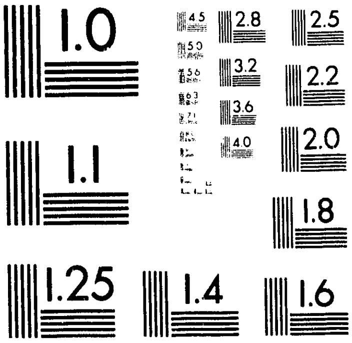

1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

# COUPLING OF SMOOTH PARTICLE HYDRODYNAMICS WITH PRONTO[1]

**S. W. Attaway, M. W. Heinstein, F. J. Mello, J. W. Swegle**
Sandia National Laboratories
Albuquerque, New Mexico

## ABSTRACT

A gridless numerical technique called smooth particle hydrodynamics (SPH) has been coupled to the transient dynamics finite element code, PRONTO. In this paper, a new weighted residual derivation for the SPH method will be presented, and the methods used to embed SPH within PRONTO will be outlined. Example SPH-PRONTO calculations will also be presented.

One major difficulty associated with the Lagrangian finite element method is modeling materials with no shear strength; for example, gases, fluids and explosive bi-products. Typically these materials can be modeled for only a short time with a Lagrangian finite element code. Large distortions cause tangling of the mesh, which will eventually lead to numerical difficulties such as negative element area or "bow tie" elements. Remeshing will allow the problem to continue for a short while, but the large distortions can prevent a complete analysis.

Smooth particle hydrodynamics is a gridless Lagrangian technique. Requiring no mesh, SPH has the potential to model material fracture, large shear flows, and penetration. SPH computes the strain rate and the stress divergence based on the nearest neighbors of a particle, which are determined using an efficient particle sorting technique.

Embedding the SPH method within PRONTO allows part of the problem to be modeled with quadrilateral finite elements while other parts are modeled with the gridless SPH method. SPH elements are coupled to the quadrilateral elements through a contact like algorithm.

## A WEIGHTED RESIDUAL DERIVATION FOR SMOOTH PARTICLE HYDRODYNAMICS

Here, we derive the equations for smooth 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.

SPH was first applied by Lucy [1] to astrophysical problems and was extended by Gingold [2] and

[1] This work performed at Sandia National Laboratories supported by the U. S. Department of Energy under contract DE-AC0476DP00789.

Monaghan [3]. Cloutman [4] has shown that SPH could be used to model hypervelocity impacts. Libersky and Petschek [5] have shown that SPH can be used to model materials with strength.

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 [7].

## Displacement and Strain Approximation

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

$$\underline{u}(\underline{r}) = \int \underline{u}(\underline{r}')\,\delta(\underline{r}-\underline{r}')\,d\underline{r}' \tag{1}$$

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

$$\underline{u}(\underline{r}) \approx \int_{\Omega} \underline{u}(\underline{r}')\,W(\underline{r}-\underline{r}', h)\,d\underline{r}' \tag{2}$$

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

$$\lim_{h \to 0} W(\underline{r}-\underline{r}', h) = \delta(\underline{r}-\underline{r}') \tag{3}$$

and that

$$\int_{\Omega} W(\underline{r}, h)\,dV = 1 \tag{4}$$

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

$$W(\underline{r}, h) = 0 \qquad \text{for} \quad |\underline{r}| \geq 2h \tag{5}$$

## Polynomial Kernel Function

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

$$W(\underline{r}, h) = \begin{bmatrix} \dfrac{1.5}{0.7\pi h^2}\left(\dfrac{2}{3} - z + \dfrac{z^2}{2}\right) & (0 < z < 1) \\ \dfrac{0.25}{0.7\pi h^2}(2-z)^3 & (1 < z < 2) \\ 0 & (2 < z) \end{bmatrix} \tag{6}$$

where $z = \frac{|\underline{r}|}{h}$. In the following derivation, the short hand notation: $W^I = W(\underline{r}-\underline{r}^J, h)$, $W^{IJ} = W(\underline{r}^{IJ}, h) = W(\underline{r}^I-\underline{r}^J, h)$, 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 (6) was selected because it is used in classical SPH methods and it satisfies the constraints expressed in Equations (3) and (4). Other forms of the kernel function using an exponential form have been suggested. (see [7], [8]) The displacements, velocity and acceleration are interpolated using the kernel basis function $W$ in Equation(6).

