress times the second derivative of the kernel function.

Figure 5.1 schematically summarizes the stability regimes for the specific case of

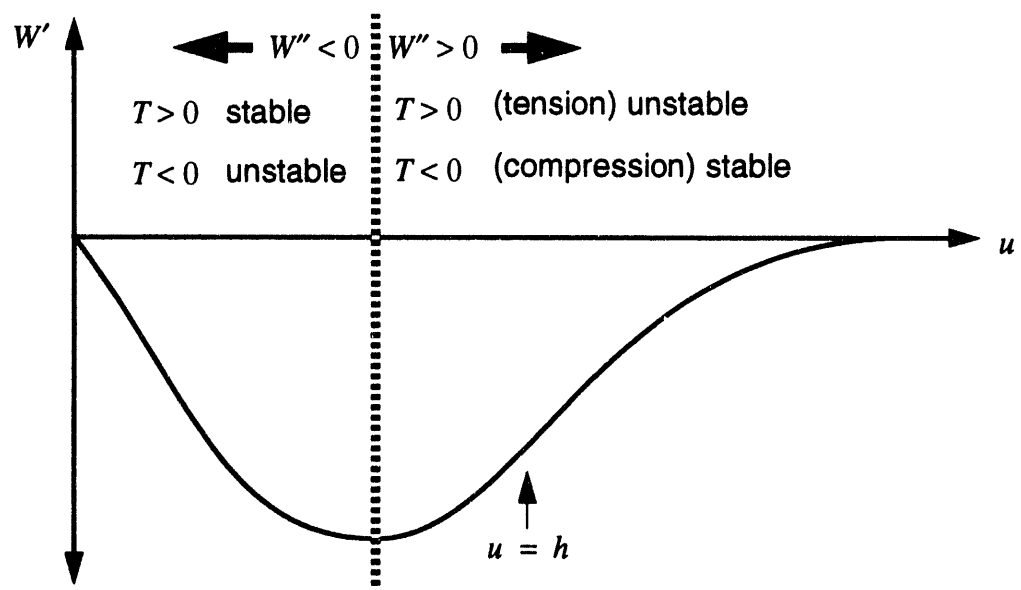


Figure 5.1 Stability Regimes for the cubic b-spline kernel

the cubic b-spline kernel. If the slope of the derivative of the kernel function is positive, the method is unstable in tension and stable in compression. If the slope is negative, it is unstable in compression and stable in tension. The derivative of the cubic b-spline kernel has its minimum value at $u = 2/3h$. In the standard configuration in which the particle spacing is equal to the smoothing length, the nearest-neighbor particles are at $u = h$, and the next-nearest-neighbors are at $u = 2h$ and do not interact. Thus, the standard configuration is stable if the stress is compressive, but unstable if it is tensile.

5.5 Physics of Instability Growth

While Eq. (5.22) provides a precise mathematical condition for instability, it does not provide a simple physical explanation of why SPH is unstable. In this section arguments are made to explain the reasons for the behavior in terms of the form of the equations derived in the course of the stability analysis. The mathematical models presented here are concerned with providing an intuitively understandable picture of the physics of the instability. This provides insight into the properties of the kernel function which are responsible for the instability and thus indicates possible means of removing the instability.

In one dimension, conservation of momentum is expressed by

$$\ddot{x} = \frac{\partial \sigma}{\rho_0 \partial X}, \tag{5.23}$$

where viscous stresses have been ignored, so that $T = \sigma$. In the discretization process associated with standard finite-difference methods, the partial derivatives are replaced by differences, so

$$\ddot{x} \approx \frac{\Delta \dot{x}}{\Delta t} = \frac{\Delta \sigma}{\rho_0 \Delta X}. \tag{5.24}$$

However, $\rho_0 \Delta X$ is just the mass, so the finite-difference equation of motion has the form

$$\ddot{x} \approx \frac{\Delta \sigma}{m} \propto \Delta \sigma. \tag{5.25}$$

However, Eq. (5.11) shows that the form of the SPH equation of motion is

$$\ddot{x} \propto \Delta \left(-\sigma W' \right), \tag{5.26}$$

where the difference operator Δ is a result of the sum over particles. Comparison of Eqs. (5.25) and (5.26) shows that in SPH the effective stress is not $\sigma(\varepsilon)$ but $-\sigma(\varepsilon) W'(\varepsilon)$. Figure 5.2 shows that when the normal equation-of-state stress is multiplied by a function which eventually goes to zero as particles approach or separate, the resulting effective stress must also go to zero. This results in regimes where the slope of the stress-strain curve changes sign, producing a negative modulus, or an imaginary sound speed. Stress-strain curves with this slope are clearly unstable, since changes in stress act to amplify, rather than reduce, any applied strain.

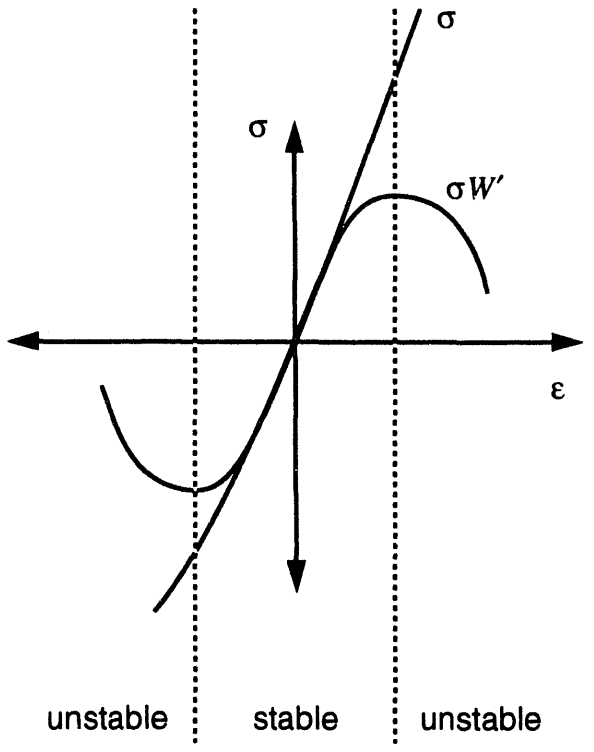


Figure 5.2 Interaction of the stress and the kernel function

Figure 5.2, while showing how the kernel function can introduce non-linearities into the problem which can cause instabilities, seems to indicate that an unstable regime would not be entered until enough strain had accrued to cause the product of the equation-of-state stress times the kernel function to change slope. However, the instability criterion, Eq. (5.22), shows no thresholds. This is due to the fact that Eq. (5.26) does not include the frequency dependence of the stability analysis leading to Eq. (5.22).

Referring to the details of the stability analysis in Appendix B, it can be seen that Eq. (B.10), the perturbation propagation equation, has the form

$$\delta\ddot{x} = \frac{\Delta\delta\dot{x}}{\Delta t} \propto \Delta\left(\delta\left(-\sigma W'\right)\right) = \Delta\left(-\sigma\delta W' - W'\delta\sigma\right) , \tag{5.27}$$

where the coefficients of the perturbations are constants, and again viscous stresses have been ignored, so that $T = \sigma$. However, when the frequency dependence of the perturbation is considered, as in Eqs. (B.24) to (B.27), it is found that at the shortest wavelength,

$$W'\delta\sigma = 0 , \tag{5.28}$$

Thus, at λ_{min},

$$\delta\ddot{x} \propto -\sigma\Delta(\delta W') \ , \tag{5.29}$$

or

$$\ddot{x} \propto -\sigma\Delta W' \ . \tag{5.30}$$

Comparison with Eq. (5.25) shows that at the minimum wavelength, the effective stress is just a constant multiple of the kernel function derivative, dependent only on the signed magnitude of the stress. Thus, as shown in Figure 5.3, the effective

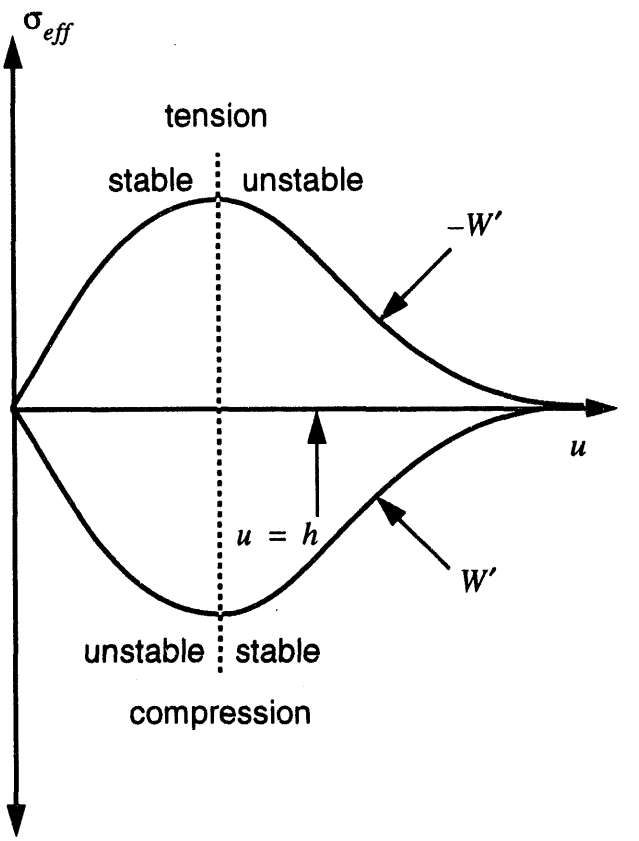


Figure 5.3 Effective stress at the minimum wavelength

stress is just an image of the kernel function.

If the stress is compressive, σ is negative and the effective stress is proportional to W'. When the slope of W' is positive, compressive stress decreases as particles separate, and increases as particles approach, which is stable. When the slope of W' is negative, compressive stress increases as particles separate, and decreases as particles approach, which is unstable.

If the stress is tensile, σ is positive and the effective stress is proportional to $-W'$. When the slope of $-W'$ is positive, tensile stress increases as particles separate, and decreases as particles approach, which is stable. When the slope of $-W'$ is negative, tensile stress decreases as particles separate and increases as particles approach, which is unstable.

The above constitutes a lengthy statement of the condition expressed so concisely in Eq. (5.22), but it does give a physical interpretation of the instability condition. The normal situation in which the cubic b-spline kernel is used with nearest-neighbor particles located at $u = h$ is unstable in tension and stable in compression. However, it can be seen that whenever the slope of W' is not zero, either tension or compression will be unstable. This result is independent of the numerical time integration algorithm, being instead the result of the kernel approximation producing an effective stress with a negative modulus.

5.6 Instability Criterion for an Arbitrary Number of Neighbors

The stability criterion derived in Appendix B assumes that only nearest neighbors interact, in that it includes terms only at $I \pm 1$. However, having the analysis as a guide, it becomes clear that extension to include an arbitrary number of neighbors will result only in extra terms in the equations for particles at $I \pm 2$, $I \pm 3$, $I \pm 4$, etc. For instance, each term in Eq. (B.10) will be repeated with indices $I \pm 1$ changed to $I \pm 2$, then repeated again using $I \pm 3$, and so on. As these terms are carried forward they result in terms with different powers of E in Eq. (B.24). In these equations, the values of W' and W'' have been evaluated at u equal to the distance from particle I to particles at $I \pm 1$. These values will change for particles at $I \pm 2$, etc., but if it is still assumed that other quantities such as stress are the same at all particles, the generalized form of the instability criterion can easily be demonstrated to be

$$(W''^{I,I+1} + W''^{I,I+3} + W''^{I,I+5} + \ldots)\, T > 0. \tag{5.31}$$

Thus, it is just the sum of the values of W'' at all particles with odd separations from particle I falling within the smoothing length $2h$ which determines whether the system is stable in tension or compression. The particles at even separations, such as $I \pm 2$, $I \pm 4$, etc., do not contribute because they result in terms in Eq. (B.27) which contain $\cos 2l\pi - 1$, where l is an integer, while the odd particles result in terms which contain $\cos(2l+1)\pi - 1$. Thus, various situations can result, as indicated in Figure 5.4 depending on the relation of the smoothing length to the initial particle spacing. In this figure, the even particles which do not contribute have been crossed out. Note that for the cubic b-spline kernel, a smoothing length of $1.5\Delta X$ results in odd particles being located where the slope of W' is zero. This configuration is thus stable in 1D for very small perturbations, but for larger

