

## An improved generalized particle algorithm that includes boundaries and interfaces

Gordon R. Johnson<sup>1,2,\*†</sup>, Stephen R. Beissel<sup>1</sup> and Robert A. Stryk<sup>2</sup>

<sup>1</sup>*Network Computing Services Inc., Minneapolis, MN 55415, U.S.A.*

<sup>2</sup>*Alliant Techsystems Inc., Hopkins, MN 55343, U.S.A.*

### SUMMARY

This paper presents an improved generalized particle algorithm (GPA), as well as new boundary and interface algorithms for particle interaction with finite elements and other particles (of different materials). The improved GPA uses a local co-ordinate system that is aligned with the boundaries and/or interfaces for the determination of the strain rates and forces. This enables the boundary and interface algorithms to be applied in a straightforward manner. It also provides an invariant solution that is independent of the orientation of the global co-ordinate system. Several examples are presented to illustrate the effects of the various algorithms. Copyright © 2001 John Wiley & Sons, Ltd.

### 1. INTRODUCTION

A generalized particle algorithm (GPA) was recently presented for high velocity impact and other dynamics problems [1]. Included was a discussion of interface problems and potential solutions for the GPA. This paper is an extension of the previous GPA paper [1] inasmuch as it provides an improved GPA, as well as new boundary and interface algorithms for particle interaction with finite elements and other particles (of different materials).

An appeal of meshless particle methods is that they are Lagrangian techniques that can have variable nodal connectivity for severe distortions. The Lagrangian feature is desirable because it allows the grid to be embedded in the material; this reduces some of the material interface problems and material history problems associated with Eulerian codes. Furthermore, the ability to handle severe distortions allows the particle techniques to be applied to problems that historically have been reserved for Eulerian approaches. The Lagrangian formulation also allows the particle algorithms to be linked to Lagrangian finite element algorithms in a straightforward manner.

---

\*Correspondence to: Gordon R. Johnson, Alliant Techsystems, Inc., Mail Station MN11-1614, 600 Second Street NE, Hopkins, MN 55343, U.S.A.

†E-mail: Gordon.Johnson@ATK.COM

Contract/grant sponsor: U.S. Army ERDC; contract/grant number: DACA 39-98-K-0015

It is not possible to review all of the previous work concerning particle methods. A 1996 volume of *Computer Methods in Applied Mechanics and Engineering* is dedicated to advances in meshless methods [2] and it provides a comprehensive collection of a variety of particle (or meshless) algorithms. A more recent volume of *Computational Mechanics* [3] provides another collection of meshless particle papers. Some of the key historical contributions, most closely associated with the work presented in this paper, are as follows:

In 1977 Lucy [4] and Gingold and Monaghan [5] introduced the smooth particle hydrodynamics (SPH) approach, and two comprehensive SPH reviews are presented by Benz [6] and Monaghan [7]. A significant contribution to SPH was the addition of strength by Libersky and Petschek in 1990 [8]. More recently, a normalized smoothing function (NSF) algorithm was presented by Johnson and Beissel [9]. This algorithm increases the accuracy of the basic SPH algorithm and enables it to pass the patch test, which means that it produces the correct constant strain rates for a linear velocity field.

A completely different particle algorithm, designated NABOR, was presented by Johnson for liquids in 1983 [10], and by Johnson *et al.* for solids in 1986 [11]. The 1986 reference includes the linking together of NABOR (particle) nodes and finite elements. Some recent examples of linking are provided by Johnson *et al.* [12], and by Swegle and Attaway [13].

It should be noted that the GPA carries all of the variables (position, velocity, strain, stress, etc.) at the nodes, and this general type of formulation is subject to tensile instabilities [14]. As a result, its applications are currently limited. The authors are aware of numerous current attempts to develop particle (meshless) algorithms that are accurate, robust, fast and suitable for a wide range of applications. It is not yet clear if any one algorithm will meet all of these requirements. It is more probable that different algorithms will be better suited for different applications. It is anticipated, however, that the interface algorithms presented in this paper could be applied to a variety of particle algorithms.

## 2. PREVIOUS GPA

The previous GPA presented by the authors [1] will be included herein to provide a basis for the new modifications and additions. It will be presented in 2D plane strain for simplicity, but the extension to 2D axisymmetric geometry is straightforward [1].

Figure 1 represents some features of the GPA technique. Node  $i$  is designated as the centre node and the neighbour nodes are designated as nodes  $j$ . The distance between nodes is  $r_{ij}$ , the diameters of the nodes are  $d_i$  and  $d_j$ , and the masses of the nodes are  $M_i$  and  $M_j$ . The masses remain constant throughout the computation, and are obtained from  $M = \rho_0 V_0$  where  $\rho_0$  and  $V_0$  represent the initial density of the material and the initial volume represented by the node.

### 2.1. Weighting functions

To determine the strain rates for each centre node  $i$ , the contributions of the neighbour nodes  $j$  must be considered. The effect of the neighbour nodes is dependent on the distance between nodes  $i$  and  $j$ , the size (2D area or 3D volume) of node  $j$ , and the alignment of node  $j$

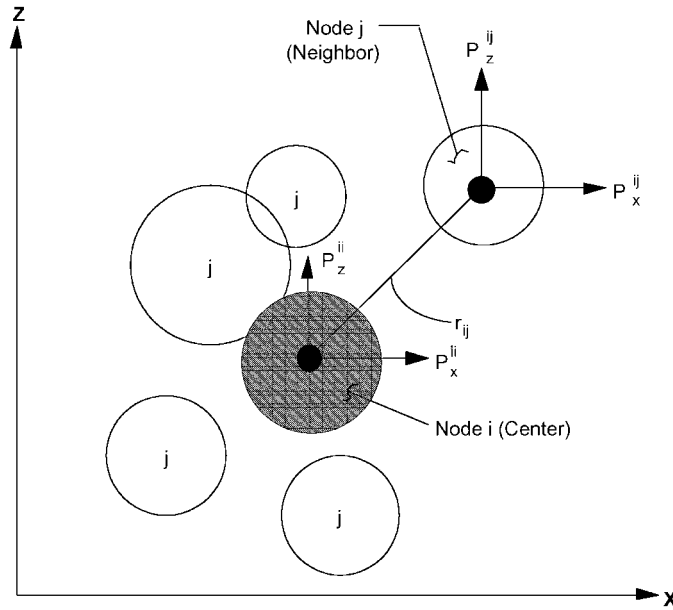


Figure 1. GPA characteristics.

relative to the direction of the strain rate being determined. These same weighting functions are used for the determination of the forces.

The distance weighting function is assumed to decrease with distance.

$$f(r) = \begin{cases} R_{ij} = 1.0 - v_{ij}/2 & \text{for } 0 \leq v_{ij} \leq 2 \\ 0 & \text{for } v_{ij} \geq 2 \end{cases} \quad (1)$$

where  $v_{ij} = r_{ij}/h_{ij}$ , and the influence distance is

$$h_{ij} = \beta(d_i + d_j)/2 \quad (2)$$

The dimensionless influence distance,  $\beta$ , is a user supplied input. It is usually taken as  $\beta = 1.0$ , but other values can be used ( $0.8 \leq \beta \leq 1.5$ ). The diameters,  $d_i$  and  $d_j$ , can be obtained (for plane strain geometry) from

$$d = d_0 \sqrt{1 + \varepsilon_v} \quad (3)$$

where  $\varepsilon_v$  is the volumetric strain (defined later) and  $d_0$  is the initial node diameter. The radius of influence is  $2h_{ij}$ , which is 2.0 node diameters for  $\beta = 1.0$ . This definition is used to be consistent with previous SPH formulations.

The weighting function for the size of the node is directly proportional to the cross-sectional area represented by the node.

$$g(s) = A_j \quad (4)$$

The weighting functions for the alignment of node  $j$  with node  $i$ , for the strain rates of interest, are given by

$$h(\theta_x) = (\ell_x^{ij})^2 \quad (5)$$

$$h(\theta_z) = (\ell_z^{ij})^2 \quad (6)$$

where  $\theta_x^{ij}$  is the angle between the line connecting nodes  $i$  and  $j$ , with the  $x$ -axis, and  $\theta_z^{ij}$  is the angle with the  $z$ -axis. The corresponding direction cosines are  $\ell_x^{ij} = (x_j - x_i)/r_{ij}$  and  $\ell_z^{ij} = (z_j - z_i)/r_{ij}$ .