## Volume Weighted Sum Integral Approximation

The volume integral in Equation (2) is approximated by a volume weighted sum at particle $I$:

$$\underline{u}(\underline{r}^I) = \sum_{J \in h_i} \underline{u}(\underline{r}^J)\;W(\underline{r}^{IJ}, h)\;\Delta V^J \tag{7}$$

where $\Delta V^J$ 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, $\rho$, at the particle:

$$\Delta V^J = \frac{m^J}{\rho^J} \tag{8}$$

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:

$$\underline{\dot{u}}^I = \sum_{J \in h_J} \underline{\dot{u}}^J \; W^{IJ} \; \Delta V^J \tag{9}$$

## 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(2).

$$\nabla \underline{\dot{u}} \approx \int \nabla \underline{\dot{u}}(\underline{r}') \, W(\underline{r} - \underline{r}', h) \, dV' \tag{10}$$

Green's theorem and Equation (5) can be used to transform Equation (10) to

$$\nabla \underline{\dot{u}}(\underline{r}) \approx -\int_{\Omega} \underline{\dot{u}}(\underline{r}') \, \nabla W(\underline{r} - \underline{r}', h) \, dV' \tag{11}$$

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

$$\nabla \underline{\dot{u}}^I = \sum_{J \in h_I} (\underline{\dot{u}}^J - \underline{\dot{u}}^I) \;\; \nabla W^{IJ} \;\; \Delta V^J \tag{12}$$

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

## 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, $\delta \underline{u}$.

$$\int_{\Omega} \delta \underline{u} \, (\nabla \bullet \underset{\sim}{\sigma} + \rho \underline{\ddot{u}} + \underline{f}_b) \, dV = 0 \tag{13}$$

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

$$\int_{\Omega} \delta \underline{u} \, (\nabla \bullet \underset{\sim}{\sigma} + \rho \underline{\ddot{u}}) \, dV = 0. \tag{14}$$

Integration by parts of Equation (14) gives:

$$\int_{\Omega} \delta \underline{u} \;\; \rho \;\; \underline{\ddot{u}} \;\; dV - \int_{\Omega} \underset{\sim}{\sigma} \;\; \bullet \nabla \delta \underline{u} \;\; dV + \int_{\partial\Omega} \bar{\underline{t}} \;\; \delta \underline{u} \;\; dS = 0 \tag{15}$$

where $\bar{\underline{t}} = \underset{\sim}{\sigma} \cdot \underline{n}$ is the traction on surface with normal $\underline{n}$. To obtain a weighted residual statement for the discrete system, substitute Equation (7) into Equation (15).

$$R_u = \int_{\Omega} \rho \underline{\ddot{u}} \, (\Sigma \delta \underline{u}^I \;\; W^I \;\; \Delta V^I) \, dV \tag{16}$$

$$-\int_{\Omega} \underset{\sim}{\sigma} \bullet \, (\Sigma \delta u^I \;\; \nabla W^I \;\; \Delta V^I) \, dV + \int_{\partial\Omega} \underline{t} \, (\Sigma \delta \underline{u}^I \;\; W^I \;\; \Delta V^I) \, dS$$

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

$$\frac{\partial R_u}{\partial \delta \underline{u}} = 0 \tag{17}$$

For node $I$, Equation (17) gives:

$$\int \rho \; \ddot{\underline{u}} \; W^I \; \Delta V^I \; dV - \int \underset{\sim}{\sigma} \; \nabla W^I \; \Delta V^I \; dV + \int \underline{t} \; W^I \; \Delta V^{\mathrm{I}} \; dS = 0 \tag{18}$$

Note that the integral in Equation (18) is over the total volume; however, $W^{\mathrm{I}} = 0$ and $\frac{\partial W^I}{\partial \underline{r}} = 0$ for $|\underline{r}| > h$, so that Equation (18) can be approximated using a volume weighted sum

$$\int_{\Omega} \rho \; \ddot{\underline{u}} \; W^I \; \Delta V^I \; dV \approx \sum_J \rho^J \; \ddot{\underline{u}}^J W^{IJ} \; \Delta V^I \; \Delta V^J \tag{19}$$

$$\int_{\Omega} \underset{\sim}{\sigma} \; \frac{\partial W^I}{\partial \underline{r}} \; \Delta V^I \; dV \approx \sum_J \underset{\sim}{\sigma}^J \; \nabla W^{IJ} \; \Delta V^I \; \Delta V^J \tag{20}$$