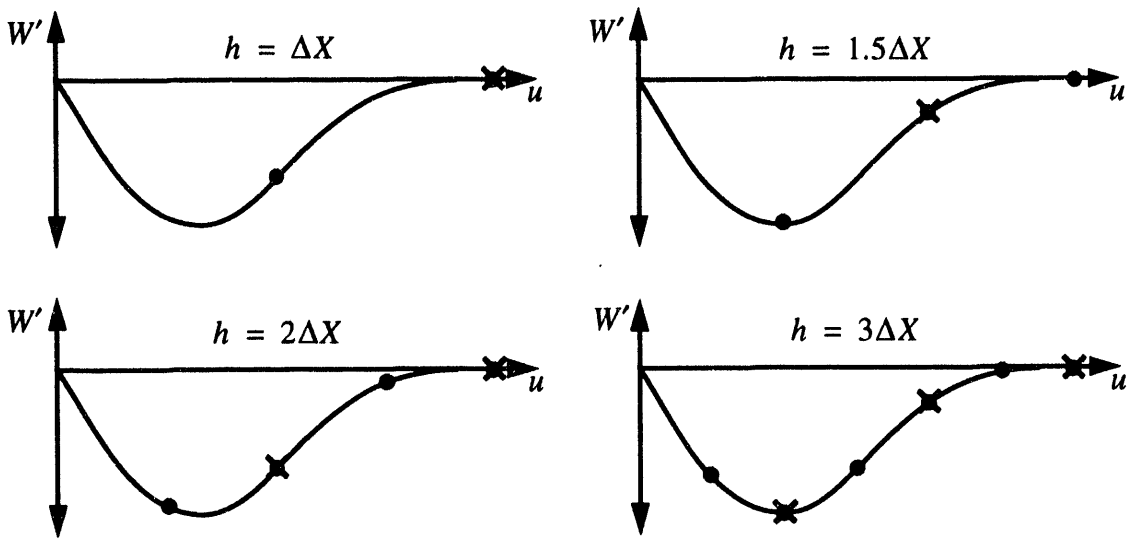


Figure 5.4 Particle-kernel interactions for various smoothing lengths

perturbations the particles move to regions of non-zero slope and instability again results.

5.7 Discussion

The stability analysis has resulted in a a simple criterion for the stability or instability of the method. The criterion for instability depends only on the sign of the stress and the slope of the kernel function derivative, and there are no stress or strain thresholds for the instability. The instability is not related to the time integration algorithm, but is instead due to a negative effective modulus resulting from the interaction of the kernel function with the constitutive relation. The instability criterion is independent of the viscosity coefficients, so the instability cannot be removed by increasing the amount of artificial viscosity. However, the perturbation growth rate does depend on the stress level and the amount of artificial viscosity. Thus, whether or not effects of the instability are observed depends on the perturbation growth rate, the amplitude of the perturbations, and the amount of time the system remains in an unstable regime. The instability manifests itself as a clumping of the particles which resembles fracture and fragmentation, but is in fact a numerical artifact.

6. Coupling Particle Methods with Lagrangian Methods

We presented a derivation of the SPH method that shows that it can be reduced to a Lagrangian weighted residual method in Section 3. The SPH method can be easily embedded within existing finite element code architecture, if the particles are viewed as elements whose connectivity must be determined for each time step. To embed the SPH method within a finite element code:

- SPH particles are treated as elements with only one node.
- A kernel sum approximation is used to compute the velocity gradient and stress divergence.
- Constitutive relations for particle elements and finite elements are the same and remain unchanged.
- Algorithms for kinematics of large strain and large deformation are the same for particle elements and finite elements.
- A particle search algorithm is required to determine particle interaction.
- A contact surface can be used to couple the finite element mesh to the particle elements.

7. Contact Coupling of Particle and Finite Element Methods

A contact algorithm can be used to couple the motion of SPH nodes to finite element surfaces. Contact detection algorithms for finite element methods define a set of nodes called slave nodes and a set of surface patches called master surfaces. For a finite element mesh, a slave node is simply a nodal point on the surface of the mesh. A master surface is defined using the side of a finite element on the surface. For particle method/finite element method coupling, the SPH nodes are treated as slave nodes.

Contact detection is accomplished by monitoring the displacements of the slave nodes throughout the calculation for possible penetration of a master surface. Following contact detection, a contact constraint is defined so that the slave node is "pushed back" to remain on the master surface. Based on this description, it is convenient to separate contact algorithms into a location phase and a restoration phase. The location phase consists of a neighborhood identification and a detailed contact check. The neighborhood identification matches a slave node to a set of master surfaces that it potentially could contact. The detailed contact check determines which of the candidate master surfaces is in contact with a slave node, the point of contact, the amount of penetration, and the direction of push-back. The point of contact, amount of penetration, and the direction of push-back define a contact constraint that is then enforced in the contact enforcement or restoration phase of the contact algorithm. This constraint is enforced in the following time step or possibly over several time steps.

7.1 Location phase

Only an outline of the location phase will be presented here; the reader is referred to Heinstein, et al[17] for a complete description of the algorithms used in the neighborhood identification and detailed contact check.

During the location phase, a subset of the SPH nodes which are in the vicinity of a master surface is collected for a later detailed contact check. This subset is formed using the point-in-box search algorithm where a capture box is defined around the master surface and a global search for all SPH nodes inside this capture box is performed. The known locations of contacting surfaces and their velocities are used to construct a master surface capture box. This guarantees that only physically meaningful contacts are considered in the detailed contact check.

The detailed contact check uses projected motions of the particle and surface. Both the point of contact and the direction of push-back for each slave node are determined during the detailed contact check. The position and velocity of both the

slave node and master surface are considered in determining initial contact. This results in a physically correct determination of the contact location.

A distinction between a concave and convex surface is made for slave nodes already in contact with a master surface. This results in a more accurate determination of the point of contact, amount of penetration, and the direction of push-back. The location phase run time is proportional to $m\log n$, where m is the number of master surfaces and n is the number of slave nodes.

7.2 Contact enforcement:

For the contact enforcement, a predictor-corrector method is used. First, the location of master surfaces and slave nodes assuming no contacts is predicted by

$$\hat{a} = \frac{f}{m} \qquad \text{(finite element method)}, \tag{7.1}$$

$$\hat{a} = \sum m^J \left(\frac{\sigma^J}{\rho^I \rho^J} \right) \frac{\partial W}{\partial \underset{\sim}{x}^J} \quad \text{(smoothed particle method)}, \tag{7.2}$$

$$\hat{v} = v + \Delta t\, \hat{a}, \tag{7.3}$$

$$\hat{x} = x + \Delta t\, \hat{v}, \tag{7.4}$$

where $\hat{a}$, $\hat{v}$, and $\hat{x}$ are the predicted acceleration, velocity and position respectively. The detailed contact check results in a calculated depth of penetration for each slave node into the master surface

$$\delta = max(\hat{n} \cdot (\bar{x} - \hat{x}), 0)\,. \tag{7.5}$$

The contact constraint is satisfied by simultaneously applying a contact force to the slave node and the master surface so that the penetration is removed during the next time step. The application of this penalty force will result in both the surfaces moving, and therefore the force must be determined with an iterative method. The iterative method is outlined as follows:

Compute acceleration (or force) needed to cancel the slave node penetration assuming it is contacting a rigid surface

$$a_n = \frac{\delta}{\Delta t^2}, \tag{7.6}$$

or

$$f_s = \frac{\delta\, m}{\Delta t^2}. \tag{7.7}$$

Next, compute the resulting acceleration of master surface due to the application of all slave node forces. The resulting master surface nodal forces can be determined by (here we assume 2D, but calculations extend easily to 3D)

$$F_i = \sum_s \left(\frac{1}{2} - \xi\right) f_s, \text{ and} \tag{7.8}$$

$$F_{i+1} = \sum_s \left(\frac{1}{2} + \xi\right) f_s. \tag{7.9}$$

The forces acting on the master surface nodes are assembled and their accelerations are computed as

$$a_i = \frac{\sum F_{is}}{m_I}. \tag{7.10}$$

Since master surface has moved, the initial guess for penalty force must be corrected. The acceleration of the contact point on master surface due to the acceleration of master nodes is given by

$$a_{ps} = \left(\frac{1}{2} - \xi\right) a_{n1} + \left(\frac{1}{2} + \xi\right) a_{n2}. \tag{7.11}$$

This leads to a corrected penalty force

$$f_s = \frac{\delta_s\, m_s}{\Delta t^2} - a_{ps} m_s, \tag{7.12}$$

and a new master node acceleration

$$m_i a_i = \sum_s \left(f_{Is} - a_{ps} m_s\right). \tag{7.13}$$

Note: we should iterate to find the 'best' penalty force; however, one pass is usually all that is required for an accurate solution. Any errors in the contact enforcement will be accounted for in the next time step.

If only one iteration is done, the mass and force can be assembled to obtain the acceleration of all master nodes

$$\sum_s f_{Is} = \left(m_I + \sum_s m_{Is}\right) a_{nI}, \tag{7.14}$$

where $m_{1s} = \left(\frac{1}{2} - \xi\right) m_s$, $n_{2s} = \left(\frac{1}{2} + \xi\right) m_s$, $f_{1s} = \left(\frac{1}{2} - \xi\right) f_s$ and $f_{2s} = \left(\frac{1}{2} + \xi\right) f_s$.

After assembling and solving for the motion of the master surface, the slave node acceleration can be corrected to account for the relative motion between the slave node and the master surface

$$a_{ns} = a_{ps} - \frac{f_p}{m_s}. \tag{7.15}$$

Finally, the predicted accelerations for all nodes can now be corrected by

$$a = \hat{a} + a_n \quad . \tag{7.16}$$

In PRONTO, the accuracy of the penalty force is improved by using a symmetric (or partitioned) contact when two finite element surfaces are in contact. This allows both surfaces to act as the master for a portion of the time step. With SPH nodes, however, a strict master slave approach is required.

8. Example Coupled Problems

Two example problems are presented that demonstrate the ability to couple particle and finite element calculations. In the first example, a simple SPH mesh impacts a simple FEM mesh. The second example considers a thin structure that impacts water.

8.1 Two bars impacting

This example considers two one inch square copper bars impacting at 1000 in/s. The bar on the left was modeled using SPH elements, and the bar on the right was modeled using FEM. The purpose of this example is to show that the coupled method produces symmetric results. Figure 8.1 shows a plot of the pressure that results from the impact.

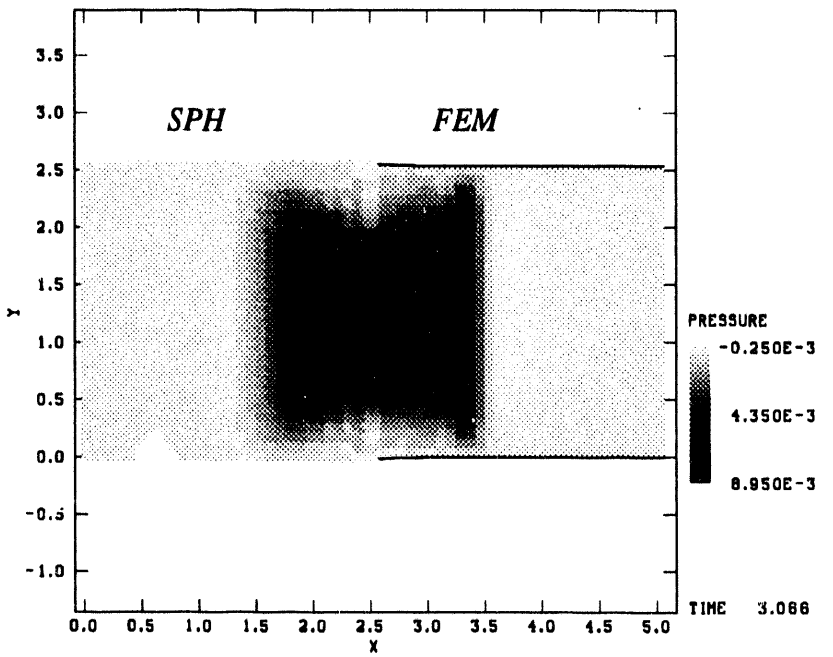


Figure 8.1 Two copper bars impacting.

The pressure from the finite element model was plotted by shading a square the size of the finite element. The SPH results were visualized by simply drawing a sphere with the correct intensity at the location of the SPH element. Since the SPH elements overlap, this simple method of plotting distorted the SPH result.

This example shows that the SPH method tends to be more diffusive than the finite element method. Here, both the SPH nodes and the finite element nodes had the same spacing. However, since SPH method requires more particles to interpolate, the effective h of the SPH mesh is bigger than the effective h of the finite element

mesh. Despite these differences, both methods predicted a similar pressure wave from the impact.