## 2.2. Strain rates

For plane strain geometry, the two normal strain rates ( $\dot{\epsilon}_x^i, \dot{\epsilon}_z^i$ ), the shear strain rate,  $\dot{\gamma}_{xz}^i$ , the rotational rate,  $\omega_{xz}^i$ , and the volumetric strain rate,  $\dot{\epsilon}_v^i$ , for centre node  $i$ , are as follows:

$$\dot{\epsilon}_x^i = \alpha_x^i \sum_j R_{ij} A_j (\dot{u}_j - \dot{u}_i) \ell_x^{ij} / r_{ij} \quad (7)$$

$$\dot{\epsilon}_z^i = \alpha_z^i \sum_j R_{ij} A_j (\dot{v}_j - \dot{v}_i) \ell_z^{ij} / r_{ij} \quad (8)$$

$$\dot{\gamma}_{xz}^i = \sum_j R_{ij} A_j [\alpha_z^i (\dot{u}_j - \dot{u}_i) \ell_z^{ij} + \alpha_x^i (\dot{v}_j - \dot{v}_i) \ell_x^{ij}] / r_{ij} \quad (9)$$

$$\omega_{xz}^i = \sum_j R_{ij} A_j [\alpha_x^i (\dot{v}_j - \dot{v}_i) \ell_x^{ij} - \alpha_z^i (\dot{u}_j - \dot{u}_i) \ell_z^{ij}] / 2r_{ij} \quad (10)$$

$$\dot{\epsilon}_v^i = \dot{\epsilon}_x^i + \dot{\epsilon}_z^i \quad (11)$$

Where  $R_{ij}$  is the distance weighting function,  $A_j$  is the current area of node  $j$ ,  $\dot{u}_i$  and  $\dot{u}_j$  are the  $x$  velocities of nodes  $i$  and  $j$ ,  $\dot{v}_i$  and  $\dot{v}_j$  are the  $z$  velocities,  $\ell_x^{ij}$  and  $\ell_z^{ij}$  are the direction cosines from node  $i$  to node  $j$ , and  $r_{ij}$  is the distance between nodes  $i$  and  $j$ . The volumetric strain is obtained by integrating the volumetric strain rate.

$$\epsilon_v^{t+\Delta t} = \epsilon_v^t + \dot{\epsilon}_v \Delta t (1 + \epsilon_v^t) \quad (12)$$

Where  $\Delta t$  is the integration time increment and the factor  $(1 + \epsilon_v^t)$  converts the incremental strain ( $\dot{\epsilon}_v \Delta t$ ) from the current configuration back to the initial configuration. The volumetric strain,  $\epsilon_v = V/V_0 - 1$  must be based on the initial volume for the pressure computations, where  $V$  is the current volume and  $V_0$  is the initial volume.

The remaining two geometry factors in Equations (7)–(10) are

$$\alpha_x^i = \frac{1}{\sum_j R_{ij} A_j (\ell_x^{ij})^2} \quad (13)$$

$$\alpha_z^i = \frac{1}{\sum_j R_{ij} A_j (\ell_z^{ij})^2} \quad (14)$$

It should be noted that these geometry factors were obtained in the global co-ordinate system for the previous algorithm [1]. For interior nodes this is not generally important, but for boundary and interface nodes it can be significant. This effect is addressed in the improved algorithm.

The derivation of the preceding equations begins with the strain rate in the  $x$  direction.

$$\dot{\epsilon}_x^i = \frac{\partial \dot{u}}{\partial x} = \frac{\sum_j (\Delta \dot{u}_{ij} / \Delta x_{ij}) f(r) g(s) h(\theta_x)}{\sum_j f(r) g(s) h(\theta_x)} \quad (15)$$

Where  $\Delta \dot{u}_{ij} = \dot{u}_j - \dot{u}_i$  is the  $x$  velocity difference between nodes  $i$  and  $j$ , and  $\Delta x_{ij}$  is the  $x$  co-ordinate difference. Basically, all values of  $\Delta \dot{u}_{ij} / \Delta x_{ij}$  are weighted by the distance function,  $f(r)$ , the size function,  $h(s)$ , and the alignment function with the  $x$  axis,  $g(\theta_x)$ . It is clear that the strain rate in Equation (15) passes the patch test because  $\dot{\epsilon}_x^i \equiv \Delta \dot{u}_{ij} / \Delta x_{ij}$  for a linear velocity field (where  $\Delta \dot{u}_{ij} / \Delta x_{ij}$  is constant). The details of the deviations to obtain Equations (7)–(11) are provided in Reference [1].

### 2.3. Forces

After the strain rates, rotational rate, and volumetric strain in Equations (7)–(12) are determined, it is possible to determine the shear and deviator stresses, the pressure, and the nodal artificial viscosity in the standard manner [15]. These stresses must then be converted to forces, with the final form of the force equations given by

$$P_x^{ij} = \frac{-R_{ij} A_i A_j}{r_{ij}} [\alpha_x^i (\sigma_x^i - Q_{ij}) \ell_x^{ij} + \alpha_z \tau_{xz}^i \ell_z^{ij}] \quad (16)$$

$$P_z^{ij} = \frac{-R_{ij} A_i A_j}{r_{ij}} [\alpha_z^i (\sigma_z^i - Q_{ij}) \ell_z^{ij} + \alpha_x \tau_{xz}^i \ell_x^{ij}] \quad (17)$$

$$P_x^{ii} = -\sum_j P_x^{ij} \quad (18)$$

$$P_z^{ii} = -\sum_j P_z^{ij} \quad (19)$$

In Equations (16) and (17)  $P_x^{ij}$  and  $P_z^{ij}$  are the forces on node  $j$  due to the stresses in node  $i$ . The normal stress in the  $x$  direction (for node  $i$ ) is  $\sigma_x^i = s_x^i - (P_i + Q_i)$ , where  $s_x^i$  is the deviator stress,  $P_i$  is the hydrostatic pressure and  $Q_i$  is the nodal artificial viscosity. The shear stress is  $\tau_{xz}^i$ . There is also an artificial viscosity,  $Q_{ij}$ , which is dependent on the relative velocities of nodes  $i$  and  $j$  [16]. It is intended to stabilize the grid and keep adjacent nodes from becoming too close to one another. This is designated as a bond viscosity, because it acts on the bond between nodes  $i$  and  $j$ . It can also introduce a significant amount of artificial strength [17]. Equations (18) and (19) provide the  $x$  and  $z$  forces acting on centre node  $i$  due to the stresses in centre node  $i$ . They are the equal and opposite sums of the forces on the neighbour  $j$  nodes. The derivation of the preceding force equations begins with the plane strain equilibrium equations and the details are provided in Reference [1].

Equations (16)–(19) consider the forces on nodes  $i$  and  $j$ , due to the stresses in node  $i$  only. The sums of these forces (on neighbour nodes  $j$  and centre node  $i$ ) are equal and

opposite such that momentum is absolutely conserved. The stresses in neighbour node  $j$  are considered when node  $j$  becomes centre node  $i$ , and centre node  $i$  becomes neighbour node  $j$ . Furthermore, with this approach it is possible to treat the boundaries and interfaces in a straightforward manner.

Figure 2 shows what the force equations provide for interior node  $A$ , interface node  $B$  and boundary node  $C$ . For plane strain geometry with a unit thickness, if the applied pressure is taken as  $P_0 = 1.0$ , and the nodal diameters are all  $d = 1.0$ , then every node should have a net downward force of 1.0 from the nodes above, as well as a net upward force of 1.0 from the nodes below. For this analysis the influence distance in Equation (2) is  $\beta = 1.0$ , and this means that the distance weighting function in Equation (1) goes to zero at a distance of  $2d$ . The result is that an interior node exchanges forces with the three nodes above it and the three nodes below it for the specific loading in Figure 2. Again, the details of these force computations are provided in Reference [1].

For interior node  $A$ , the forces on neighbour nodes  $j$ , due to stresses in centre node  $i$ , also act on centre node  $i$ , in an equal and opposite manner, as provided in Equations (17) and (19). These are shown under the heading of 'Forces from stress in centre node  $i$ ,' for node  $A$  in Figure 2. Also shown are the 'Forces from stresses in neighbour nodes  $j$ ,' and these are identical because all of the neighbour nodes are also interior nodes with  $\alpha_z^i = 0.6306$ . The net forces on node  $A$  are exactly 1.0000 acting downward from the three neighbour nodes above ( $j = 1, 2, 3$ ), and exactly 1.0000 acting upward from the three neighbour nodes below ( $j = 6, 7, 8$ ). This tends to confirm the force equations for an interior node.