where $\Delta V^J$ is the volume contribution of node $J$:

$$\Delta V^J = \frac{m^J}{\rho^J} \tag{21}$$

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

$$\int_{\partial\Omega} \underline{t} \; W^I \; \Delta V^I \;\; dS \approx \sum_J \underline{t} \; W(\underline{r}^J - \underline{r}^B) \;\; \Delta V^I \; \Delta S^J \tag{22}$$

where $\underline{r}^B$ is the shortest distance from the boundary to the particle $I$, and $\Delta S^J$ is the surface intercept along the prescribed boundary. Campbell [7] computes $\Delta S$ based on the intersection of a sphere with a plane as:

$$\Delta S^J = C(\alpha) \; (h^2 - |\underline{r}^J - \underline{r}^B|^2)^{\frac{\alpha - 1}{2}} \tag{23}$$

where $\alpha = 1,2,3$ is the dimension of the problem and $C(\alpha) = (1,2,\pi)$. Substituting Equations (19), (20) and (22) into Equation (18) gives a total of $N$ equations with each node $I$ having an equation of the form:

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

$$- \sum_{J \in h_I} \underline{t} \; W^{IJ} \; \Delta V^I \; \Delta S^J \tag{24}$$

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 latter). Equation (24) can be assembled to give a mass matrix and force vector:

$$\boldsymbol{M} \; \ddot{\boldsymbol{u}} = \boldsymbol{F}^{\mathrm{ext}} - \boldsymbol{F}^{\mathrm{int}} \tag{25}$$

For dynamic problems, Equation (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 (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:

$$M^{IJ} = \rho^J \; W^{IJ} \; \Delta V^J \; \Delta V^J \qquad (26)$$

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:

$$m^I \frac{d\underline{v}}{dt} = \underline{F}^I = \sum_{J \in h_I} \underset{\sim}{\sigma}^J \; \nabla W^{IJ} \; \Delta V^I \; \Delta V^J \qquad (27)$$

Equation (27) can be made symmetric by adding the constant $\sigma^I$ to the sum at node I:

$$\underline{F}^I = \sum_{J \in h_I}^{K} (\underset{\sim}{\sigma}^J + \underset{\sim}{\sigma}^I) \;\; \nabla W^{IJ} \; \Delta V^I \; \Delta V^J \qquad (28)$$

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 $r^{IJ} > h$.

## 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 for SPH, the derivation suggest several areas for further work.

The accuracy of the approximating function will depended 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 proven to be stable and convergent. Swegle et. al [11] and Hicks et. al [12] have investigated the stability of the SPH method.