Point-to-point oscillations in the pressure can be seen in both the finite element mesh and the SPH mesh. Increasing the artificial viscosity will damp these oscillations at the expense of broadening the wave front.

8.2 Boat Impacting water

This example shows some of the advantages of combining the SPH and FEM method. Figure 8.2 shows a plane strain model of a 'boat'-like object impacting

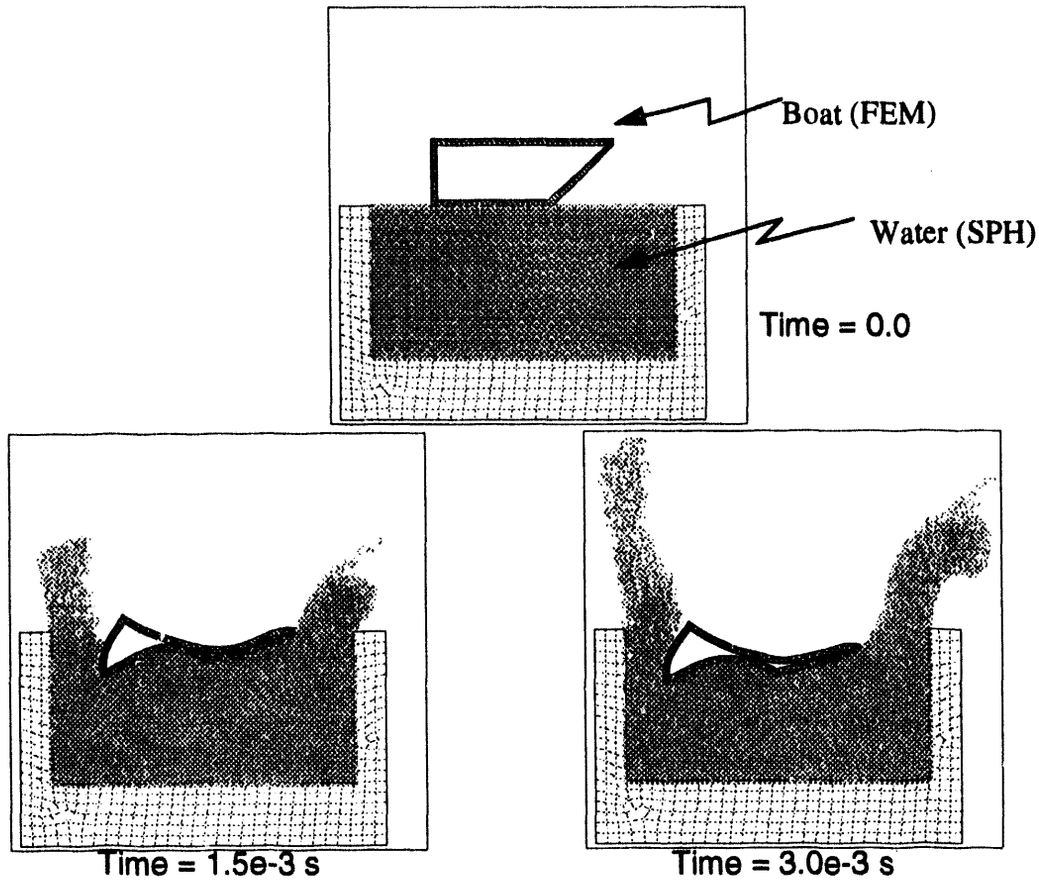
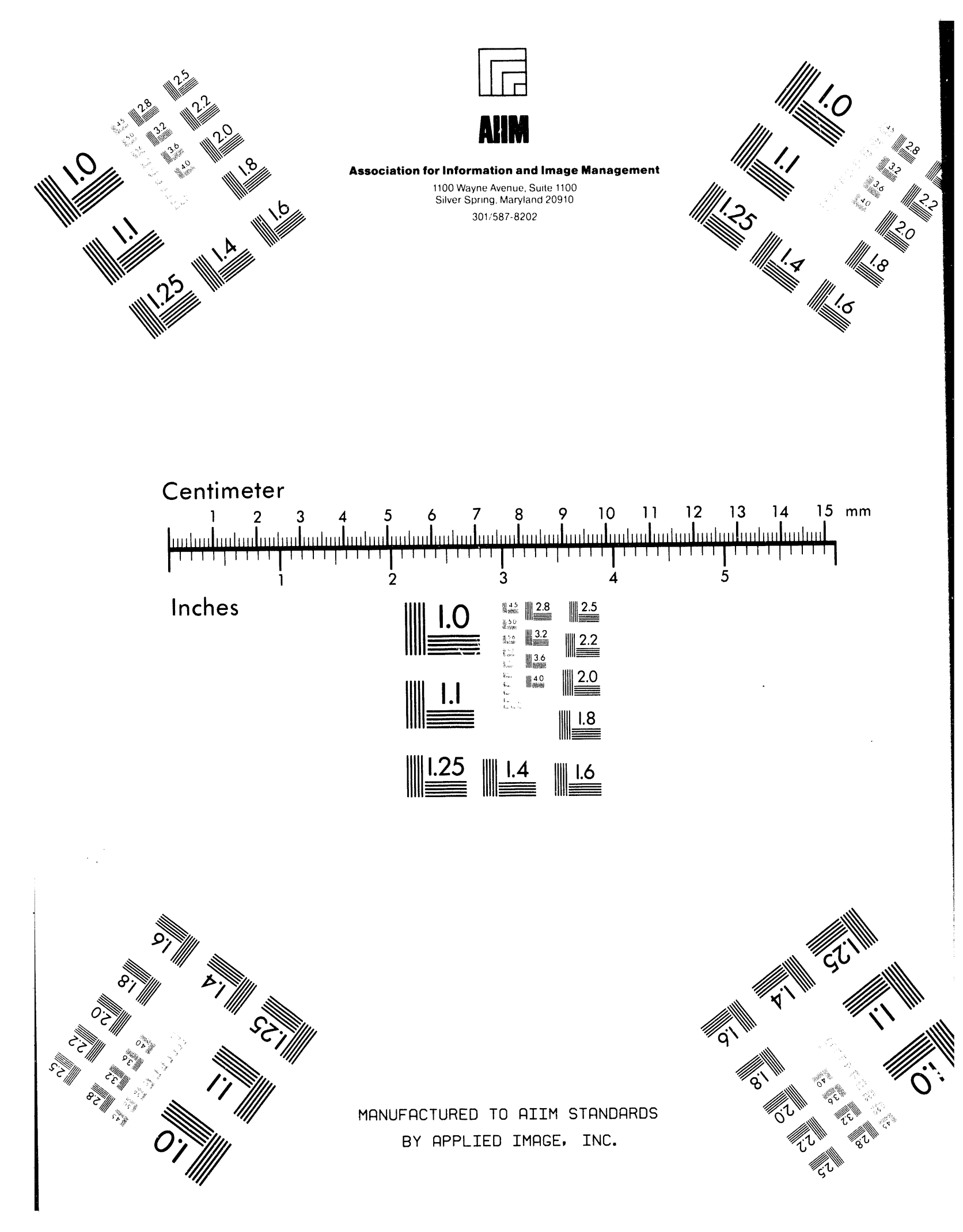


Figure 8.2 High speed impact of water by 'boat'. Impact speed = 2000 in/s.

water at very high speed. The 'boat' structure was modeled using four elements through the thickness. This thin structure would not be efficiently modeled using the SPH method. The nature of the SPH method would require 5 to 10 elements through the thickness of the boat. Since the SPH method is more accurate when the spacing between the nodes is uniform, many more nodes would be required than with the FEM method, which can have elements with high aspect ratios. Using SPH to model the boat would also require a shorter time step, since the distance


AIIM
Association for Information and Image Management
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910
301/587-8202
Centimeter
Inches
MANUFACTURED TO AIIM STANDARDS
BY APPLIED IMAGE, INC.

2 of 2

between the SPH elements would be smaller in order to accommodate the increased number of SPH elements. The splash of the water would be very hard to model with FEM. If the water was modeled using finite elements, the elements would distort and tangle resulting in an ill-posed mesh. The SPH method allows for the fluid behavior of the water and does not result in an ill-posed mesh.

A contact algorithm used between the FEM and SPH method allows for a mesh transition between the boat and the water. For this problem, four SPH elements contacted a single side of the finite element mesh used to model the boat.

This example was presented to demonstrate the advantages of a SPH-FEM coupling. The actual behavior of a plane strain water splash could differ from the behavior shown by the demonstration model. The water was modeled using SPH with a very simple equation of state. The behavior of water could be more complex than the equation of state can capture (i. e. steam formation).

9. Conclusion

Smoothed particle hydrodynamics (SPH) is a gridless Lagrangian technique which shows potential for detailed analysis of high deformation events which are not well handled at present by either Eulerian or standard Lagrangian techniques. In principle, the method should be able to overcome both the diffusion problems associated with Eulerian methods and the grid distortion associated with Lagrangian methods. The name 'smoothed particle hydrodynamics' is misleading, since the particles are actually interpolation points, and the method is not hydrodynamic, since inclusion of full stress and strain tensors is easily accomplished. The apparent strength of SPH is the calculation of spatial gradients by a kernel approximation method which does not require connectivity of the particles and should be able to treat arbitrary deformations. In the present study, the SPH algorithm has been subjected to detailed testing and analysis to determine its applicability in the field of solid dynamics.

One of the results of the study is that the basic SPH algorithm allows point-to-point fluctuations in the calculated variables, such as strain and stress, due to the inability of the SPH gradient operator to detect variations on the scale of the particle separation. This results from the combination of two factors. The first is that the lack of particle connectivity precludes the definition of information at inter-particle locations, since these are constantly being redefined as the material deforms. All information must therefore be located at the particle, or interpolation point, positions. The second factor is that kernel approximation is in fact a technique to produce a continuous functional fit to a discrete set of data, which is to say, curve fitting. If the discrete data oscillates rapidly with a wavelength equal to the particle separation, the functional fit to the data shows the same oscillations, but it naturally has its maxima and minima at the particle positions, so that the derivatives of the function are zero at the particles and non-zero in between. Since gradients are only evaluated at particle positions, such a rapid oscillation in field quantities is essentially transparent to the gradient operator, allowing noise to be locked into the solution rather than being equilibrated by particle motion.

Perhaps the most important result of the study is a rigorous stability analysis which provides a simple criterion for the stability or instability of the method. The criterion for instability depends only on the sign of the stress and the slope of the kernel function derivative, and there are no stress or strain thresholds for the instability. The instability is not related to the time integration algorithm, but is instead due to a negative effective modulus resulting from the interaction of the kernel function with the constitutive relation. In other words, the instability is due to an effective stress which amplifies, rather than reduces, applied strains. Although the effective modulus is frequency dependent, it is the smallest wavelength capable of being resolved by the discrete system, commonly called the noise frequency, which has the maximum growth. Thus particle-to-particle fluctuations which grow with time are the result of the instability. However, growth only continues so long as the

particle spacings result in a stress state and a value of W'' which satisfy the instability criterion. Particles will thus tend to clump together in stable configurations. This clumping resembles fracture and fragmentation, but is in fact a numerical artifact.

Although the usual situation involving the cubic b-spline kernel with the smoothing length equal to the particle spacing is unstable in tension, instability in compression is also possible for different kernels. Various schemes suggest themselves to create a kernel coupled with a variable smoothing length so that the system is stable for the current sign of the stress. For instance, a kernel with W'' everywhere equal to zero will be stable in all stress regimes, but such a kernel clearly cannot go to zero smoothly as the initial distance between particles increases. Unfortunately, this is one of the major requirements usually placed on the kernel in order to achieve compact support so that only particles within a limited range interact. Thus, local support results in at least some portion of the kernel function having W'' positive, which is unstable in tension. However, it seems unnecessary for the kernel to have regions where W'' is negative. A kernel modified in this way would eliminate the possibility of instability in compression. It remains the topic of further work to determine the required properties of the kernel function which maximize the accuracy and stability of the SPH method.

An important result of the analysis is that the instability criterion is independent of the viscosity coefficients, so the instability cannot be removed by increasing the amount of artificial viscosity. However, the perturbation growth rate does depend on the stress level and the amount of artificial viscosity. Thus, whether or not effects of the instability are observed depends on the perturbation growth rate, the amplitude of the perturbations, and the amount of time the system remains in an unstable regime. Also, the artificial viscosity commonly used in SPH has properties which differ from those of the standard finite-difference shock-spreading viscosity. The SPH viscosity inhibits relative particle motion, and while this can have adverse effects on the form of the solution, it also smooths point-to-point fluctuations and reduces instability growth in tension. It is possible that this form has attributes which mitigate some of the difficulties seen with the basic algorithm.