The forces acting on node  $B$  are also shown in Figure 2. This is an interface node that could be in contact with a finite element grid or a GPA node of a different material. A significant difference, between interface node  $B$  and interior node  $A$ , is that the geometry factor from Equation (14) is computed to be  $\alpha_z^i = 1.2612$ . This is exactly twice the magnitude of  $\alpha_z^i$  for an interior node, and it produces forces from the stress in node  $i$  that are twice those obtained from an interior node. The forces from stresses in neighbour nodes  $j$  are identical to those obtained for interior node  $A$  as the neighbours to node  $B$  ( $j = 1, 2, 3$ ) are all interior nodes with  $\alpha_z^i = 0.6306$ .

Summing the forces between node  $B$  and neighbour nodes  $j = 1, 2, 3$  gives a net downward force of 1.5000 on node  $B$ , instead of the correct force of 1.0000. The source of this error is that the forces from the stress in node  $i$  are two times higher than they should be, and this is because  $\alpha_z^i$  is two times higher than for an interior node. Even though the downward force acting on node  $B$  is too high, the system is in equilibrium because every force on neighbour node  $j$  (from the stress in centre node  $i$ ) is reacted by an equal and opposite force acting on node  $i$ .

The forces acting on node  $C$  are shown at the bottom of Figure 2. For this boundary node  $\alpha_z^i = 0.7735$  from Equation (14) and the net forces acting on both the top and bottom of node  $C$  are 0.9791. This is an error of only 2.1 per cent, and the forces tend to cancel one another for a constant stress state. It is only for stress gradients that errors are introduced into the solution, and these errors are small.

A very important characteristic of the examples in Figure 2 is that the boundaries and interfaces are aligned with the global system axes  $x$  and  $z$ , and this affects the determination of the geometry factors ( $\alpha_x^i$  and  $\alpha_z^i$ ) in Equations (13) and (14). If the boundaries and interfaces are not aligned with the global system axes then the errors are increased. This is illustrated in Figure 3, where the net forces on boundary node  $C$  are computed for two cases where

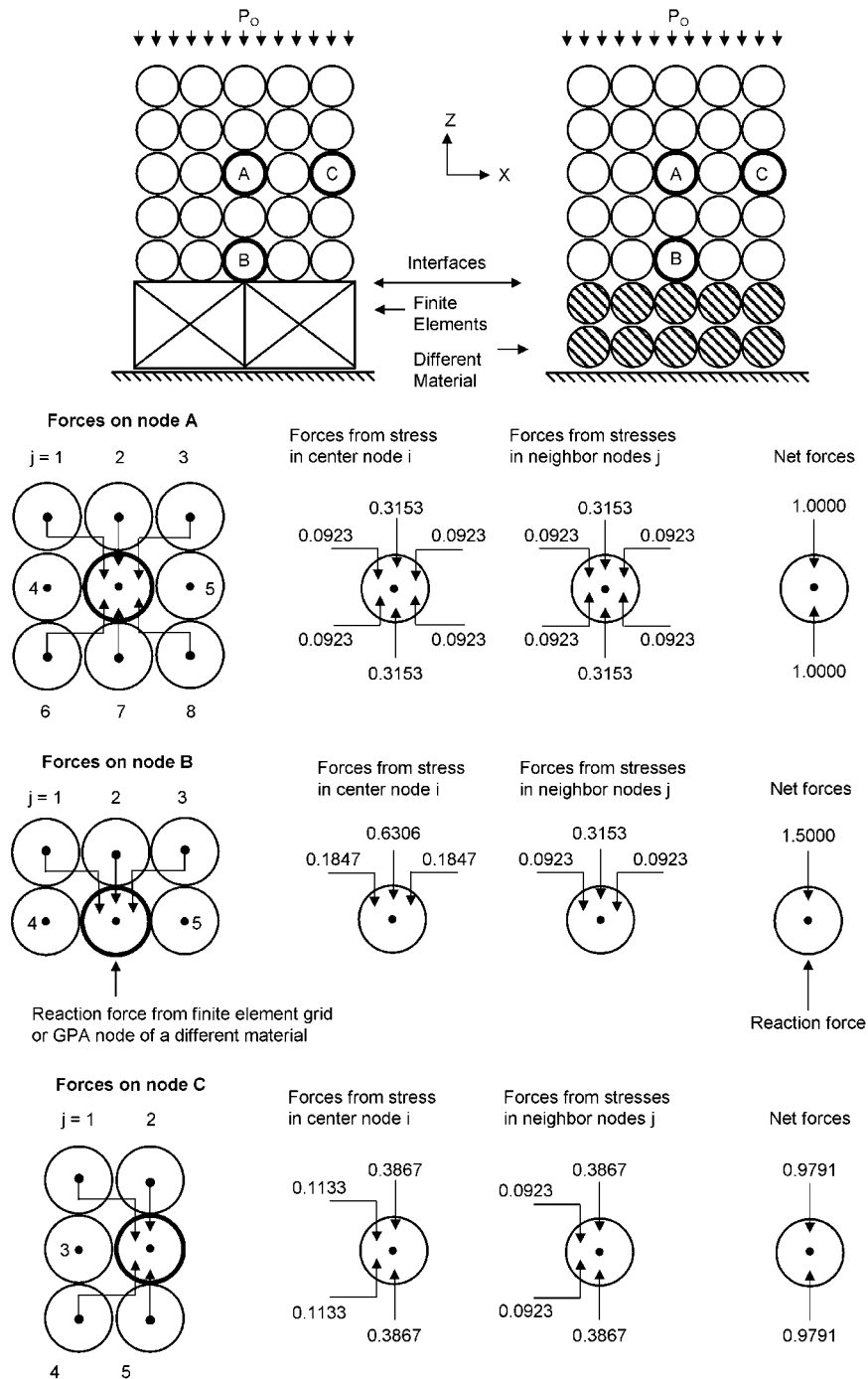


Figure 2. Forces on nodes A, B and C.

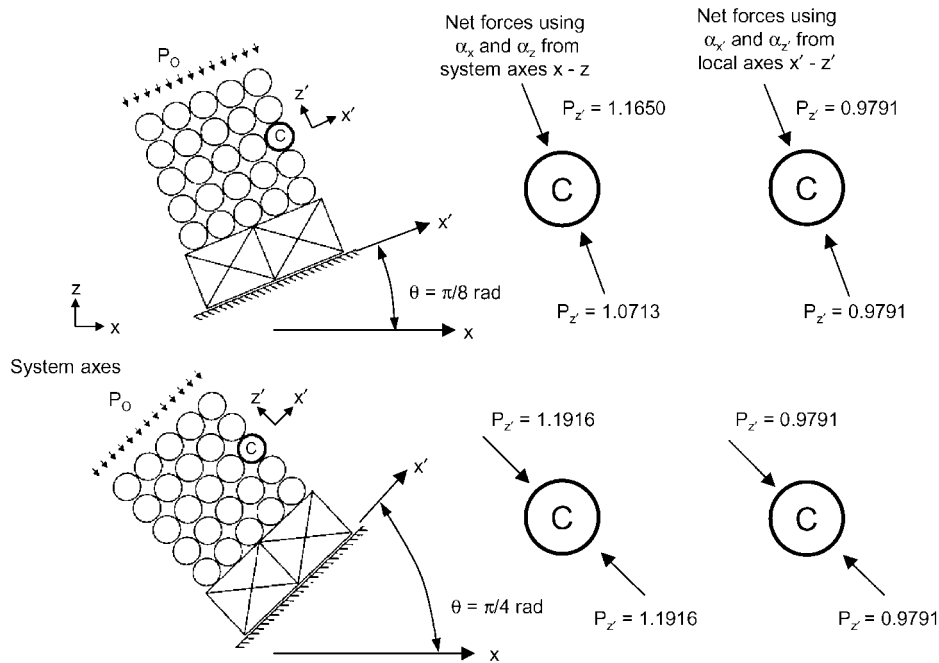


Figure 3. Forces on boundary node C for different orientations.

the boundaries and interfaces do not coincide with the system axes. For an orientation of  $\theta = \pi/8$  rad, the force (parallel to the boundary and loading) acting on the top of node C is  $P_{z'} = 1.1650$  and the force acting on the bottom is  $P_{z'} = 1.0713$ . These errors are 16.5 and 7.1 per cent, respectively, significantly greater than the 2.1 per cent error associated with the aligned conditions in Figure 2. Furthermore, they are not equal to one another. A similar situation exists when  $\theta = \pi/4$  rad, where the parallel forces are  $P_{z'} = 1.1916$  on both the top and the bottom. For this condition the forces on the top and bottom are equal, but the errors are increased to 19.2 per cent. Although the specific force distributions for interface node B (under various orientations) are not shown, the general trends are similar to those experienced by boundary node C.