## COUPLING OF PARTICLE METHODS WITH LAGRANGIAN METHODS

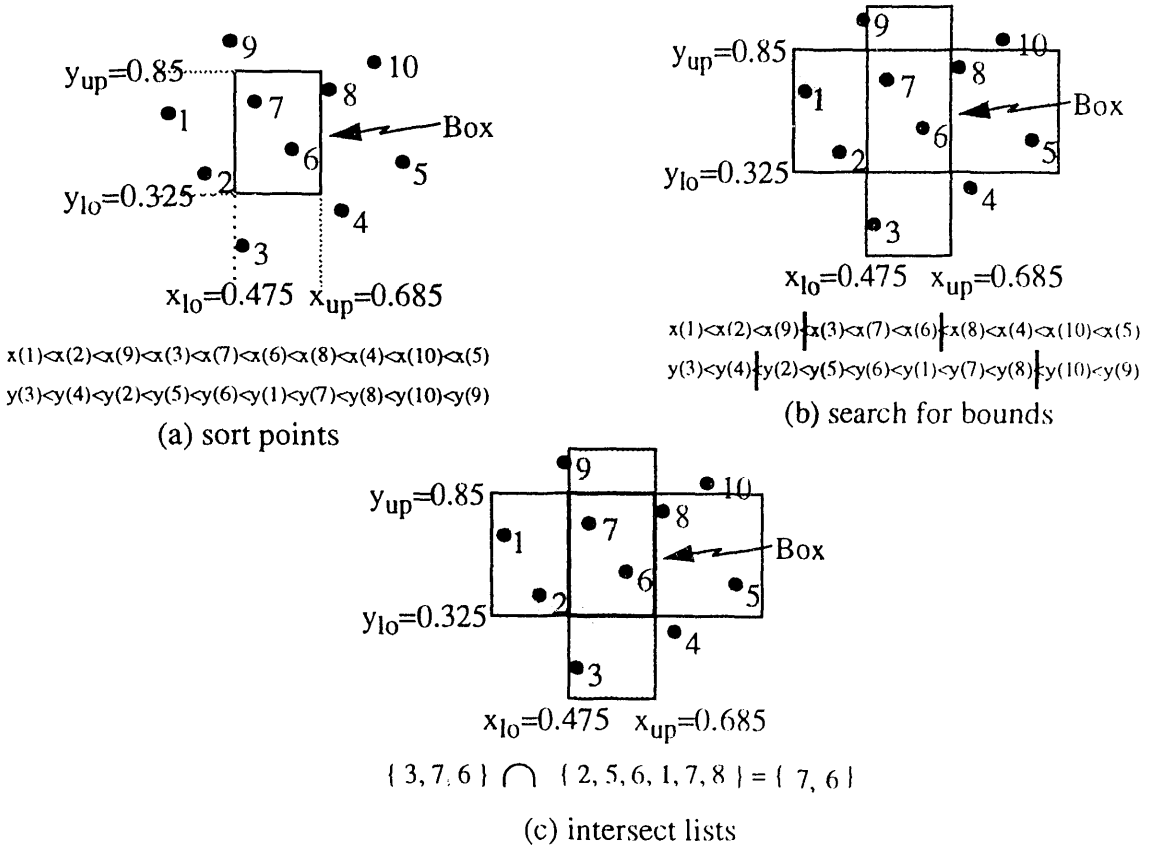
In the above section we presented a derivation of the SPH method that shows that it can be reduced to a Lagrangian weighted residual method. The SPH method can be easily embedded within existing finite element code architecture, if the particles are viewed as elements whose connectivity must be determined 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.

**Particle Searching Technique**. Determining the subset of points (particles) that are contained within the local support defined by $h$ can be one of the most time consuming parts of a SPH calculation. Two phases of the coupling problem require a particle search: i) a search is required to determine the list of particles that interact with a given particle element, and ii) a search is required to determine which particles interact with the boundary of the finite elements.

To determine which of a given set of points lie inside a box, an ordered list of points was constructed by sorting the points on each rectangular coordinate value. This list was then searched for the points that lie within the box. The search algorithm is economical in its use of memory. In 3D, for example, the algorithm requires only 7N memory locations, where N is the number of points. Here, we review the point-in-box search algorithm developed by Swegle [13]. Briefly, the algorithm consists of individual one-dimensional sorts of the points using each coordinate value as the search key, followed by binary searches of each sorted list to find the points at the edge of the search region (box). This produces separate sets of points whose positions fall within the bounds of the box for each coordinate direction. Finally the three lists (two lists for a 2D problem) are intersected to obtain the points inside the box. Figure 1 shows a schematic of the three steps in the algorithm. A description of each step is given below, followed by a detailed example.



(a) sort points

(b) search for bounds

(c) intersect lists

**FIGURE 1. Schematic of Swegle Search Algorithm**

**Sort.** The sorting step constructs an ordered list of the points for each coordinate direction. The result of this sort is an index vector for each coordinate, $\{I_x, I_y, I_z\}$, that contains the point ID's in order of increasing coordinate value. One additional set of vectors, $\{R_x, R_y, R_z\}$, called the rank vectors is also constructed. It gives the location of each point in the index vector. It can be easily constructed by looping through the index vector. For example, suppose point $j$ is stored at position $i$ in the index vector.'Then the rank vector would store the pointer *i* at its position *j*. The rank vector is required to avoid searching the index vector for a given point. The memory requirements for this step is 2*N*ND, where N is the number of points and ND is the spatial dimension of the point set.

**Search.** The second step is to form three lists (one for each coordinate) which contain those points that are within the minimum and maximum bounds of the box. Each list is formed using two binary searches on the index vector: one to find the pointer corresponding to the first point inside the box, and the other to find the pointer corresponding to the last point within the box. Figure 2 shows one step in the $O(\log_2 N)$ binary search, where the target $x_{target}$ corresponds to one of the bounds $x_{min}$ or $x_{max}$ of the box.