In the process of investigating the method, several extensions and improvements were made to the basic algorithm. The original development work in SPH was in the area of astrophysics and fluid dynamics[18] and only involved the mean pressure component of the stress. In this work, the treatment has been extended to include full stress and strain tensors so that material strength can be included. This development is straightforward and is being done concurrently by other workers[19], although it is not yet widespread.

An algorithm was also developed for the inclusion of variable smoothing length. While a density based variable smoothing length has previously been suggested[8],

the current implementation within the centered-difference formalism appears to be unique. It is clear that varying the smoothing length based on density alone will not be adequate for all deformations, and the present formalism is easily modified to base the smoothing length on, for instance, the maximum strain component. However, it is possible that certain types of deformations may require a directionally dependent smoothing length, rather than one based on radial distance only, which will entail major modifications to the SPH algorithm.

A further area of advancement involves the finite-difference viscosity treatment and the time step control. It seems to be part of the SPH lore that standard viscosity treatments do not work, so few if any current workers use anything but the SPH form of the viscosity. The current investigation has revealed that the finite-difference form of the viscosity is adequate to smooth shocks, and the SPH viscosity actually functions to mitigate other deficiencies in the algorithm. This investigation is apparently also the only one to consider a viscosity-based time step control rather than a simple CFL[13] criterion.

A major development has been the coupling of SPH with the transient dynamics finite element code, PRONTO. The ability to couple particle methods and finite element method allows fluid-structure interaction problems to be solved efficiently. The coupling embeds the SPH method within the finite element code and treats each SPH particle as an element within the finite element architecture. Contact surface algorithms used in the finite element method were modified to couple the SPH particles with the finite elements.

Although the current investigation has revealed areas in SPH that need improvement, the potential of the method in the area of large deformation Lagrangian calculations is very real. Further work should concentrate on finding a remedy for the instability and controlling the particle-to-particle noise.The SPH viscosity needs to be studied to understand and isolate those properties which appear to mitigate these difficulties. With the accomplishment of these objectives SPH can become an important computational tool for solid dynamics analyses.

References

1. L. B. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis," *A. J.* **82** (1977), 1013-1024.

2. R. A. Gingold and J. J. Monaghan, "Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics," *J. Comp. Phys.* **46** (1982), 429-453.

3. J. J. Monaghan, "Why Particle Methods Work," *SIAM J. Sci. Stat. Comput.* **3** (1982), 422-433.

4. J. J. Monaghan, "Particle Methods for Hydrodynamics," *Comput Phys. Rep.* **3** (1985), 71-124.

5. J. J. Monaghan, "An Introduction to SPH," *Comp. Phys. Comm.* **48** (1988), 89-96.

6. L. D. Cloutman, "Basics of Smoothed Particle Hydrodynamics," Lawrence Livermore National Laboratory report UCRL-ID-103698, 1990.

7. L. D. Cloutman, "An Evaluation of Smoothed Particle Hydrodynamics," T *proceedings of The NEXT Free-Lagrange Conference*, Jackson Lake Lodge, Moran, Wyoming, June 3-7, 1990.

8. W. Benz, "Smooth Particle Hydrodynamics: A Review", in The Numerical Modeling of Stellar Pulsation, ed. J. R. Buchler (Dordrecht: Kluwer), (1990), 269.

9. L. M. Taylor and D. P. Flanagan, "PRONTO 2D - A Two-Dimensional Transient Solid Dynamics Program", Sandia National Laboratories Report, SAND 86-0594, 1987.

10. J. W. Swegle, "TOODY IV - A Computer Program for Two-Dimensional Wave Propagation", Sandia National Laboratories Report, SAND78-0552,1978.

11. J. von Neumann and R. D. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks," *J. Appl. Phys.* **21** (1950), 232.

12. J. J. Monaghan and R. A. Gingold, "Shock Simulation by the Particle Method SPH", *J. Comp. Phys.* **52** (1983),374-389.

13. R. Courant, K. O. Friedrichs, and H. Lewy, *Math. Ann.* **100** (1928), 32.

14. P. M. Campbell, "Some New Algorithms for Boundary Values Problems in Smooth Particle Hydrodynamics," DNA-TR-88-286, Sept. 1988.

15. M. E. Kipp and R. J. Lawrence, "WONDY V - A One-Dimensional Finite-Difference Wave Propagation Code", Sandia National Laboratories Report, SAND 81-0930, 1982.

16. R. D. Richtmyer and K. W. Morton, Difference Methods for Initial Value Problems, Interscience, New York (1967).

17. M.W. Heinstein, S. W.Attaway, F. J. Mellow, J. W. Swegle, "A General-Purpose Contact Detection Algorithm for Nonlinear Structural Analysis Codes," SAND92-2141, Sandia National Laboratories, Albuquerque, New Mexico, May 1993.

18. W. Benz, "Applications of Smooth Particle Hydrodynamics (SPH) to Astrophysical Problems," *Comp. Phys. Comm.* **48** (1988), 97-105.

19. L. Libersky and A. G. Petschek, "Smooth Particle Hydrodynamics with Strength of Materials," *proceedings of The NEXT Free-Lagrange Conference*, Jackson Lake Lodge, Moran, Wyoming, June 3-7, 1990.

20. D. L. Hicks, "Stability Analysis of WONDY (A Hydrocode Based on the Artificial Viscosity Method of von Neumann and Richtmyer) for a Special Case of Maxwell's Law," *Mathematics of Computation* **32** (1978), 1123-1130.

APPENDIX A

Search Algorithm

The SPH algorithm requires an efficient means of determining which particles are within a specified distance of a given particle and will thus have a non-zero contribution to the kernel sums. This is a specific case of the more general problem of a search for all elements in a set whose positions fall within a specified region of space, which occurs in a variety of applications. This appendix describes a newly developed search algorithm which is appealing because of its simplicity, speed, and efficient use of memory.

In one dimension such a search can be done quite simply. The algorithm would involve sorting the elements based on position, then performing a search for the elements at the boundaries of the region. In higher dimensions tree and box techniques seem to be in favor. Each type of technique has disadvantages if elements are not optimally located. The logic involved can be quite complicated and the storage requirements can be severe. The boxing method in particular can have problems if elements are unevenly distributed in space so that a large number of boxes are required to cover the element space with many boxes being empty. The method presented here is based on an extension of the one-dimensional procedure, is quite independent of the spatial distribution of the elements, and has fixed storage requirements which are minimal compared to other types of algorithms.

The problem is to find all elements of a set whose members lie within a region of space specified by upper and lower limits on each spatial coordinate. Briefly, the algorithm consists of individual one-dimensional sorts of the elements using each coordinate value as the search key, followed by binary searches of each sorted list to find the elements at the boundaries of the search region. This produces separate lists of elements whose positions fall within the bounds on each coordinate. The list with the smallest number of elements is then compared with the bounds in the orthogonal coordinate directions to produce the final list. Details of each step are given below.

1. SORT
   Sort the elements into sequential lists (in order of increasing coordinate value) for each spatial coordinate. Since there are many other variables associated with each element, index lists are produced rather than reordering all information for all elements. This requires 2 arrays of length N (number of elements) for each coordinate. One is the index list giving the element number at each array location in the sorted list, and the other is the rank list giving the

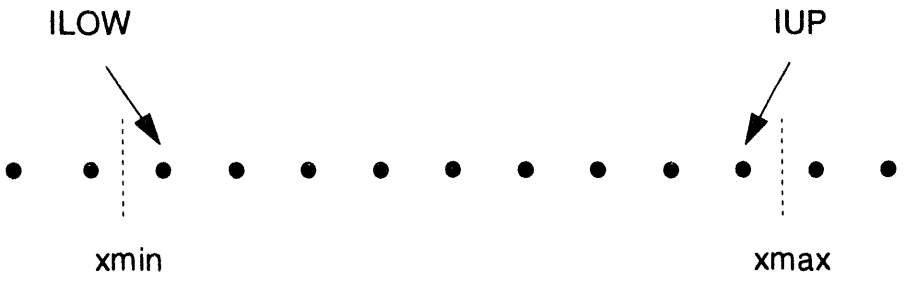
array location of each element in the sorted list. The latter is required to avoid searching the index list to find a given element. Examples of the various lists are shown below. The sorting can be done quickly with any of a number of

Table A.1 Example lists

List Position	1	2	3	4	5
Element Number	1	2	3	4	5
Coordinate value	12.1	4.6	18.3	7.5	2.8
Sorted coordinate list	2.8	4.6	7.5	12.1	18.3
Index list	5	2	4	1	3
Rank list	4	2	5	3	1

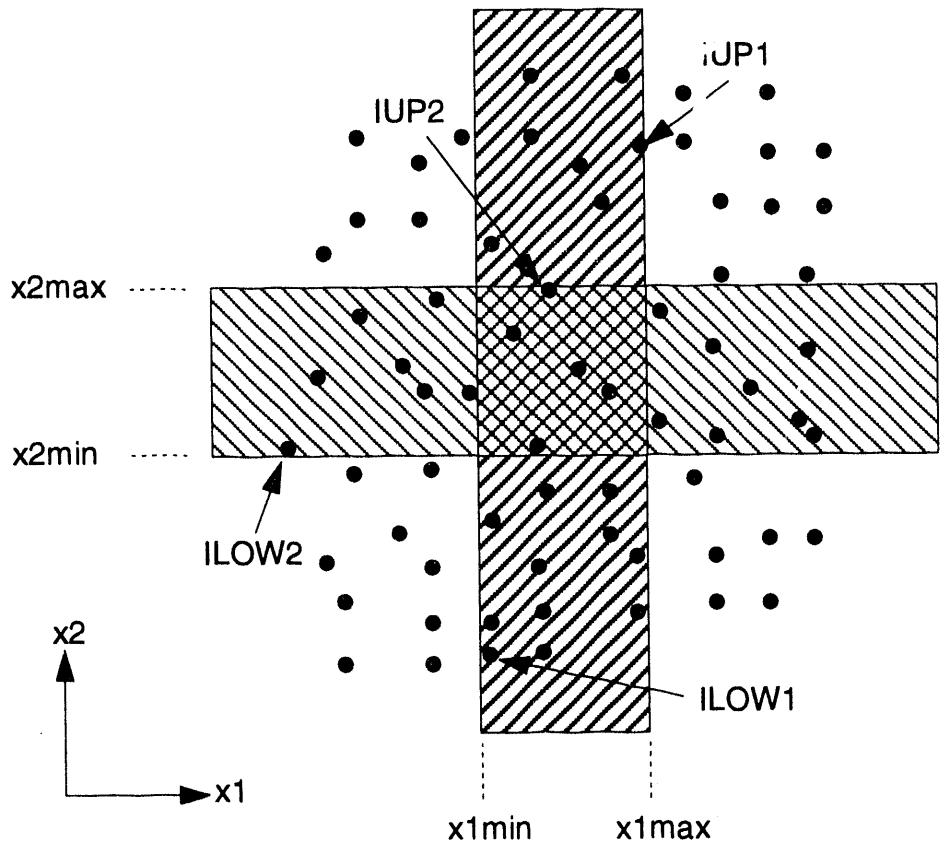
available sorting algorithms, the best of which have execution times on the order of N $LOG_2(N)$. The sort needs to be done only once for each configuration of the elements regardless of how many spatial regions are to be treated. Sorting need not be repeated until the element positions change.

2. SEARCH.
For each search region, use a binary search algorithm to find the first and last elements within the boundaries of the region in each coordinate direction, as illustrated below. The region is defined by minimum and maximum values,



xmin and xmax, for each coordinate. The binary search divides the list in half at each step and has an execution time of the order of $LOG_2(N)$. This operation requires no storage, simply defining two integers, IUP and ILOW, which are the array locations in the sorted list of the first and last elements in the region. The number of elements in the region is IUP-ILOW+1.

3. COMPARE
At this point there is one set of elements for each spatial coordinate, consisting of all elements which fall within the bounds on coordinate x1, all which fall within the bounds on x2, etc. The situation is illustrated below in two dimensions. The individual searches in the x1 and x2 directions yield the



elements in each shaded region, while the desired final list consists of the elements common to both sets, or those in the cross-hatched region. The determination of the final list is greatly simplified by the use of the rank arrays produced during the sort step.The list with the smallest number of elements is chosen. The rank of each element in the sorted list for each orthogonal direction is compared to the bounds, IUP and ILOW, for that coordinate. The final check thus involves only fast integer comparisons and vectorizes easily. Elements within the bounds are added to the final list of elements. This requires one array for the final list, which for generality should be of length N.