An obvious solution for the orientated cases in Figure 3 is to compute the geometry factors ( $\alpha_x^i$  and  $\alpha_z^i$ ) in a local co-ordinate system ( $x', z'$ ) that is parallel to the boundary and the loading. These local co-ordinate geometry factors are designated  $\alpha_{x'}$  and  $\alpha_{z'}$  (for the remainder of this paper the  $i$  superscript will not be included to indicate it is for centre node  $i$ ), and this provides an invariant set of forces ( $P_{z'} = 0.9791$ ) as shown on the right-hand side of Figure 3. The improved (invariant) algorithm which follows is based on this concept.

### 3. IMPROVED (INVARIANT) GPA

The improved GPA determines the geometry factors (Equations (13) and (14)), the strain and rotational rates (Equations (7)–(10)), and the forces (Equations (16)–(19)), in a local



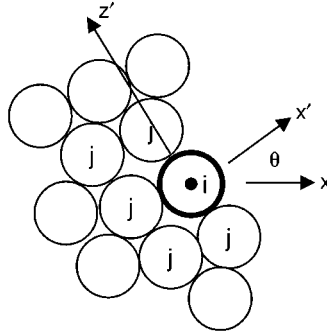


Figure 4. The local  $x'-z'$  axes for a boundary node.

co-ordinate system  $x'-z'$ . For boundaries and interfaces the  $x'$ -axis is defined to be normal to the surface as shown in Figure 4. This allows the boundary and interface algorithms to be applied in a straightforward manner.

The  $x'-z'$  axes can be determined from a technique similar to that presented by Randles and Libersky [18].

$$\eta_x = \frac{-\sum R_{ij} A_j \ell_x^{ij}}{\sum R_{ij} A_j} \quad (20)$$

$$\eta_z = \frac{-\sum R_{ij} A_j \ell_z^{ij}}{\sum R_{ij} A_j} \quad (21)$$

where  $\eta_x$  and  $\eta_z$  are vector components of the  $x'$ -axis. The rotation from the global  $x-z$  axes to the local  $x'-z'$  axes is defined by

$$\sin \theta = \eta_z / \sqrt{\eta_x^2 + \eta_z^2} \quad (22)$$

$$\cos \theta = \eta_x / \sqrt{\eta_x^2 + \eta_z^2} \quad (23)$$

The angle  $\theta$  is required for the strain rate, stress and force transformations between the two axes orientations.

Equations (20)–(23) are applied to all nodes in the initial geometry at time,  $t=0$ . If a particle interface algorithm is used (for interaction between particles of different materials), then neighbour nodes of different materials are not included. If the node is an interior node in a regular arrangement of nodes then  $\eta_x = \eta_z = 0$ . For this case it is assumed that the local  $x'-z'$  axes coincide with the system  $x-z$  axes. If the magnitude of the normal vector  $\eta = \sqrt{\eta_x^2 + \eta_z^2}$  is small compared to that of the boundary and interface nodes, then it is an interior node in a non-regular arrangement. Larger values of the normal vector indicate a boundary or interface node. For a dimensionless influence distance of  $\beta = 1.0$  the vector length is  $\eta = 0.4383$  for a boundary node along the edge of a regular arrangement of nodes. Generally, a surface or boundary node in the initial geometry will have  $\eta > 0.2$ .

If the boundary and interface nodes in the initial geometry remain as boundary and interface nodes throughout the computation, then they can be designated as boundary and interface nodes for the duration of the computation. It will later be shown that it is not important to designate boundary nodes for free boundaries, because the normal stresses on free boundaries tend to go to zero.

The following represents the computational sequence that is applied to every GPA node for every cycle of integration:

- Determine the local rotated co-ordinate system ( $x'-z'$ ) for all nodes (interior, boundary, interface) from Equations (20)–(23).
- Compute the geometry factors ( $\alpha_{x'}, \alpha_{z'}$ ) in the local rotated co-ordinate system using the form of Equations (13) and (14). For interior nodes the use of the local rotated axes provides an invariant solution that is not dependent on the choice of the global system axes orientation. For most boundary and interface nodes  $\alpha_{x'}$  is reduced to one-half the computed value. This is the only modification applied to the boundary and interface nodes and it will be discussed in more detail later.
- Compute local strain and rotational rates ( $\dot{\epsilon}_{x'}, \dot{\epsilon}_{z'}, \dot{\gamma}_{x'z'}, \omega_{x'z'}$ ) in the local rotated co-ordinate system ( $x'-z'$ ) using  $\alpha_{x'}$  and  $\alpha_{z'}$  and the form of Equations (7)–(10).
- Convert the local strain rates ( $\dot{\epsilon}_{x'}, \dot{\epsilon}_{z'}, \dot{\gamma}_{x'z'}$ ) to global system strain rates ( $\dot{\epsilon}_x, \dot{\epsilon}_z, \dot{\gamma}_{xz}$ ) using transformations through an angle  $\theta$  given by Equations (22) and (23).
- Compute the stresses ( $\sigma_x, \sigma_z, \tau_{xz}$ ) in the global co-ordinate system from the corresponding strains and strain rates ( $\dot{\epsilon}_x, \dot{\epsilon}_z, \dot{\gamma}_{xz}$ ) in the standard manner [15]. An alternative approach would be to compute the stresses in the local co-ordinate system. This alternate approach was not used for the present work because it was not as well suited for the structure of the code.
- Convert stresses in the global co-ordinate system ( $\sigma_x, \sigma_z, \tau_{xz}$ ) to stresses ( $\sigma_{x'}, \sigma_{z'}, \tau_{x'z'}$ ) in the local rotated co-ordinate system, through the angle  $\theta$ .
- Compute forces ( $P_{x'}, P_{z'}$ ) in the local rotated co-ordinate system using the form of Equations (16)–(19).
- Convert the rotated forces ( $P_{x'}, P_{z'}$ ) to equivalent forces in the global co-ordinate system ( $P_x, P_z$ ), through the angle  $\theta$ .

In addition to the preceding computational steps that are applied to each GPA node, some additional interface forces must also be applied (as described in the next section). After all forces have been applied, the equations of motion (new velocities and displacements) are updated in a standard manner [15].

#### 4. BOUNDARY AND INTERFACE ALGORITHMS

The improved (invariant) GPA ensures that the local  $z'$ -axis is parallel to the surface for boundary and interface nodes. This, in turn, minimizes the errors for the forces in the  $z'$  direction. The  $z$  forces on node  $C$  in Figure 2 and the  $z'$  forces on node  $C$  in Figure 3 are in error by only 2.1 per cent. Furthermore, they are equal and opposite such that the net force is in error only when there is a stress gradient in the  $z'$  direction.

The forces normal to the boundaries and interfaces are in the  $x'$  direction as shown in Figure 4 and described by Equations (20)–(23). In Figure 3 it was previously noted that

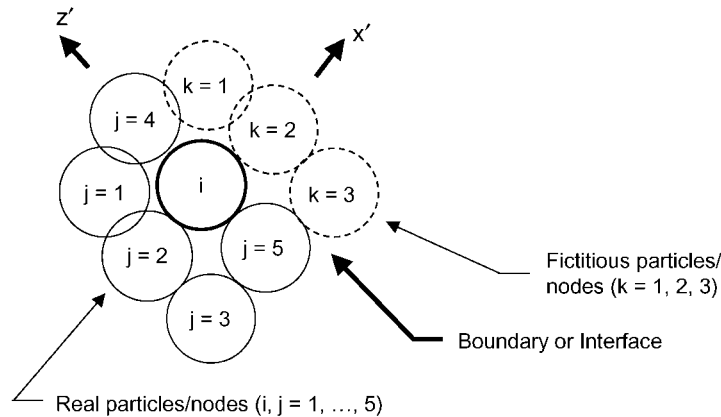


Figure 5. Effective geometry factor  $\alpha_{x'}$  for boundary and interface nodes.

the net force acting downward on node  $B$  (which would be in the  $x'$  direction for the local rotated co-ordinate system) is too large because the normal geometry factor ( $\alpha_z$  in the global co-ordinate system and  $\alpha_{x'}$  in the local rotated co-ordinate system) is too large (by a factor of 2.0) for interface node  $B$ .