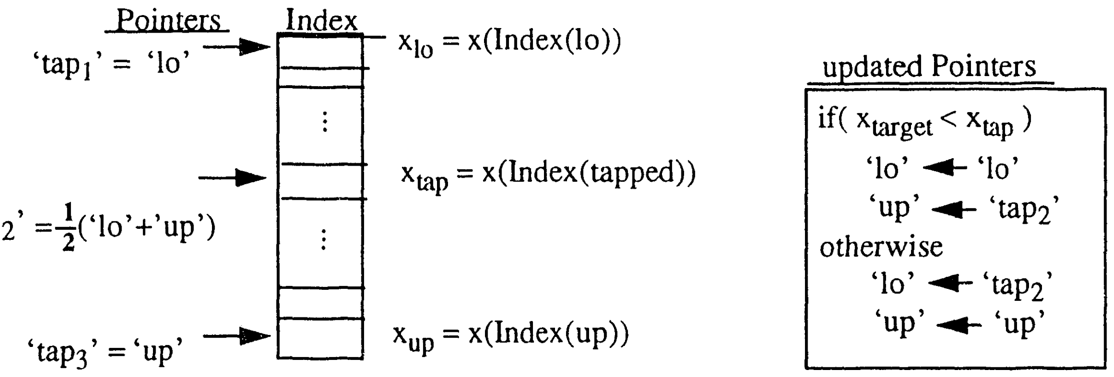


**FIGURE 2. Classical binary search step to find a pointer into index array**

The binary search continues until 'lo' + 1 = 'up' and $x_{lo} < x_{target} < x_{up}$. Upon completing the search, the pointer into the index vector is then $(i_x)_{min}$ = 'up' for the target $x_{target} = x_{min}$, and $(i_x)_{max}$ = 'lo' for the target $x_{target} = x_{max}$.

**Intersection**. Finally, the lists are intersected to find the slave nodes in the box for each coordinate simultaneously. To accomplish this, each of the slave nodes in one list is selected and then checked to see if its rank is between the lower and upper pointer in the other two coordinates. For computational efficiency the shortest list of slave nodes is selected, which can be determined by selecting the smallest of $[(i_w)_{max} - (i_w)_{min} + 1]$, w = x, y, or z. Suppose, for example, that the list for the y-coordinate contains the smallest number of points. Then the points in this list $i = I_y((i_y)_{min})$, $I_y((i_y)_{min} + 1)$, ..., $I_y((i_y)_{max})$ are in the box if $(i_x)_{min} \le R_x(i) \le (i_x)_{max}$ and $(i_z)_{min} \le R_z(i) \le (i_z)_{max}$.

## Summary of Point in Box Search

The great advantage of this algorithm is that it is nearly independent of the geometry of the point set and is very economical in its use of memory. In 3D, for example, the search algorithm requires only (7N) memory locations, where N is the number of points. Davis, *et. al* [14] have optimized the point-in-box search for the vector architecture on the CRAY Y-MP.

## CONTACT COUPLING OF PARTICLE METHODS WITH 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 pushback. The point of contact, amount of penetration, and the direction of pushback 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.

## Location phase

Only an outline of the location phase will be presented here, the reader is referred to Heinstein, et al [13] 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 master surface are 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 mater surface and a global search for all SPH nodes inside this capture box is preformed. 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 is 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 pushback. 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.

## 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:

$$\hat{a} = \frac{f}{m} \qquad \text{(finite element method)} \tag{29}$$

$$\hat{a} = \sum m^J \left(\frac{\underset{\sim}{\sigma}^J}{\rho^I \rho^J}\right) \frac{\partial W}{\partial \underset{\sim}{x}^J} \text{ (smooth particle method)} \tag{30}$$

$$\hat{v} = v + \Delta t \ \hat{a} \tag{31}$$

$$\hat{x} = x + \Delta t \ \hat{v} \tag{32}$$

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

$$\delta = max(\vec{n} \cdot (\bar{x} - \hat{x}), 0) \tag{33}$$

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:

$$a_n = \frac{\delta}{\Delta t^2} \tag{34}$$

or

$$f_s = \frac{\delta\ m}{\Delta t^2}. \tag{35}$$

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)

$$F_i = \sum_s \left(\frac{1}{2} - \xi\right) f_s \text{ and} \tag{36}$$

$$F_{i+1} = \sum_s \left(\frac{1}{2} + \xi\right) f_s. \tag{37}$$

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

$$a_i = \frac{\sum F_{is}}{m_I} \tag{38}$$

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:

$$a_{ps} = \left(\frac{1}{2} - \xi\right) a_{n1} + \left(\frac{1}{2} + \xi\right) a_{n2} \tag{39}$$