The logic of the algorithm is quite simple. The total storage requirement is (2*IDIM+1)*N, where IDIM is the dimension (2 or 3) of the problem.The final comparison step involves the only dependence on the spatial distribution of the elements, in that the number of elements in the smallest list will depend on the spatial orientation of the elements. However, tests indicate that this step takes negligible time compared to the sort and search steps, and the execution time of the total method is of order N $LOG_2(N)$ for N elements and N search regions.

The following is a listing of the FORTRAN subroutines required to implement the search algorithm.

```
      SUBROUTINE MKLST2(X1,IND1,IRNK1,X2,IND2,IRNK2,
     *XMIN1,XMAX1,XMIN2,XMAX2,NP,NDIM,LIST,NLIST)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      SAVE
      DIMENSION  X1      (NDIM),X2      (NDIM),
     *           IND1    (NDIM),IND2    (NDIM),
     *           IRNK1   (NDIM),IRNK2   (NDIM),
     *           LIST    (NDIM)
C
C  2D ROUTINE TO MAKE A LIST OF PARTICLES WHICH FALL WITHIN THE
C  RANGE XMIN1<X1<XMAX1 XMIN2<X2<XMAX2.
C  THE PARTICLE LIST HAS PREVIOULSY BEEN SORTED SEPARATELY ON THE X1
C  AND X2 COORDINATES. THE X1 AND X2 ARRAYS HAVE NOT BEEN CHANGED;
C  INSTEAD INDEX AND RANK ARRAYS HAVE BEEN CREATED.
C  THE INDEX ARRAY GIVES THE NUMBER OF THE PARTICLE AT EACH LOCATION
C  IN THE SORTED LIST, WHILE THE RANK ARRAY GIVES THE LOCATION OF EACH
C  PARTICLE IN THE SORTED LIST.
C
C  THE ALGORITHM IS TO DO INDIVIDUAL ONE-DIMENSIONAL CHECKS TO
C  FIND ALL THE PARTICLES WITHIN THE X1-LIMITS, THEN THE X2 LIMITS.
C  THIS IS DONE WITH CALLS TO GETBND.
C
C  IF ANY LIST HAS ZERO ELEMENTS, THERE ARE NO PARTICLES IN THE RANGE.
C
C  THE ONE-DIMENSIONAL LIST WITH THE SMALLEST
C  NUMBER OF PARTICLES THEN HAS EACH PARTICLE TESTED ON THE LIMITS
C  IN THE ORTHOGONAL DIRECTION TO PRODUCE THE FINAL LIST.
C
C  THE CHECK IN THE ORTHOGONAL DIRECTION CAN BE DONE BY USE OF
C  THE RANK ARRAY AND THE BOUNDS ON THE INDEX LIST DETERMINED BY
C  GETBND. IF THE RANK OF THE PARTICLE IS NOT WITHIN THE BOUNDS
C  DETERMINED BY GETBND, THE PARTICLE IS EXCLUDED.
C
C  INPUT
C  X1      -  X1 COORDINATE ARRAY IN UNSORTED ORDER
C  IND1    -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED X1 LIST
C  IRNK1   -  RANK ARRAY GIVING PARTICLE LOCATION IN THE SORTED X1 LIST
C  X2      -  X2 COORDINATE ARRAY IN UNSORTED ORDER
C  IND2    -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED X2 LIST
C  IRNK2   -  RANK ARRAY GIVING PARTICLE LOCATION IN THE SORTED X2 LIST
C  XMIN1   -  THE LOWER LIMIT OF THE X1 INTERVAL
C  XMAX1   -  THE UPPER LIMIT OF THE X1 INTERVAL
C  XMIN2   -  THE LOWER LIMIT OF THE X2 INTERVAL
C  XMAX2   -  THE UPPER LIMIT OF THE X2 INTERVAL
C  NP      -  THE NUMBER OF PARTICLES IN THE LIST
C  NDIM    -  THE DIMENSION OF THE ARRAYS
C
C  OUTPUT
C  LIST    -  LIST OF PARTICLES TO TEST FOR INTERACTIONS
C  NLIST   -  NUMBER OF PARTICLES IN THE LIST
C
      NLIST=0
      CALL GETBND(X1,IND1,NP,XMIN1,XMAX1,NDIM,ILO1,IUP1)
```

```
      NUM1=IUP1-ILO1+1
      IF(NUM1.LE.0)RETURN
      CALL GETBND(X2,IND2,NP,XMIN2,XMAX2,NDIM,ILO2,IUP2)
      NUM2=IUP2-ILO2+1
      IF(NUM2.LE.0)RETURN
      IF(NUM1.LT.NUM2)THEN
       DO 100 I1=ILO1,IUP1
        N1=IND1(I1)
        IF(IRNK2(N1).LT.ILO2.OR.IRNK2(N1).GT.IUP2)GO TO 100
        NLIST=NLIST+1
        LIST(NLIST)=N1
  100  CONTINUE
      ELSE
       DO 200 I2=ILO2,IUP2
        N2=IND2(I2)
        IF(IRNK1(N2).LT.ILO1.OR.IRNK1(N2).GT.IUP1)GO TO 200
        NLIST=NLIST+1
        LIST(NLIST)=N2
  200  CONTINUE
      ENDIF
      RETURN
      END
      SUBROUTINE MKLST3(X1,IND1,IRNK1,X2,IND2,IRNK2,X3,IND3,IRNK3,
     *XMIN1,XMAX1,XMIN2,XMAX2,XMIN3,XMAX3,NP,NDIM,LIST,NLIST)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      SAVE
      DIMENSION  X1     (NDIM),X2      (NDIM),X3      (NDIM),
     *           IND1   (NDIM),IND2    (NDIM),IND3    (NDIM),
     *           IRNK1  (NDIM),IRNK2   (NDIM),IRNK3   (NDIM),
     *           LIST   (NDIM)
C
C  3D ROUTINE TO MAKE A LIST OF PARTICLES WHICH FALL WITHIN THE
C  RANGE XMIN1<X1<XMAX1 XMIN2<X2<XMAX2 XMIN3<X3<XMAX3.
C  THE PARTICLE LIST HAS PREVIOULSY BEEN SORTED SEPARATELY ON THE X1,
C  X2, AND X3 COORDINATES. THE ARRAYS HAVE NOT BEEN CHANGED;
C  INSTEAD INDEX AND RANK ARRAYS HAVE BEEN CREATED.
C  THE INDEX ARRAY GIVES THE NUMBER OF THE PARTICLE AT EACH LOCATION
C  IN THE SORTED LIST, WHILE THE RANK ARRAY GIVES THE LOCATION OF EACH
C  PARTICLE IN THE SORTED LIST.
C
C  THE ALGORITHM IS TO DO INDIVIDUAL ONE-DIMENSIONAL CHECKS TO
C  FIND ALL THE PARTICLES WITHIN THE X1-LIMITS, THEN THE X2 LIMITS,
C  THEN THE X3 LIMITS. THIS IS DONE WITH CALLS TO GETBND.
C
C  IF ANY LIST HAS ZERO ELEMENTS, THERE ARE NO PARTICLES IN THE RANGE.
C
C  THE ONE-DIMENSIONAL LIST WITH THE SMALLEST
C  NUMBER OF PARTICLES THEN HAS EACH PARTICLE TESTED ON THE LIMITS
C  IN THE ORTHOGONAL DIRECTIONS TO PRODUCE THE FINAL LIST.
C
C  THE CHECK IN THE ORTHOGONAL DIRECTIONS CAN BE DONE BY USE OF
C  THE RANK ARRAY AND THE BOUNDS ON THE INDEX LIST DETERMINED BY
C  GETBND. IF THE RANK OF THE PARTICLE IS NOT WITHIN THE BOUNDS
C  DETERMINED BY GETBND, THE PARTICLE IS EXCLUDED.
C
C  INPUT
C  X1     -  X1 COORDINATE ARRAY IN UNSORTED ORDER
C  IND1   -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED X1 LIST
C  IRNK1  -  RANK ARRAY GIVING PARTICLE LOCATION IN THE SORTED X1 LIST
```

```
C  X2      -  X2 COORDINATE ARRAY IN UNSORTED ORDER
C  IND2    -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED X2 LIST
C  IRNK2   -  RANK ARRAY GIVING PARTICLE LOCATION IN THE SORTED X2 LIST
C  X3      -  X3 COORDINATE ARRAY IN UNSORTED ORDER
C  IND2    -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED X3 LIST
C  IRNK2   -  RANK ARRAY GIVING PARTICLE LOCATION IN THE SORTED X3 LIST
C  XMIN1   -  THE LOWER LIMIT OF THE X1 INTERVAL
C  XMAX1   -  THE UPPER LIMIT OF THE X1 INTERVAL
C  XMIN2   -  THE LOWER LIMIT OF THE X2 INTERVAL
C  XMAX2   -  THE UPPER LIMIT OF THE X2 INTERVAL
C  XMIN3   -  THE LOWER LIMIT OF THE X3 INTERVAL
C  XMAX3   -  THE UPPER LIMIT OF THE X3 INTERVAL
C  NP      -  THE NUMBER OF PARTICLES IN THE LIST
C  NDIM    -  THE DIMENSION OF THE ARRAYS
C
C  OUTPUT
C  LIST    -  LIST OF PARTICLES TO TEST FOR INTERACTIONS
C  NLIST   -  NUMBER OF PARTICLES IN THE LIST
C
      NLIST=0
      CALL GETBND(X1,IND1,NP,XMIN1,XMAX1,NDIM,ILO1,IUP1)
      NUM1=IUP1-ILO1+1
      IF(NUM1.LE.0)RETURN
      CALL GETBND(X2,IND2,NP,XMIN2,XMAX2,NDIM,ILO2,IUP2)
      NUM2=IUP2-ILO2+1
      IF(NUM2.LE.0)RETURN
      CALL GETBND(X3,IND3,NP,XMIN3,XMAX3,NDIM,ILO3,IUP3)
      NUM3=IUP3-ILO3+1
      IF(NUM3.LE.0)RETURN
      IF(NUM1.LE.NUM2.AND.NUM1.LE.NUM3)THEN
       DO 100 I1=ILO1,IUP1
        N1=IND1(I1)
        IF(IRNK2(N1).LT.ILO2.OR.IRNK2(N1).GT.IUP2)GO TO 100
        IF(IRNK3(N1).LT.ILO3.OR.IRNK3(N1).GT.IUP3)GO TO 100
        NLIST=NLIST+1
        LIST(NLIST)=N1
  100  CONTINUE
      ELSEIF(NUM2.LE.NUM1.AND.NUM2.LE.NUM3)THEN
       DO 200 I2=ILO2,IUP2
        N2=IND2(I2)
        IF(IRNK1(N2).LT.ILO1.OR.IRNK1(N2).GT.IUP1)GO TO 200
        IF(IRNK3(N2).LT.ILO3.OR.IRNK3(N2).GT.IUP3)GO TO 200
        NLIST=NLIST+1
        LIST(NLIST)=N2
  200  CONTINUE
      ELSE
       DO 300 I3=ILO3,IUP3
        N3=IND3(I3)
        IF(IRNK1(N3).LT.ILO1.OR.IRNK1(N3).GT.IUP1)GO TO 300
        IF(IRNK2(N3).LT.ILO2.OR.IRNK2(N3).GT.IUP2)GO TO 300
        NLIST=NLIST+1
        LIST(NLIST)=N3
  300  CONTINUE
      ENDIF
      RETURN
      END
      SUBROUTINE GETBND(X,IND,NP,XMIN,XMAX,NDIM,ILO,IUP)
C---5---10---15---20---25---30---35---40---45---50---55---60---65---70--
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      SAVE
      DIMENSION X(NDIM),IND(NDIM)
C
C  FIND THE ELEMENTS IN A SORTED ARRAY X WHOSE VALUES FALL IN THE
C  INTERVAL BETWEEN XMIN AND XMAX. NO ELEMENTS HAVING
C  VALUES EQUAL TO XMIN OR XMAX ARE INCLUDED. SINCE THE ARRAY IS
C  SORTED, THE ELEMENTS CAN BE SPECIFIED BY THE UPPER AND
C  LOWER ELEMENT NUMBERS IN THE RANGE.
C
C     |    X(ILO)   . . . . . . . . . . . . . .   X(IUP)    |
C   XMIN                                                    XMAX       X>
C
C  IT IS ASSUMED THAT THE ARRAY X HAS BEEN SORTED IN INCREASING ORDER,
C  BUT THE ELEMENTS HAVE NOT BEEN MOVED.
C  THE SORTED LIST IS DETERMINED BY THE ARRAY INDX,
C  WHICH POSITIONS THE ORIGINAL UNSORTED X ARRAY ELEMENTS
C  IN THE SORTED LIST. THUS, THE 5TH ELEMENT IN THE SORTED LIST IS
C    X(IND(5))
C
C  INPUT
C  X       -  ARRAY IN UNSORTED ORDER
C  IND     -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED LIST
C  NP      -  THE NUMBER OF PARTICLES IN THE LIST
C  XMIN    -  THE LOWER LIMIT OF THE INTERVAL
C  XMAX    -  THE UPPER LIMIT OF THE INTERVAL
C  NDIM    -  THE DIMENSION OF THE ARRAYS
C
C  OUTPUT
C  ILO     -  THE FIRST ELEMENT IN THE SORTED LIST .GT. XMIN
C  IUP     -  THE LAST ELEMENT IN THE SORTED LIST .LT. XMAX
C
C
C  SEARCH TO FIND THE FIRST ELEMENT .GT. XMIN
      CALL SRCHGT(X,IND,XMIN,1,NP,NDIM,ILO)
C
C  SEARCH TO FIND THE FIRST ELEMENT .GE. XMAX
      CALL SRCHGE(X,IND,XMAX,1,NP,NDIM,IUP)
C  THE PREVIOUS ELEMENT IS THE LAST ONE .LT. XMAX
      IUP=IUP-1
C
      RETURN
      END
      SUBROUTINE SRCHGT(X,IND,XV,IMIN,IMAX,NDIM,I)
C---5---10---15---20---25---30---35---40---45---50---55---60---65---70--
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      SAVE
      DIMENSION X(NDIM),IND(NDIM)
C
C  PERFORM A BINARY SEARCH TO FIND THE ELEMENT NUMBER I
C  OF A SORTED ARRAY FOR WHICH ALL ELEMENTS AT I OR ABOVE ARE
C  GREATER THAN SOME VALUE XV,
C  WHILE ALL ELEMENTS BELOW I ARE LESS THAN OR EQUAL TO THE VALUE.
C
C       X(I-2)     X(I-1)     X(I)     X(I+1)   X(I+2)
C                     XV                              X>
C
C  IT IS ASSUMED THAT THE ARRAY X HAS BEEN SORTED IN INCREASING ORDER,
C  BUT THE ELEMENTS HAVE NOT BEEN MOVED.
C  THE SORTED LIST IS DETERMINED BY THE ARRAY INDX,
```

```
C  WHICH POSITIONS THE ORIGINAL UNSORTED X ARRAY ELEMENTS
C  IN THE SORTED LIST. THUS, THE 5TH ELEMENT IN THE SORTED LIST IS
C    X(IND(5))
C
C  INPUT
C  X       -  ARRAY IN UNSORTED ORDER
C  IND     -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED LIST
C  XV      -  X VALUE TO TEST AGAINST
C  IMIN    -  THE LOWEST NUMBERED POSITION IN THE SORTED LIST TO TEST
C  IMAX    -  THE HIGHEST NUMBERED POSITION IN THE SORTED LIST TO TEST
C  NDIM    -  THE DIMENSION OF THE ARRAYS
C
C  OUTPUT
C  I       -  THE FIRST POSITION IN THE SORTED LIST .GE. XV
C
C
C  FIRST MAKE SURE THE VALUE IS IN THE SEARCH RANGE
      IF(XV.LT.X(IND(IMIN)))THEN
C  ALL VALUES ARE .GT. XV
       I=IMIN
       RETURN
      ELSEIF(XV.GE.X(IND(IMAX)))THEN
C  CAN'T RETURN AN ELEMENT NUMBER HAVING A VALUE .GT. THAN XV
C  SET I=IMAX+1 TO INDICATE THAT ALL ELEMENTS AT I-1=IMAX AND BELOW
C  HAVE VALUES LESS THAN OR EQUAL TO XV
       I=IMAX+1
       RETURN
      ENDIF
C
C  EXECUTE THE BINARY SEARCH
C
      IL=IMIN
      IU=IMAX
  100 IT=(IU+IL)/2
      XTST=X(IND(IT))
C     WRITE(6,'(A,3I4,2F20.10)')' IL,IU,IT,XTST,XV',
C    *IL,IU,IT,XTST,XV
      IF(XTST.LE.XV)THEN
       IL=IT+1
      ELSE
       IU=IT-1
      ENDIF
      IF(IL.LE.IU)GO TO 100
C  RANGE HAD NARROWED TO 1 LOCATION. HOWEVER, THE POINT LAST TESTED
C  COULD BE ABOVE, BELOW, OR ON THE SEARCH POINT. CHECK FOR PROPER CASE
      IF(XTST.LE.XV)THEN
       I=IT+1
      ELSE
       I=IT
      ENDIF
      RETURN
      END
      SUBROUTINE SRCHGE(X,IND,XV,IMIN,IMAX,NDIM,I)
C---5---10---15---20---25---30---35---40---45---50---55---60---65---70--
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      SAVE
      DIMENSION X(NDIM),IND(NDIM)
C
C  PERFORM A BINARY SEARCH TO FIND THE ELEMENT NUMBER I
```

```
C  OF A SORTED ARRAY FOR WHICH ALL ELEMENTS AT I OR ABOVE ARE
C  GREATER THAN OR EQUAL TO SOME VALUE XV,
C  WHILE ALL ELEMENTS BELOW I ARE LESS THAN THE VALUE.
C
C       X(I-2)      X(I-1)     X(I)     X(I+1)    X(I+2)
C                              XV                          X>
C
C  IT IS ASSUMED THAT THE ARRAY X HAS BEEN SORTED IN INCREASING ORDER,
C  BUT THE ELEMENTS HAVE NOT BEEN MOVED.
C  THE SORTED LIST IS DETERMINED BY THE ARRAY INDX,
C  WHICH POSITIONS THE ORIGINAL UNSORTED X ARRAY ELEMENTS
C  IN THE SORTED LIST. THUS, THE 5TH ELEMENT IN THE SORTED LIST IS
C    X(IND(5))
C
C  INPUT
C  X       -  ARRAY IN UNSORTED ORDER
C  IND     -  INDEX ARRAY GIVING THE ELEMENT ORDER IN THE SORTED LIST
C  XV      -  X VALUE TO TEST AGAINST
C  IMIN    -  THE LOWEST NUMBERED POSITION IN THE SORTED LIST TO TEST
C  IMAX    -  THE HIGHEST NUMBERED POSITION IN THE SORTED LIST TO TEST
C  NDIM    -  THE DIMENSION OF THE ARRAYS
C
C  OUTPUT
C  I       -  THE FIRST POSITION IN THE SORTED LIST .GE. XV
C
C
C  FIRST MAKE SURE THE VALUE IS IN THE SEARCH RANGE
      IF(XV.LE.X(IND(IMIN)))THEN
C  ALL VALUES ARE .GE. XV
       I=IMIN
       RETURN
      ELSEIF(XV.GT.X(IND(IMAX)))THEN
C  CAN'T RETURN AN ELEMENT NUMBER HAVING A VALUE .GE. THAN XV
C  SET I=IMAX+1 TO INDICATE THAT ALL ELEMENTS AT I-1=IMAX AND BELOW
C  HAVE VALUES LESS THAN XV
       I=IMAX+1
       RETURN
      ENDIF
C
C  EXECUTE THE BINARY SEARCH
C
      IL=IMIN
      IU=IMAX
  100 IT=(IU+IL)/2
      XTST=X(IND(IT))
C     WRITE(6,'(A,3I4,2F20.10)')' IL,IU,IT,XTST,XV',
C    *IL,IU,IT,XTST,XV
      IF(XTST.LT.XV)THEN
       IL=IT+1
      ELSE
       IU=IT-1
      ENDIF
      IF(IL.LE.IU)GO TO 100
C  RANGE HAD NARROWED TO 1 LOCATION. HOWEVER, THE POINT LAST TESTED
C  COULD BE ABOVE, BELOW, OR ON THE SEARCH POINT. CHECK FOR PROPER CASE
      IF(XTST.LT.XV)THEN
       I=IT+1
      ELSE
       I=IT
```

```
      ENDIF
      RETURN
      END
      SUBROUTINE INDEXX(N,ARRIN,INDX,NDIM)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C  CREATE AN INDEX ARRAY SO THAT ARRIN(INDX(J)) IS IN ASCENDING ORDER
C  COPIED FROM THE NUMERICAL RECIPES BOOK
C  ONLY CHANGE IS TO ADD THE ARRAY DIMENSION NDIM
      DIMENSION ARRIN(NDIM),INDX(NDIM)
      DO 11 J=1,N
        INDX(J)=J
11    CONTINUE
      L=N/2+1
      IR=N
10    CONTINUE
        IF(L.GT.1)THEN
          L=L-1
          INDXT=INDX(L)
          Q=ARRIN(INDXT)
        ELSE
          INDXT=INDX(IR)
          Q=ARRIN(INDXT)
          INDX(IR)=INDX(1)
          IR=IR-1
          IF(IR.EQ.1)THEN
            INDX(1)=INDXT
            RETURN
          ENDIF
        ENDIF
        I=L
        J=L+L
20      IF(J.LE.IR)THEN
          IF(J.LT.IR)THEN
            IF(ARRIN(INDX(J)).LT.ARRIN(INDX(J+1)))J=J+1
          ENDIF
          IF(Q.LT.ARRIN(INDX(J)))THEN
            INDX(I)=INDX(J)
            I=J
            J=J+J
          ELSE
            J=IR+1
          ENDIF
        GO TO 20
        ENDIF
        INDX(I)=INDXT
      GO TO 10
      END
      SUBROUTINE RANK(N,INDX,IRANK,NDIM)
C  CREATE A RANK ARRAY FROM AN INDEX ARRAY
C  COPIED FROM THE NUMERICAL RECIPES BOOK
C  ONLY CHANGE IS TO ADD THE ARRAY DIMENSION NDIM
      DIMENSION INDX(NDIM),IRANK(NDIM)
      DO 11 J=1,N
        IRANK(INDX(J))=J
11    CONTINUE
      RETURN
      END
```