Figure 5 shows how the factor of 2.0 is introduced. Centre node  $i$  has five neighbours ( $j=1, \dots, 5$ ). The  $j=1, 2, 3$  neighbours contribute to  $\alpha_{x'}$  and the  $j=4, 5$  neighbours contribute little because  $\ell_{x'}^{ij}$  (from the form of Equation (13)) is very small. For a given stress state in nodes  $i$  and  $j=1, \dots, 5$ , the forces between node  $i$  and its neighbours ( $j=1, \dots, 5$ ) should be independent of any forces applied at the boundary/interface. If the three fictitious particles/nodes corresponding to nodes  $k=1, 2, 3$  are included in the determination of  $\alpha_{x'}$ , their contribution is identical to that of nodes  $j=1, 2, 3$  and the revised (or effective) value of  $\alpha_{x'}$ , is approximately one-half of that obtained from the form of Equation (13) using only the real neighbour nodes ( $j=1, \dots, 5$ ). Including the fictitious nodes ( $k=1, 2, 3$ ) makes node  $i$  appear to be in the centre of a group of neighbour nodes ( $j=1, \dots, 5$  and  $k=1, 2, 3$ ) such that the application of force Equations (16)–(19) is appropriate. The use of this approach provides the correct net force (1.0000) acting downward on node  $B$  in Figure 2.

#### 4.1. Boundary and interface conditions

Figure 6 shows six different classes of boundaries and interfaces. For all cases the  $x'-z'$  axes are shown for reference. The free boundary condition can be used with or without the revised (reduced)  $\alpha_{x'}$  factor. If the distortions are mild, and the initial boundary nodes are the only boundary nodes throughout the computation, then they can be marked in the initial geometry and treated as boundaries (with the revised  $\alpha_{x'}$ ) for the duration of the problem. If the distortions in the problem are very severe and new free boundaries are introduced during the course of the computation then it is probably preferable not to apply any special treatment to the free boundaries. For free boundaries the normal stresses are very low and the normal

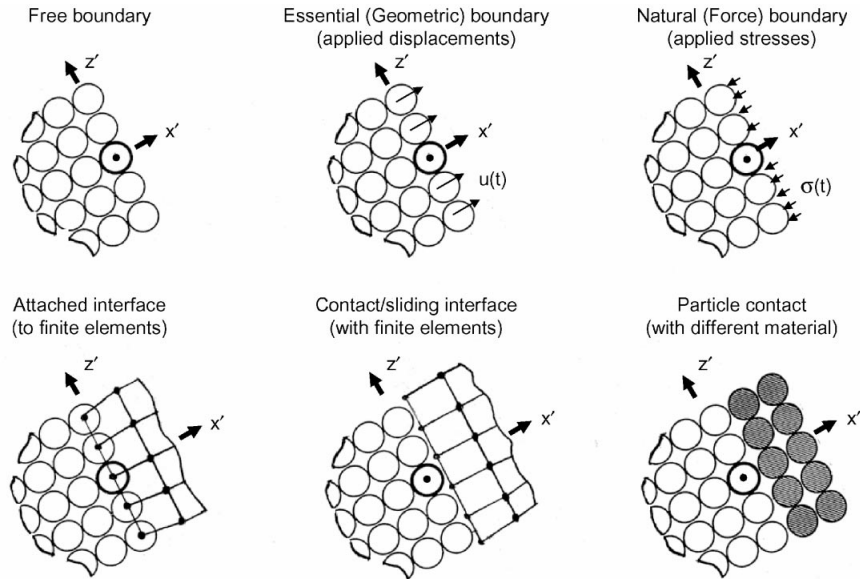


Figure 6. The different classes of boundaries and interfaces.

forces are also low. The factor of 2.0 simply has little effect in the relatively low normal forces.

Another possible way to identify boundaries during the course of the computation would be to evaluate the length of the normal vector,  $\eta$ , of each node for every cycle. This length is always in the range of 0.0–1.0. For the initial geometry the interior nodes have very small vector lengths and the boundaries have distinctly greater lengths (such as  $\eta = 0.4843$  for an edge node as noted previously). For problems with high distortions, such as hypervelocity impact problems, there can be a continuous range of vector lengths and the free boundaries may not be readily distinguishable.

The next two boundary conditions in Figure 6 are the essential (geometric) boundary and the natural (force) boundary. For these two conditions there can be significant stresses and forces normal to the boundary (in the  $x'$  direction) and the revised  $\alpha_{x'}$  factor should be used. For these classes of problems the boundaries are generally well defined throughout the duration of the computation and the boundaries should thereby be identified and marked in the initial geometry.

The fourth condition consists of a finite element grid attached to GPA nodes. For this condition the GPA interface nodes accept forces from both the GPA nodes and the finite elements. Because there can be significant stresses and forces normal to the interface (in the  $x'$  direction) the revised  $\alpha_{x'}$  factor should be used. These attached interface GPA nodes can be readily identified and marked in the initial geometry.

The fifth condition occurs when GPA nodes contact and/or slide on the surface of a finite element grid. Contact occurs when the outer edge of the circular GPA node crosses the edge of a finite element. Again, there can be significant stresses and forces normal to the

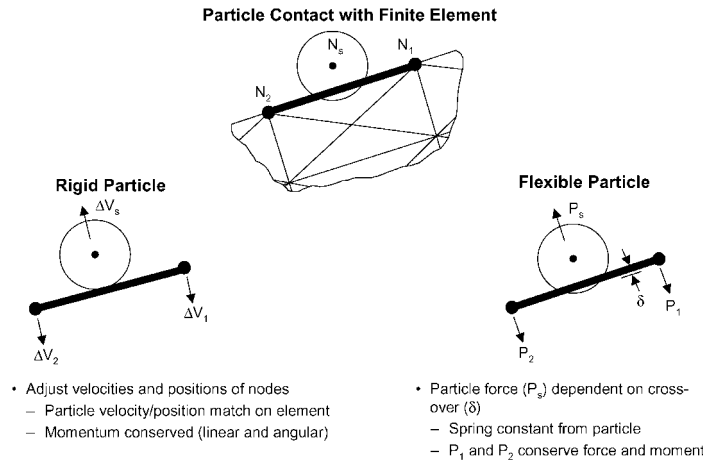


Figure 7. Contact/sliding algorithms for a GPA node interacting with a finite element grid.

interface (in the  $x'$  direction) and the revised  $\alpha_{x'}$  factor should be used (but only when there is contact).

The last interface condition is when GPA nodes of different materials interact at an interface. Here the neighbour nodes  $j$  are only those nodes that are of the same material as centre node  $i$ . Contact occurs when the outer edges of the circular GPA nodes cross over one another. This final condition can also experience significant stresses and forces normal to the interface (in the  $x'$  direction) and the revised  $\alpha_{x'}$  factor should be used (but only when there is contact).

#### 4.2. Interface algorithm for particle–finite element interaction

Figure 6 shows two different conditions for particles interacting with finite elements. The lower left shows particles attached to the finite element grid and this requires no special interface algorithm (other than using the revised  $\alpha_{x'}$  factor). The interface GPA node simply accepts forces from both the neighbouring GPA nodes and the attached elements.

The lower centre portion of Figure 6 shows a GPA node contacting and/or sliding on a finite element grid. More detail is shown in Figure 7. Node  $N_s$  is the GPA node (sometimes referred to as a slave node), and nodes  $N_1$  and  $N_2$  are the finite element nodes that define the surface (sometimes referred to as the master nodes/surface) with which the GPA node must interact. The particle GPA node is assumed to be in contact with the finite element grid when the equations of motion cause the edge of the circular particle to cross over the master surface defined by nodes  $N_1$  and  $N_2$ . Recall that the GPA nodal diameter is given by Equation (3).

The lower left portion of Figure 7 shows how the velocities are updated ( $\Delta V_s, \Delta V_1, \Delta V_2$ ) if it is assumed that the particle is rigid. The three conditions required to determine the three velocity changes are as follows:

- Conservation of linear momentum normal to the finite element grid (master surface).
- Conservation of angular momentum.

- A velocity and position match at the GPA (slave) node location (contact point) in the finite element grid (master surface). This condition requires the use of an iterative technique because some master nodes are affected by more than one slave node.

If the surface is frictionless, the nodal equations of motion are unaffected in the direction parallel to the master surface and this allows for sliding between the GPA node and the finite element grid. Frictional effects can also be included and they are constrained to the direction parallel to the master surface. This rigid particle approach is similar to that used for some finite element contact/sliding algorithms [19, 20].