This leads to a corrected penalty force:

$$f_s = \frac{\delta_s\ m_s}{\Delta t^2} - a_{ps} m_s. \tag{40}$$

And a new master node acceleration

$$m_i a_i = \sum_s (f_{Is} - a_{ps} m_s). \tag{41}$$

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

$$\sum_s f_{Is} = \left(m_I + \sum_s m_{Is}\right) a_{nI} \tag{42}$$

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.

$$a_{ns} = a_{ps} - \frac{f_p}{m_s} \tag{43}$$

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

$$a = \hat{a} + a_n \tag{44}$$

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.

## EXAMPLE 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.

### Two bars impacting

This example considers two one inch square copper bars impacting at 1000 in/sec. 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 3 shows a plot of the pressure that results from the impact.

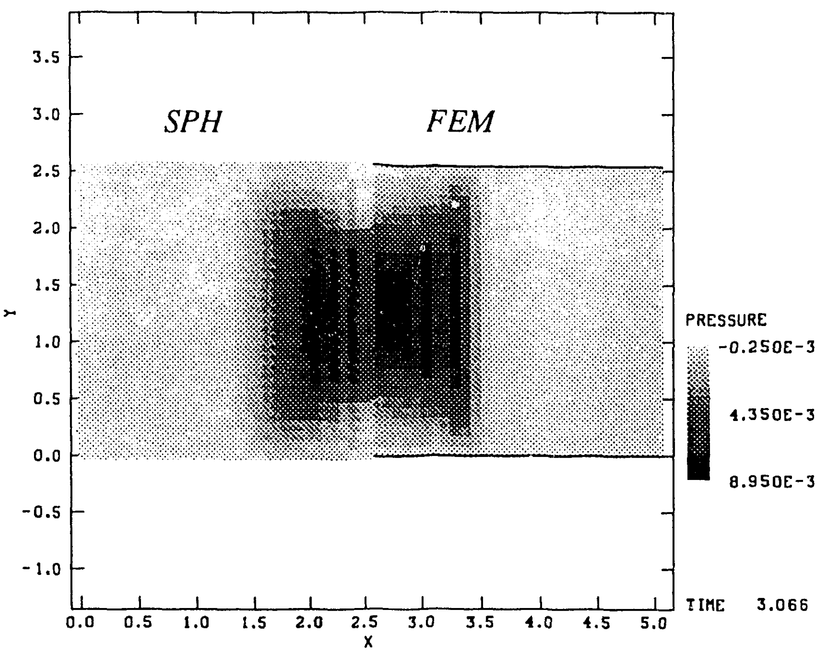


**FIGURE 3. Two copper bars impacting.**

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 method 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.

### Boat Impacting water

This example shows some of the advantages of combining the SPH and FEM method. Figure [4] shows a plane strain model of a boat like object impacting 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 for 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 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).