The following is a listing of sample code to call the search algorithm subroutines.

```
      PROGRAM TEST
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NMAX=500)
      DIMENSION  X1     (NMAX),X2      (NMAX),X3      (NMAX),
     *           IND1   (NMAX),IND2    (NMAX),IND3    (NMAX),
     *           IRNK1  (NMAX),IRNK2   (NMAX),IRNK3   (NMAX),
     *           LIST   (NMAX)
      OPEN(2,FILE='tst.dat')
C
C  SQUARE GEMOETRY FOR SEARCH TESTS
      H=1.D0
      I=0
      X3V=0.D0
      DO 620 ILV3=1,5
       X3V=X3V+H
       X2V=0.D0
       DO 620 ILV2=1,5
        X2V=X2V+H
        X1V=0.D0
        DO 610 ILV1=1,5
         X1V=X1V+H
         I=I+1
         IF(I.GT.NMAX)STOP 'NMAX'
         X1(I)=X1V
         X2(I)=X2V
         X3(I)=X3V
         WRITE (2,'(A,I4,3F20.12)')' I,X1,X2,X3',
     *   I,X1(I),X2(I),X3(I)
  610   CONTINUE
  620  CONTINUE
  630 CONTINUE
      NP=I
C
      CALL INDEXX(NP,X1,IND1,NMAX)
      CALL RANK(NP,IND1,IRNK1,NMAX)
      WRITE (2,'(A,3I4,F30.22)')(' I,IND1(I),IRNK1(I),X1(IND1(I))',
     *I,IND1(I),IRNK1(I),X1(IND1(I)),I=1,NP)
      CALL INDEXX(NP,X2,IND2,NMAX)
      CALL RANK(NP,IND2,IRNK2,NMAX)
      WRITE (2,'(A,3I4,F30.22)')(' I,IND2(I),IRNK2(I),X2(IND2(I))',
     *I,IND2(I),IRNK2(I),X2(IND2(I)),I=1,NP)
      CALL INDEXX(NP,X3,IND3,NMAX)
      CALL RANK(NP,IND3,IRNK3,NMAX)
      WRITE (2,'(A,3I4,F30.22)')(' I,IND3(I),IRNK3(I),X3(IND3(I))',
     *I,IND3(I),IRNK3(I),X3(IND3(I)),I=1,NP)
        XMIN1=.9
        XMAX1=1.1
        XMIN2=2.
        XMAX2=4.
        XMIN3=2.
        XMAX3=4.
        CALL MKLST2(X1,IND1,IRNK1,X2,IND2,IRNK2,
     * XMIN1,XMAX1,XMIN2,XMAX2,NP,NMAX,LIST,NLIST)
        WRITE (2,'(A,I6,/,(20I4))')' NLIST,LIST',
```

```
* NLIST,(LIST(ILV),ILV=1,NLIST)
  CALL MKLST3(X1,IND1,IRNK1,X2,IND2,IRNK2,X3,IND3,IRNK3,
* XMIN1,XMAX1,XMIN2,XMAX2,XMIN3,XMAX3,NP,NMAX,LIST,NLIST)
  WRITE (2,'(A,I6,/,(20I4))') NLIST,LIST',
* NLIST,(LIST(ILV),ILV=1,NLIST)
 STOP
 END
```

APPENDIX B

Details of Stability Analysis

B.1 Perturbation Propagation Equations

In order to investigate the stability properties of Eqs. (5.11) and (5.18), we wish to consider the evolution of small perturbations in the independent variables. That is, let positions be perturbed to

$$x^{l,n+1} \rightarrow x^{l,n+1} + \delta x^{l,n+1}, \tag{B.1}$$

and velocities be perturbed to

$$\dot{x}^{l,n+\frac{1}{2}} \rightarrow \dot{x}^{l,n+\frac{1}{2}} + \delta\dot{x}^{l,n+\frac{1}{2}}. \tag{B.2}$$

Each position and velocity, regardless of its spatial and temporal index, will be replaced by a perturbed value as demonstrated in the above two equations. The equations describing the evolution of the perturbations are obtained by substituting the perturbed quantities into Eqs. (5.11) and (5.18), retaining only those terms linear in the perturbations, and subtracting the original equations. All quantities other than the perturbations are assumed constant for purposes of the analysis.