The specific expression for the normal velocity change to the slave node is

$$\Delta V_s = \frac{(\delta/\Delta t)}{(1 + R_1^2 M_s/M_1 + R_2^2 M_s/M_2)} \quad (24)$$

where  $\delta$  is the distance the slave node crosses over the master surface before the contact algorithm is applied,  $\Delta t$  is the integration time increment,  $R_1$  and  $R_2$  are the fractions of momentum transferred to master nodes  $N_1$  and  $N_2$ , and  $M_s, M_1$ , and  $M_2$  are the nodal masses. An expanded discussion and the derivation of Equation (24) is provided elsewhere [19, 20].

The lower right portion of Figure 7 shows how resistance forces ( $P_s, P_1, P_2$ ) are generated if it is assumed that the particle is flexible. This approach may provide increased accuracy for some problems and it is well suited for particle nodes with real physical diameters. The force on the GPA (slave) node is determined first, and the reaction forces on the master nodes ( $P_1, P_2$ ) are determined by conserving normal forces and moments. This approach does not require an iterative technique.

The force on the GPA (slave) node contains an elastic component and sometimes a viscous component.

$$P_s = P_{\text{elastic}} + P_{\text{viscous}} \quad (25)$$

The elastic portion is simply

$$P_{\text{elastic}} = k\delta \quad (26)$$

where  $k$  is the effective spring constant of the GPA node and  $\delta$  is the distance the GPA node has crossed over the master surface.

For an elastic bar the spring stiffness is

$$k = AE/\ell \quad (27)$$

where  $A$  is the cross-sectional area,  $E$  is the modulus of elasticity and  $\ell$  is the length. The projected area of the particle onto the master surface is  $A = V/d$  where  $V$  is the current volume and  $d$  is the current diameter. (This expression for  $A$  is valid for both plane strain and axisymmetric geometry.) The effective current modulus of elasticity can be expressed as  $E = \rho c^2$ , where  $\rho$  is the current density and  $c$  is the sound velocity. Using the current density and sound velocity of the GPA node provides an approximation for nonlinear materials (including both solids and fluids). Using the substitutions for  $A$  and  $E$ , and letting  $\ell = d/2$ , gives

$$k = 2V\rho c^2/d^2 \quad (28)$$

A final substitution of  $M = V\rho$ , where  $M$  is the nodal mass, provides the final expression.

$$k = 2Mc^2/d^2 \quad (29)$$

It is possible to compare the expressions for the velocity changes to the GPA slave node using both the rigid particle and the flexible particle. The velocity change for the rigid particle is given by Equation (24) and the analogous velocity change for the flexible particle (not including the viscous force,  $P_{\text{viscous}}$ ) can be obtained from

$$\Delta V_s = P_s \Delta t / M_s = 2c^2 \delta \Delta t / d^2 \quad (30)$$

Equation (25) also includes a viscous term that is analogous to the bond viscosity often used with particle algorithms [17]. It has been designated a bond viscosity because it acts on the bond between nodes  $i$  and  $j$ . It is sometimes required to keep adjacent nodes from becoming too close to one another.

The viscous compressive stress that resists the closing motion between the GPA slave node and the master surface is taken as

$$Q = C_L \rho c d |\dot{\epsilon}| + C_Q \rho d^2 \dot{\epsilon}^2 \quad (31)$$

where  $C_L$  and  $C_Q$  are dimensionless coefficients,  $\rho$  is the density,  $c$  is the sound velocity,  $d$  is the diameter, and  $\dot{\epsilon} = \Delta V_n / h$  is the closing strain rate between the GPA node and the master surface [16, 17]. The normal closing velocity is  $\Delta V_n$  and  $h$  is the normal distance between the centre of the GPA node and the master surface. This viscosity is applied only in compression.

Now the viscous force can be determined from

$$P_{\text{viscous}} = QA \quad (32)$$

Substituting  $Q$  from Equation (31), and  $A = V/d$  gives

$$P_{\text{viscous}} = C_L \rho c d |\dot{\epsilon}| V/d + C_Q \rho d^2 \dot{\epsilon}^2 V/d \quad (33)$$

Letting  $M = \rho V$  gives the final expression for the viscous force

$$P_{\text{viscous}} = C_L M c |\dot{\epsilon}| + C_Q M d \dot{\epsilon}^2 \quad (34)$$

#### 4.3. Interface algorithm for particles of different materials

Figure 8 shows two GPA nodes in contact. If node  $j$  is a different material than node  $i$  then it could introduce errors into the strain and rotational rates given by Equations (7)–(10) if it was included as a neighbour. The magnitude of the error is dependent on the material properties and the deformation. If the two materials have similar strength and stiffness, and if they are loaded in compression, then the errors would be relatively small. If the materials are very different (such as metal and explosive), or if they are loaded in shear and/or tension, then the errors could be higher. If the only nodes that are considered to be neighbours of node  $i$  in Figure 1 are those of the same material as node  $i$ , then a contact algorithm is necessary to provide interaction between the adjacent nodes of different materials.

The effective cross-over between nodes in Figure 8 is given as  $\delta$ . This is defined as the current cross-over less any cross-over that may have occurred in the initial geometry. The velocities of nodes  $i$  and  $j$ , along the direction from node  $j$  to node  $i$ , are designated as  $V_n^i$

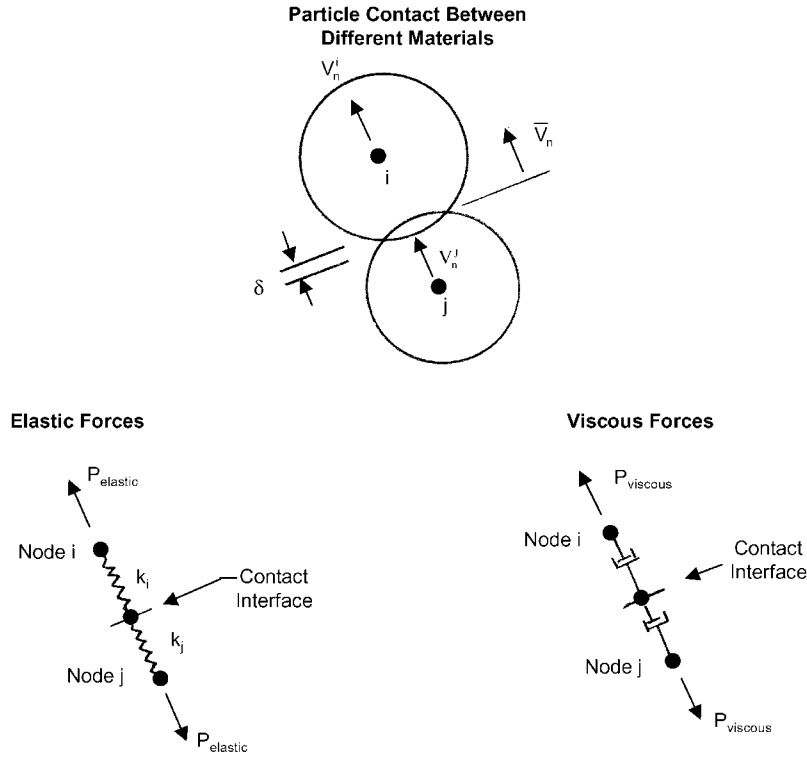


Figure 8. Contact algorithm for GPA nodes of different materials.

and  $V_n^j$ . The velocity of the interface between the nodes is  $\bar{V}_n$ . The velocities of nodes  $i$  and  $j$  are known but the interface velocity must be determined. The forces acting on nodes  $i$  and  $j$  are equal and opposite, and they are composed of an elastic component and sometimes a viscous component, as shown in the lower portion of Figure 8.

$$P_i = -P_j = P_{\text{elastic}} + P_{\text{viscous}} \quad (35)$$

The elastic portion is

$$P_{\text{elastic}} = \bar{k} \delta \quad (36)$$

where  $\bar{k}$  is the effective spring constant of two springs ( $k_i$  and  $k_j$ ) in series.

$$\bar{k} = \frac{k_i k_j}{k_i + k_j} \quad (37)$$

This formulation ensures that the force in each spring is equal to that of the other, even though the individual spring constants and compressed distances may not be equal for the



two springs. The effective spring constants for nodes  $i$  and  $j$  are simply

$$k_i = 2M_i c_i^2 / d_i^2 \quad (38)$$

$$k_j = 2M_j c_j^2 / d_j^2 \quad (39)$$

where each individual spring constant is dependent on the mass, sound velocity, and diameter of each individual node. The derivation of Equations (38) and (39) is identical to that of Equations (26)–(29) for the interface algorithm for particle–finite element interaction.