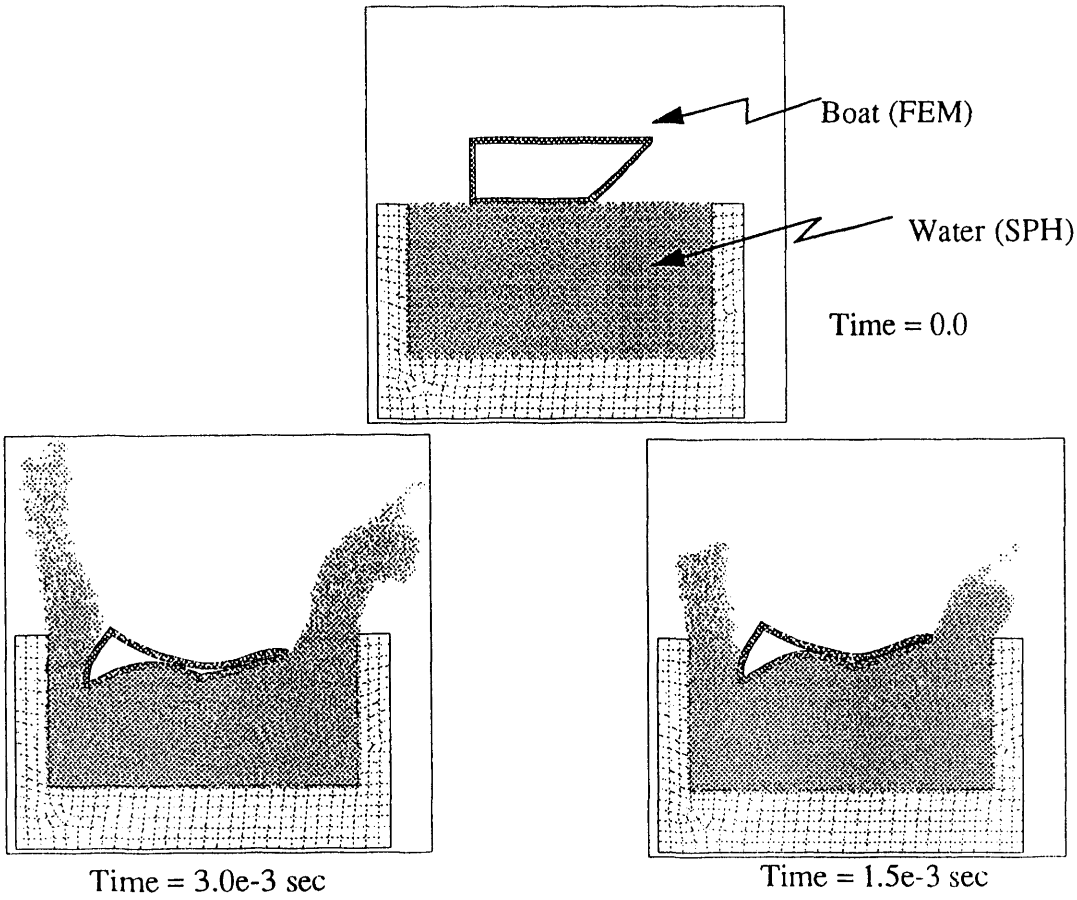


**FIGURE 4. High speed impact of water by 'boat' like object. Impact speed = 2000 in/sec.**

## SUMMARY

The ability to couple particle methods and finite element method will allow fluid structure interaction problems to be solved efficiently. Here we have demonstrated the ability to couple smooth particle hydrodynamics to the transient finite element code PRONTO. This coupling actually 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 element particles.

## REFERENCES

1. L. B. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis," *The Astronomical*

*Journal,* Vol. 82, No. 12, p.p. 1013, Dec. 1977.

2. R. A. Gingold, and J. J. Monaghan, "Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics," *Journal of Computational Physics*, Vol. 46, p.p. 429-453, 1982.

3. J. J. Monaghan, "Why Particle Methods Work," SIAM J. Sci. Comput., Vol. 3, No. 4, Dec. 1982.

4. L. D. Cloutman, "SPH Simulations of Hypervelocity Impacts" Lawrence Livermore National Laboratory, UCRL-ID-105520, 1991.

5. L. D. Libersky and A. G. Petschek, "Smooth Particle Hydrodynamics With Strength of Materials," New Mexico Institute of Mining and Technology, Socorro, New Mexico.

6. J. J. Monaghan, "An Introduction to SPH," *Computer Physics Communications*, Vol. 48, p.p. 89-96, 1988.

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

8. W. Benz, "Smooth Particle Hydrodynamics: A Review," Harvard-Smithsonian Center for Astrophysics, Preprint Series No. 2884, May 1989.

9. G. S. Grest, B. Dunweg, and K. Kremer, "Vectorized Link Cell FORTRAN Code for Molecular Dynamics Simulations for A Large Number of Particles," *Computer Physics Communications*, Vol. 55, p.p. 269-285, 1989.

10. L. M. Taylor and D. S. Preece, "DMC - A Rigid Body Motion Code for Determining the Interaction of Multiple Spherical Particles," SAND88-3482, June 1989.

11. J. W. Swegle, D. L. Hicks, S. W. Attaway, "Stability Analysis of Smoothed Particle Hydrodynamics," submitted to Journal of Computational Physics, 1993.

12. D. L. Hicks, J. W. Swegle, S. W. Attaway, "Smothed Particle Hydrodynamics, Discrete - Numerical Instabilities and Conservative Smoothing," submitted to Mathematics of Computation, 1993.

13. 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.

14. M. E. Davis, M. W. Heinstein, S. W. Attaway, and J. W. Swegle, "Optimizing the Point-in-Box Search Algorithm for the Cray Y-MP Supercomptuer", SAND93-0933, in preparation.

## DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DATE

FILMED

11 / 9 / 93

END