The resulting expressions may be simplified by noting that the position and velocity perturbations of Eqs. (B.1) and (B.2) result in the following perturbations in the equation of state stress,

$$\sigma^{l+1,n} \rightarrow \sigma^{l+1,n} + \delta\sigma^{l+1,n}, \tag{B.3}$$

the viscous stress,

$$Q^{l+\frac{1}{2},n} \rightarrow Q^{l+\frac{1}{2},n} + \delta Q^{l+\frac{1}{2},n}, \tag{B.4}$$

and the kernel function derivative,

$$W'^{\,I+\frac{1}{2},n} \rightarrow W'^{\,I+\frac{1}{2},n} + \delta W'^{\,I+\frac{1}{2},n}, \tag{B.5}$$

where

$$\delta\sigma^{I+1,n} = \frac{\rho_0 K}{2m}\left(\delta x^{I+2,n} - \delta x^{I,n}\right), \tag{B.6}$$

$$\delta Q^{I+\frac{1}{2},n} = \alpha\left(\delta\dot{x}^{\,I+1,n-\frac{1}{2}} - \delta\dot{x}^{\,I,n-\frac{1}{2}}\right), \tag{B.7}$$

and

$$\delta W'^{\,I+\frac{1}{2},n} = W''^{\,I+\frac{1}{2},n}\left(\delta x^{I+1,n} - \delta x^{I,n}\right). \tag{B.8}$$

The last equation follows from a first-order Taylor series expansion of W' and is independent of the form of the kernel function. W'' is the second derivative of W with respect to its argument, and is thus the slope of W'.

Using the above notation, the perturbed form of Eq. (5.11) becomes

$$\dot{x}^{\,I,n+\frac{1}{2}} + \delta\dot{x}^{\,I,n+\frac{1}{2}} - \dot{x}^{\,I,n-\frac{1}{2}} - \delta\dot{x}^{\,I,n-\frac{1}{2}} =$$

$$-\frac{m\Delta t}{\rho^2}\left[\left(\sigma^{I+1,n} + Q^{I+\frac{1}{2},n} + \delta\sigma^{I+1,n} + \delta Q^{I+\frac{1}{2},n}\right)\left(W'^{\,I+\frac{1}{2},n} + \delta W'^{\,I+\frac{1}{2},n}\right)\right.$$

$$\left.-\left(\sigma^{I-1,n} + Q^{I-\frac{1}{2},n} + \delta\sigma^{I-1,n} + \delta Q^{I-\frac{1}{2},n}\right)\left(W'^{\,I-\frac{1}{2},n} + \delta W'^{\,I-\frac{1}{2},n}\right)\right]. \tag{B.9}$$

The linearized form of this equation, after dropping higher-than-linear terms in the perturbations and subtracting the original equation, becomes

$$\delta\dot{x}^{I,n+\frac{1}{2}} - \delta\dot{x}^{I,n-\frac{1}{2}} = -\frac{m\Delta t}{\rho^2}\Big(T^{I+1,n}\delta W'^{I+\frac{1}{2},n} - T^{I-1,n}\,\delta W'^{I-\frac{1}{2},n}$$

$$+ W'^{I+\frac{1}{2},n}\delta\sigma^{I+1,n} - W'^{I-\frac{1}{2},n}\delta\sigma^{I-1,n}$$

$$+ W'^{I+\frac{1}{2},n}\delta Q^{I+\frac{1}{2},n} - W'^{I-\frac{1}{2},n}\delta Q^{I-\frac{1}{2},n}\Big) \tag{B.10}$$

where the total stress, T is given by

$$T^{I+1,n} = \sigma^{I+1,n} + Q^{I+\frac{1}{2},n} \tag{B.11}$$

This result can easily be verified by noting that each cross-product produces surviving terms which consist of the constants in the first term times the perturbations in the second, plus the constants in the second term times the perturbations in the first. Substituting Eq. (B.6), Eq. (B.7), and Eq. (B.8) into Eq. (B.10) yields

$$\delta\dot{x}^{I,n+\frac{1}{2}} = \delta\dot{x}^{I,n-\frac{1}{2}}$$

$$-\frac{m\Delta t}{\rho^2}\Bigg\{ W'''^{I+\frac{1}{2},n}T^{I+1,n}(\delta x^{I+1,n} - \delta x^{I,n}) - W'''^{I-\frac{1}{2},n}T^{I-1,n}\,(\delta x^{I,n} - \delta x^{I-1,n})$$

$$+\frac{\rho_0 K}{2m}\Big[W'^{I+\frac{1}{2},n}(\delta x^{I+2,n} - \delta x^{I,n}) - W'^{I-\frac{1}{2},n}(\delta x^{I,n} - \delta x^{I-2,n})\Big]$$

$$+\alpha\Big[W'^{I+\frac{1}{2},n}\Big(\delta\dot{x}^{I+1,n-\frac{1}{2}} - \delta\dot{x}^{I,n-\frac{1}{2}}\Big)$$

$$-W'^{I-\frac{1}{2},n}\Big(\delta\dot{x}^{I,n-\frac{1}{2}} - \delta\dot{x}^{I-1,n-\frac{1}{2}}\Big)\Big]\Bigg\} \tag{B.12}$$

and

$$\delta x^{I,n+1} = \delta x^{I,n} + \Delta t \delta \dot{x}^{I,n+\frac{1}{2}}. \tag{B.13}$$

These equations describe the propagation of small perturbations in the velocity and position.

B.2 Fourier Decomposition

A Fourier analysis is now performed which involves the assumption of a separation of spatial and temporal variables of the form

$$\delta x^{I,n} = \delta x^n E^I, \tag{B.14}$$

$$\delta \dot{x}^{I,n-\frac{1}{2}} = \delta \dot{x}^{n-\frac{1}{2}} E^I, \tag{B.15}$$

where

$$E^I = (e^{ik\Delta X})^I = \cos(Ik\Delta X) + i \sin(Ik\Delta X) \tag{B.16}$$

describes the spatial variation of the perturbed quantities. In this expression X is the initial, or Lagrangian coordinate, so that ΔX is the initial uniform spacing between particles, and E^I represents E being raised to the power I, rather than an index. The position of particle I at time zero is thus given by

$$x^I = I\Delta X. \tag{B.17}$$

The perturbation is assumed to be periodic with wavenumber, k, which is the circular inverse of the perturbation wavelength, λ, so that

$$k\Delta X = \frac{2\pi}{\lambda}\Delta X. \tag{B.18}$$

Substituting the separated variable solutions into the perturbation propagation equations and dividing by E^I yields

$$\delta\dot{x}^{n+\frac{1}{2}} = \delta\dot{x}^{n-\frac{1}{2}}$$

$$-\frac{m\Delta t}{\rho^2}\Bigg\{ W''^{\,l+\frac{1}{2},n} T^{l+1,n}(E-1) - W''^{\,l-\frac{1}{2},n} T^{l-1,n}\ (1-E^{-1})$$

$$+\frac{\rho_0 K}{2m}\left[W'^{\,l+\frac{1}{2},n}(E^2-1) - W'^{\,l-\frac{1}{2},n}(1-E^{-2})\right]\Bigg\}\,\delta x^n$$

$$-\frac{\alpha m\Delta t}{\rho^2}\left[W'^{\,l+\frac{1}{2},n}(E-1) \quad - W'^{\,l-\frac{1}{2},n}(1-E^{-1})\right]\delta\dot{x}^{n-\frac{1}{2}} \tag{B.19}$$

and

$$\delta x^{n+1} = \delta x^n + \Delta t\,\delta\dot{x}^{n+\frac{1}{2}} \ . \tag{B.20}$$

These expressions can be simplified by considering the case of a uniform initial state, so that

$$W'^{\,l+\frac{1}{2},n} = W'^{\,l-\frac{1}{2},n} = W' \ , \tag{B.21}$$

$$W''^{\,l+\frac{1}{2},n} = W''^{\,l-\frac{1}{2},n} = W'' , \tag{B.22}$$

and

$$T^{l+1,n} = T^{l-1,n} = T \ . \tag{B.23}$$

Equation (B.19) then becomes

$$\delta\dot{x}^{n+\frac{1}{2}} = \delta\dot{x}^{n-\frac{1}{2}}$$

$$-\frac{m\Delta t}{\rho^2}\left[W''T(E+E^{-1}-2) + \frac{\rho_0 KW'}{2m}(E^2+E^{-2}-2)\right]\delta x^n$$

$$-\frac{\alpha m W'\Delta t}{\rho^2}(E+E^{-1}-2)\,\delta\dot{x}^{n-\frac{1}{2}} \tag{B.24}$$

However,

$$E+E^{-1}-2 = 2\left(\cos\frac{2\pi\Delta X}{\lambda}-1\right), \tag{B.25}$$

while

$$E^2+E^{-2}-2 = 2\left(\cos\frac{4\pi\Delta X}{\lambda}-1\right). \tag{B.26}$$

Equation (B.24) thus becomes

$$\delta\dot{x}^{I,n+\frac{1}{2}} = \delta\dot{x}^{I,n-\frac{1}{2}}$$

$$-\frac{2m\Delta t}{\rho^2}\left[W''T\left(\cos\frac{2\pi\Delta X}{\lambda}-1\right)+\frac{\rho_0 KW'}{2m}\left(\cos\frac{4\pi\Delta X}{\lambda}-1\right)\right]\delta x^n$$

$$-\frac{2\alpha m W'\Delta t}{\rho^2}\left(\cos\frac{2\pi\Delta X}{\lambda}-1\right)\delta\dot{x}^{I,n-\frac{1}{2}} \tag{B.27}$$

The first term in brackets multiplying δx^n involves kernel variations at constant stress, while the second term in brackets multiplying δx^n involves stress variations at a constant value of the kernel. The term multiplying $\delta\dot{x}^{I,n-\frac{1}{2}}$ involves the artificial viscosity. The shortest wavelength perturbation which can be resolved by the discrete system is

$$\lambda_{min} = 2\Delta X. \tag{B.28}$$

At this wavelength the term involving stress variations goes to zero, so that the equation of state stress has no effect on the propagation of perturbations at the shortest wavelength.