The viscous forces, shown in the lower right portion of Figure 8, are more difficult to determine. The viscous force in each node must be equal to that of the other node, which gives

$$P_{\text{viscous}} = Q_i A_i = Q_j A_j \quad (40)$$

The viscous stresses in the two nodes are  $Q_i$  and  $Q_j$ , and the projected areas are  $A_i$  and  $A_j$ .

The expressions for the artificial viscosities (viscous stresses) have the same general form as Equation (31), and are expressed as

$$Q_i = -C_L \rho_i c_i d_i \dot{\epsilon}_i + C_Q \rho_i d_i^2 \dot{\epsilon}_i^2 \quad (41)$$

$$Q_j = -C_L \rho_j c_j d_j \dot{\epsilon}_j + C_Q \rho_j d_j^2 \dot{\epsilon}_j^2 \quad (42)$$

Equations (41) and (42) differ from Equation (31) inasmuch as the absolute value of the strain rate has been replaced by the negative of the strain rate. Because compressive strain rates are negative, both terms in Equations (31), (41), and (42) are positive for compressive (negative) strain rates.

The only unknowns in Equations (41) and (42) are the strain rates,  $\dot{\epsilon}_i$  and  $\dot{\epsilon}_j$ , which can be expressed as

$$\dot{\epsilon}_i = 2\Delta V_n^i / d_i \quad (43)$$

$$\dot{\epsilon}_j = 2\Delta V_n^j / d_j = 2(\Delta V - \Delta V_n^i) / d_j \quad (44)$$

In Equation (43),  $\Delta V_n^i = V_n^i - \bar{V}_n$ , is the velocity difference between the nodal velocity  $V_n^i$  and the interface velocity  $\bar{V}_n$ . Similarly, in Equation (44),  $\Delta V_n^j = \bar{V}_n - V_n^j$ . Noting that  $\Delta V = \Delta V_n^i + \Delta V_n^j = V_n^i - V_n^j$  allows the substitution,  $\Delta V - \Delta V_n^i = \Delta V_n^j$ , in Equation (44).

Now, if the viscous forces from each node are equated ( $Q_i A_i = Q_j A_j$ ) from Equation (40), and the strain rates in Equations (43) and (44) are expressed as a function of the unknown,  $\Delta V_n^i$ , then the resulting quadratic equation can be solved for  $\Delta V_n^i$ .

$$\Delta V_n^i = \frac{-C_2 \pm \sqrt{C_2^2 - 4C_1 C_3}}{2C_1} \quad (45)$$

where

$$C_1 = 2C_Q(M_i/d_i - M_j/d_j) \quad (46)$$

$$C_2 = -C_L(M_i c_i/d_i + M_j c_j/d_j) + 4C_Q M_j \Delta V/d_j \quad (47)$$

$$C_3 = C_L M_j c_j \Delta V/d_j - 2C_Q M_j \Delta V^2/d_j \quad (48)$$

The correct solution (from the two solutions provided by Equation (45)) is when  $\Delta V < \Delta V_n^i < 0$ .

The solution for  $\Delta V_n^i$  allows the strain rate,  $\dot{\epsilon}_i$ , to be obtained from Equation (43) such that  $Q_i$  can be obtained from Equation (41). Recalling that the projected area is  $A_i = V_i/d_i$  allows the viscous force to be obtained from Equation (40).

## 5. EXAMPLES

### 5.1. Wave propagation through interfaces

The first examples are shown in Figures 9–11. These examples demonstrate the effects of the improved interface algorithms for wave propagation. All involve impact of water onto a rigid

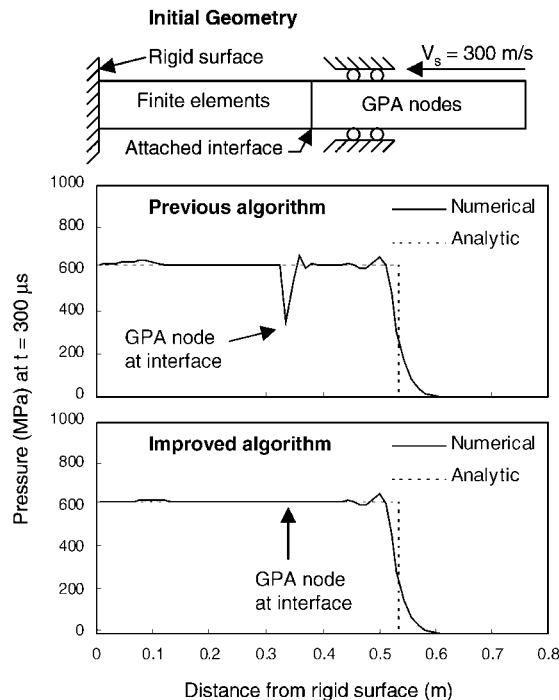


Figure 9. Wave propagation in water, through an attached interface of finite elements and GPA nodes.

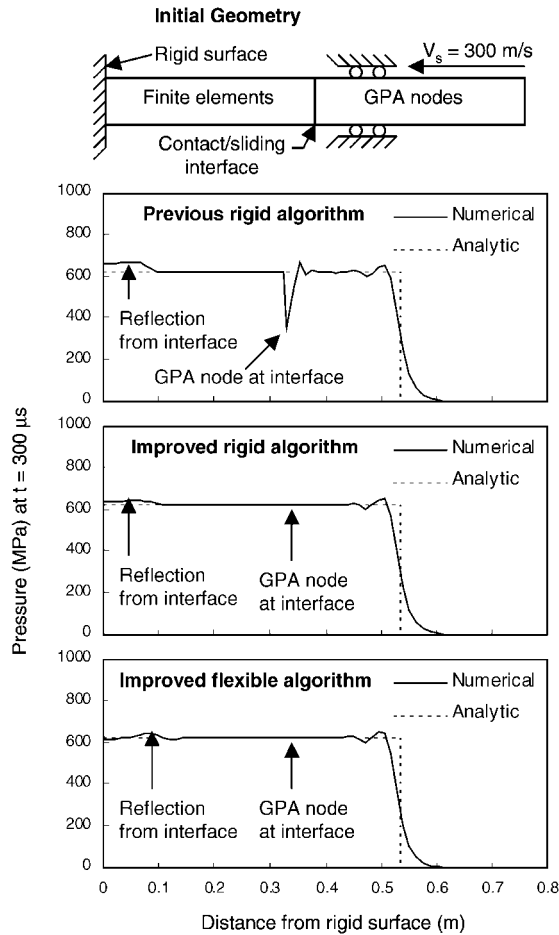


Figure 10. Wave propagation in water, through a contact/sliding interface between finite elements and GPA nodes.

surface at 300 m/s. The plane strain geometry extends sufficiently in the lateral direction such that there are no edge effects. The pressures shown in the figures are for the finite elements and GPA nodes along the centreline. A nodal artificial viscosity, with  $C_L = 0.5$  and  $C_Q = 4.0$ , was used for the results shown in Figures 9–11. For these problems the results are almost identical when a bond viscosity is used.

The pressure in the water is expressed as

$$P = K_1 \mu + K_2 \mu^2 + K_3 \mu^3 \quad (49)$$

where  $\mu = \rho/\rho_0 - 1$  for current and initial densities  $\rho$  and  $\rho_0$ . The constants are  $K_1 = 2.18$  GPa,  $K_2 = 5.17$  GPa,  $K_3 = 23.00$  GPa, and  $\rho_0 = 1000$  kg/m<sup>3</sup>. At an impact velocity of 300 m/s

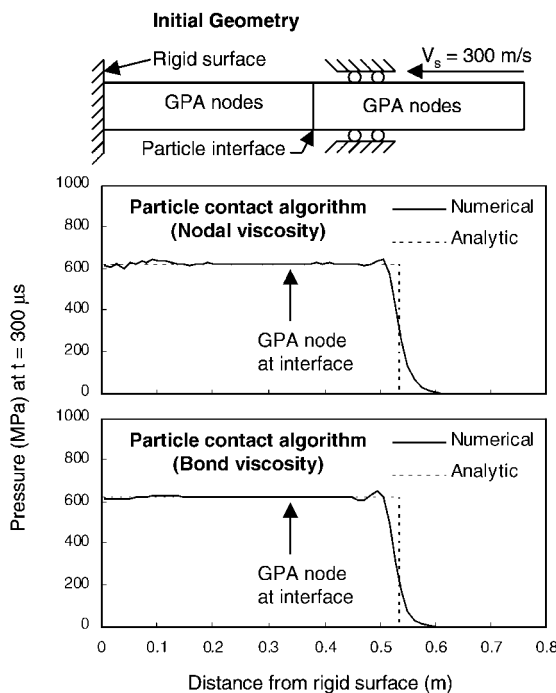


Figure 11. Wave propagation in water, through a particle contact interface.