B.3 Amplification Matrix Eigenvalues

The perturbation propagation equations can be rewritten in the form

$$\delta\dot{x}^{n+\frac{1}{2}} = (1 - r\Delta t)\,\delta\dot{x}^{n-\frac{1}{2}} + s\Delta t\delta x^{n}, \tag{B.29}$$

and

$$-\Delta t\delta\dot{x}^{n+\frac{1}{2}} + \delta x^{n+1} = \delta x^{n}, \tag{B.30}$$

where

$$r = \frac{2\alpha m W'}{\rho^2}\left(\cos\frac{2\pi\Delta X}{\lambda} - 1\right), \tag{B.31}$$

and

$$s = -\frac{2m}{\rho^2}\left[W''T\left(\cos\frac{2\pi\Delta X}{\lambda} - 1\right) + \frac{\rho_0 K W'}{2m}\left(\cos\frac{4\pi\Delta X}{\lambda} - 1\right)\right]. \tag{B.32}$$

Rewriting these equations in matrix format yields

$$\underset{\sim}{L}\underset{\sim}{U}^{n+1} = \underset{\sim}{R}\underset{\sim}{U}^{n}, \tag{B.33}$$

where the vector of new velocities and positions is

$$\underset{\sim}{U}^{n+1} = \begin{bmatrix} \delta\dot{x}^{n+\frac{1}{2}} \\ \delta x^{n+1} \end{bmatrix}, \tag{B.34}$$

the vector of old velocities and positions is

$$\underset{\sim}{U}^n = \begin{bmatrix} \delta\dot{x}^{n-\frac{1}{2}} \\ \delta x^n \end{bmatrix}, \tag{B.35}$$

and

$$\underset{\sim}{L} = \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix}, \tag{B.36}$$

$$\underset{\sim}{R} = \begin{bmatrix} (1 - r\Delta t) & s\Delta t \\ 0 & 1 \end{bmatrix}. \tag{B.37}$$

The stability of this set of equations is determined by the eigenvalues of the amplification matrix $\underset{\sim}{A}$, where

$$\underset{\sim}{U}^{n+1} = \underset{\sim}{A}\underset{\sim}{U}^n. \tag{B.38}$$

Comparison with Eq. (B.33) shows that

$$\underset{\sim}{A} = \underset{\sim}{L}^{-1}\underset{\sim}{R}. \tag{B.39}$$

Determination of the eigenvalues of $\underset{\sim}{A}$ can be simplified by noting that

$$\left|\underset{\sim}{A} - \hat{\lambda}\underset{\sim}{I}\right| = 0 \Leftrightarrow \left|\underset{\sim}{L}\right|\left|\underset{\sim}{A} - \hat{\lambda}\underset{\sim}{I}\right| = \left|\underset{\sim}{R} - \hat{\lambda}\underset{\sim}{L}\right| = 0. \tag{B.40}$$

Thus, an equivalent procedure is to find the eigenvalues of

$$\underset{\sim}{R} - \hat{\lambda}\underset{\sim}{L} = \begin{bmatrix} (1 - r\Delta t - \hat{\lambda}) & s\Delta t \\ \hat{\lambda}\Delta t & 1 - \hat{\lambda} \end{bmatrix} \tag{B.41}$$

The resulting eigenvalue equation is

$$\hat{\lambda}^2 + (r\Delta t - s\Delta t^2 - 2)\hat{\lambda} + 1 - r\Delta t = 0. \tag{B.42}$$

B.4 Instability Condition

The system is unstable, which is to say that values will grow exponentially, if the largest value of $\hat{\lambda}$ which results from the solution of Eq. (B.42) exceeds unity. Determination of stability is simplified by writing the eigenvalue equation in the form

$$\hat{\lambda}^2 - 2B\hat{\lambda} + C = 0 \,, \tag{B.43}$$

where

$$B = 1 + \theta \,, \tag{B.44}$$

$$\theta = \frac{s\Delta t^2 - r\Delta t}{2} \,, \tag{B.45}$$

and

$$C = 1 - r\Delta t \quad . \tag{B.46}$$

The value of the maximum eigenvalue depends on the value of the discriminant D, where

$$D = B^2 - C \,, \tag{B.47}$$

so that the eigenvalues are given by

$$\hat{\lambda} = B \pm D^{1/2}. \tag{B.48}$$

There are several cases to consider based on the sign of D and the magnitude of B, but in the current analysis all cases reduce to the statement that

$$D > 0 \tag{B.49}$$

is a sufficient condition for instability[20]. Combining Eqs. (B.44) to (B.47) yields

$$D = s\Delta t^2 + \theta^2 \,, \tag{B.50}$$

so a sufficient condition for instability is

$$s > 0. \tag{B.51}$$

Note that the viscosity coefficient α is contained only in the constant r, so viscosity cannot stabilize the system if Eq. (B.51) is satisfied.

At λ_{min}, Eqs. (B.28) and (B.32) show that

$$s = \frac{4mW''T}{\rho^2}. \tag{B.52}$$

Therefore, a sufficient condition for unstable growth of the shortest wavelength (twice the particle spacing) is

$$W''T > 0, \tag{B.53}$$

where T is negative in compression and positive in tension. On the other hand, if

$$W''T < 0, \tag{B.54}$$

the system is conditionally stable, which means that the time step must be limited in order to achieve stability.

DISTRIBUTION:

S. Atluri
Center for the Advancement of
Computational Mechanics
School of Civil Engineering
Georgia Institute of Technology
Atlanta, GA 30332

E. B. Becker
Department of Aerospace Eng.
and Engineering Mechanics
The University of Texas at Austin
Austin, TX 78712-1085

T. Belytschko
Department of Civil Engineering
Northwestern University
Evanston, IL 60201

Dave Benson
Department of Applied Mechanics
and Engineering Sciences
University of California San Diego
La Jolla, CA 92093

Naury K. Birnbaum
Century Dynamics Incorporated
7700 Edgewater Dr., Suite 626
Oakland CA 94621

Gregory Clifford
Cray Research Park
655E Lone Oak Drive
Eagan, MN 55121

Peter Cundall
ITASCA Consulting Group, Inc.
1313 Fifth Street, S.E.
Minneapolis, MN 55414

Carl Dyka
Naval Research Lab
Materials Science & Technology
Building 28, Code 6386
4555 Overlook Avenue SW
Washington, DC 20375-5000

R. Douglas Everhart
Battelle
505 King Avenue
Columbus, OH 43201-2693

D. P. Flanagan
Hibbitt, Karlsson & Sorrensen, Inc.
100 Medway St.
Providence, RI 02906

Arlo Fossum
RE/SPEC Inc.
Box 725
Rapid City, SD 57709

Gerry Goudreau
Methods Development Group
Mechanical Engineering Department
Lawrence Livermore National Lab
Livermore, CA 94550

Jerome B. Johnson
USACRREL
Building 4070
Ft. Wainwright, AK 99703

Sheldon Jones
Kaman Sciences
P.O. Box 7463
Colorado Springs, CO 80933-7463

David W. Keck
CONVEX Computer Corporation
P.O. Box 833851 M.S. MAR
Richardson, TX 75083-3851

Raymond D. Krieg
Engineering Science and
Mechanics
301 Perkins Hall
University of Tennessee
Knoxville, TN 37996-2030

Hans Mair, Code R14
Naval Surface Warfare Center
10901 New Hampsure Ave.
Silver Spring, MD 20903-5000

Loren K. Miller
Goodyear Technical Center
P.O. Box 3531
Akron, OH 44309-3531

S. Nemat-Nasser
Department of Applied Mechanics
and Engineering Sciences
University of California San Diego
La Jolla, CA 92093

J. T. Oden
Department of Aerospace Eng.
and Engineering Mechanics
The University of Texas at Austin
Austin, TX 78712-1085

Allan B. Pifko
Grumman Corporate Technology
Bethpage, NY 11714-3580

Mark Rashid
Department of Civil &
Environmental Engineering
University of California
Davis, CA 95616-5294

J. S. (Gus) Rice
Caterpillar Inc. Technical Center
Division 927
P.O. Box 1875
Peoria, IL 61656-1875

Steven F. Rieco
POD Associates, Inc.
2309 Renard Pl, Suite 201
Albuquerque, NM 87106

R. G. Sauvé
Mechanical Research Department
Ontario Hydro
700 University Avenue C26
Toronto, Ontario M5G 1X6
Canada

L. M. Taylor
Hibbitt, Karlsson & Sorrensen,
Inc.
100 Medway St.
Providence, RI 02906

2 Air Force Institute of Technology
Dept. of Mathematics & Statistics
2950 P Street
Attn: Michael Stoecker
Dave Fulk
Wright-Patterson Air Force Base
Dayton, OH 45433-7765

Gordon Johnson
Alliant Tech Systems , Inc.
(MN 11-2925)
600 2nd St. NE
Hopkins, MN 55343

Marv Alme
Alme and Associates
2 Stevens Forest Prof. Center
9650 Santiago Road
Columbia, MD 21045

Ted Carney
Applied Research Associates
4300 San Mateo Blvd. NE
Suite A-220
Albuquerque, NM 87110

3 Batelle
505 King Avenue
Attn: Mike Fisher
Doug Everhart
Chuck Hargraves
Columbus, OH 43201-2693

2 Cray Research, Inc.
6565 Americas Parkway, NE
Suite 830
Attn: Phil Campbell
Dave Shirley
Albuquerque, NM 87110

Lou Baker
Dagonet Software
2904 La Veta Dr. NE
Albuquerque, NM 87110-3110

Lawrence Livermore National Lab
Bill Hoover
MS L-794
P. O. Box 808
Livermore, CA 94550

Darrell L. Hicks
Michigan Tech. University
Math Department
Houghton, MI 49931

Joe Monaghan
Monash University
Mathematics Department
Clayton, Vic. 3168
Australia

Larry Libersky
NM Institute of Mining and Tech.
Center for Explosives Tech.
Research
Socorro, NM 87801

5 Phillips Laboratory
PL/WSSD
Kirtland Air Force Base
Attn: Charles Luehr
David Medina
James Ninter
Brad Smith
Firooz Allahdadi
Albuquerque, NM 87117

Willy Benz
University of Arizona
Steward Observatory
Tucson, AZ 85721

Rita Smith
University of New Mexico
Dept. of Chemical/Nuclear Engin.
Farris Engineering Center
Albuquerque, NM 87131-1341

12 Los Alamos National Laboratory
Los Alamos, NM 87545
Attn:
J. P. Hill, WX-11, MS C931
B. L. Holian, T-12, MS B268
D. A. Rabern, MEE-4, MS G787
P. S. Follansbee, MS G756
D. Mandell, X-3, MS F663
R. F. Davidson, N-6, MS K557
N. L. Johnson, T-3, MS B261
J. K. Dienes, N-6, MS K557
C. A. Anderson, MS J576
M. W. Lewis, MEE-4, MS G787
C. Wingate, MS F645
B. Stellingwerf, MS F645

Sandia Internal:

MS1187 1271 George Allshouse
MS1184 1239 Frank Dempsey
MS0321 1400 Ed Barsis
MS0441 1425 Johnny Biffle
10 MS0441 1425 Stephen Attaway
MS0441 1425 Mark Blanford

	MS0441	1425	Marilyn Smith
	MS0820	1433	M. E. Kipp
	MS0439	1434	David Martinez
	MS0841	1500	D. J. McCloskey
	MS0836	1501	C. W. Peterson
	MS0827	1502	P. J. Hommert
	MS0827	1511	J. S. Rottler
	MS0827	1511	Jim Schutt
	MS0834	1512	A. C. Ratzel
	MS0835	1513	R. D. Skocypec
	MS0832	1551	W. P. Wolfe
	MS0833	1552	C. E. Hailey
	MS0826	1553	W. L. Hermina
	MS0825	1554	W. H. Rutledge
15	MS0443	1561	H. S. Morgan & Staff
14	MS0437	1562	R. K. Thomas & Staff
25	MS0437	1562	J. W. Swegle
	MS0336	1707	Kim Mahin
	MS0515	2561	S. T. Montgomery
	MS0660	2861	Randall Lober
	MS0457	5600	Dennis Hayes
	MS0574	5941	John Schamaun
	MS0724	6000	Dan Hartley
	MS0751	6117	Dale Preece
	MS1325	6313	Joseph Jung
	MS1143	6500	James Rice
	MS1145	6514	Jim Fisk
	MS1145	6514	Joel Miller
	MS9214	8117	William Mason
	MS9401	8702	Bill Robinson
	MS9043	8743	George Johnson
	MS9042	8741	Juanita Benson
	MS9042	8741	Michael Chiesa
	MS9042	8742	Jay Dike
	MS9042	8742	Paul Jin
	MS9042	8742	Bruce Kistler
	MS9042	8742	Khanh Trinh
	MS9043	8743	Melvin Callabresi
	MS9043	8743	Douglas Bammann
	MS9043	8743	Lee Bertram
	MS9043	8743	Mark Horstemeyer
	MS9043	8743	James Lathrop
	MS9043	8743	Arthur Ortega
	MS9043	8743	Vincent Prantil
	MS9043	8745	William Winters
	MS0974	9421	Jeanne Ramage
5	MS0899	7141	Technical Library
	MS0619	7151	Tech Publications
10	MS01007613		Document Processing for DOE/OSTI
	MS90188523		Central Tech Files

DATE

FILMED

7/15/94

END