(mach 0.2) the volumetric compression is  $\mu = 0.169$ , the pressure is  $P = 624$  MPa and the wave propagation velocity is 2080 m/s (which is much higher than the acoustic sound velocity of 1480 m/s) [21, 22].

In Figure 9, the finite elements are attached to the GPA nodes at the interface in a manner identical to that shown in the lower left portion of Figure 6. For the previous algorithm the GPA node at the interface has a significantly reduced pressure. This occurs because the force on the GPA interface node is too large for the pressure (without the boundary correction on the  $\alpha_x$  factor). The volumetric strain and pressure tend to compensate for the excessive force by dropping down until the force is in equilibrium with the other nodes. The net result is that the interface error is very localized such that the wave is propagated through the interface at the correct magnitude and velocity. Other than the GPA interface node, the computed result is in good agreement with the analytic solution.

For all of the water impact problems the interface at 300  $\mu$ s is to the left of the interface in the initial geometry. This is due to the significant compression ( $\mu = 0.169$ ) of the water as noted previously.

When the improved algorithm is used, the pressure distribution is smooth across the interface and this is due to the use of the revised  $\alpha_{x'}$  factor for the GPA nodes at the interface. The improved algorithm is also invariant with respect to the selection of the system axes, and the identical result would be obtained for any orientation of the system axes.

In Figure 10 the interfaces are represented by the contract/sliding algorithms illustrated in the lower centre portion of Figure 6, and 7. The previous rigid algorithm result indicates a significantly reduced pressure at the interface GPA node. This is similar to the results for the attached interface shown in Figure 9, and it also occurs because the force on the GPA interface node is too large for the pressure (without the boundary correction on the  $\alpha_x$  factor).

For the improved rigid algorithm and the improved flexible algorithm, the pressure distribution is smooth across the interface and this again is due to the use of the revised  $\alpha_{x'}$  factor. The improved rigid algorithm is shown on the left side of Figure 7 and described by Equation (24). The improved flexible algorithm is shown on the right side of Figure 7 and described by Equations (25)–(34). For this problem the results are almost identical for the improved rigid and improved flexible algorithms. Although it would appear that the flexible algorithm would be more physically realistic, for this problem there is no noticeable difference between the two improved algorithms.

All of the results in Figure 10 show a slightly increased pressure designated as a reflection from the interface. This probably occurs because there is not a uniform node spacing and mass distribution at the interface. The nodal spacing in the initial geometry (for both GPA nodes and finite elements) is the nodal diameter,  $d_0$ . At the interface, however, the spacing between the GPA interface node, and the finite element surface, is reduced to  $d_0/2$ . Also, the nodal masses on the surface of the finite element grid are only one-half as much as the GPA nodes and interior finite element nodes. This mismatch in nodal mass and spacing is probably the cause of this rather small effect. For comparison, the attached algorithm results in Figure 9 do not having as pronounced a reflection and this is probably because the nodal masses and spacing are uniform across the interface.

The final wave propagation example in Figure 11 is for the particle contact (with different material) algorithm as shown in Figure 8 and the lower right portion of Figure 6. It is described by Equations (35)–(48). Here the GPA nodes on both sides of the interface have identical water properties, but they are designated as different material numbers. The interactions between the interface nodes are through the contact algorithm only. The results in Figure 11 are for the nodal arrangement shown in the lower right portion of Figure 6. The results are similar when one side of the interface is shifted a distance of  $d_0/2$  in the  $z'$  direction such that each interface node contacts two nodes rather than only one. The lower portion of Figure 11 shows the response when a bond viscosity is used. This also includes the viscous interface force from Equation (40). The results are very similar to those obtained with the nodal viscosity.

### 5.2. Cylinder impact with plastic flow

The next examples are shown in Figure 12. These axisymmetric cylinder impact computations include large plastic flow deformations. The initial length and diameter of the cylinder are  $L_0 = 15.2$  mm and  $D_0 = 10.2$  mm, and the material is Armco Iron. The impact velocity is 305 m/s and the nodes on the rigid surface are not allowed to lift off the surface. All of the GPA computations use ghost (fictitious mirror image) nodes on the centreline  $z$ -axis and the rigid surface to enforce zero displacement boundary conditions. The constitutive model and constants for the Armco Iron are as provided by Johnson and Cook [23].

The outlines of the deformed cylinder and the plastic strain contours on the left side of the cylinders come from a finely grided (1728 triangular elements) finite element computation. The

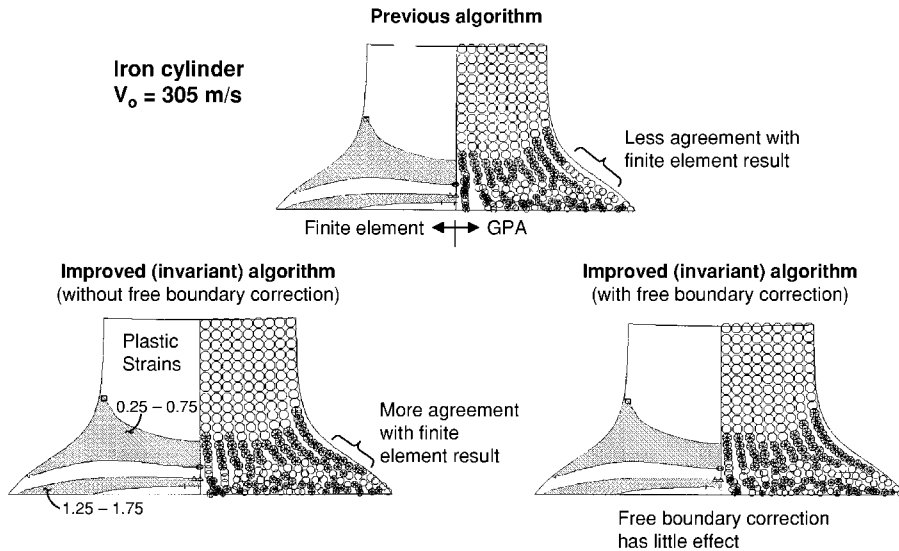


Figure 12. Effect of the improved (Invariant) algorithm for a cylinder impact problem.

darkened GPA nodes are for plastic strain regions of 0.25–0.75 and 1.25–1.75. These regions correspond to the shaded regions of the finite element computation. The GPA computations use nodal viscosity. The effects of other viscosity approaches are presented elsewhere [12, 17].

All three of the comparisons in Figure 12 provide good general agreement between the GPA and finite element solutions. The deformed lengths and maximum diameters are essentially identical for all of the results. The plastic strain distributions are in general agreement but there are some subtle differences. For the previous algorithm results at the top of Figure 12 there are several GPA nodes on the surface that have plastic strains in the range of 0.75–1.25. These are noted as having less agreement with the finite element results because the finite element plastic strains are in the range of 0.25–0.75. The normal to these surface GPA nodes is at an angle of approximately  $\pi/4$  rad to the system axes and this is an orientation that introduces maximum errors as previously indicated in Figure 3.

In the lower left portion of Figure 12 the GPA result is obtained from the improved (invariant) algorithm presented previously. Here there is no special treatment for the surface GPA nodes, but the improved algorithm chooses the local  $x'-z'$  axes to be aligned with the surface such that the errors are minimized. The GPA surface nodes that were noted to have incorrect plastic strains for the previous algorithm, are now in the correct range of 0.25–0.75 for the improved invariant algorithm.

The GPA results in the lower right portion of Figure 12 use both the improved (invariant) algorithm, as well as the boundary correction (modified  $\alpha_{x'}$ ). Here, the correction on the free boundary makes very little difference because the normal forces on the surface are small, whether they are corrected or not corrected. For this problem the boundaries remain intact and it is straightforward to identify and mark the surface nodes (from the  $x'$  outward normal) in the initial geometry, and to then apply the boundary correction throughout the duration of the problems. For other high distortion problems, however, it is not possible to clearly

designate the surface nodes for the duration of the response. Fortunately, for free boundaries the boundary correction makes little difference and is not needed.

### 5.3. Hypervelocity impact

Figure 13 shows hypervelocity impact computations using GPA nodes and finite elements. The problem consists of a 2024-T351 aluminum projectile impacting an Armco iron target at 4000 m/s. The projectile has a diameter of 1.27 mm and a thickness of 0.64 mm, and the target has a diameter of 102 mm and a thickness of 25.4 mm. Again, the constitutive model

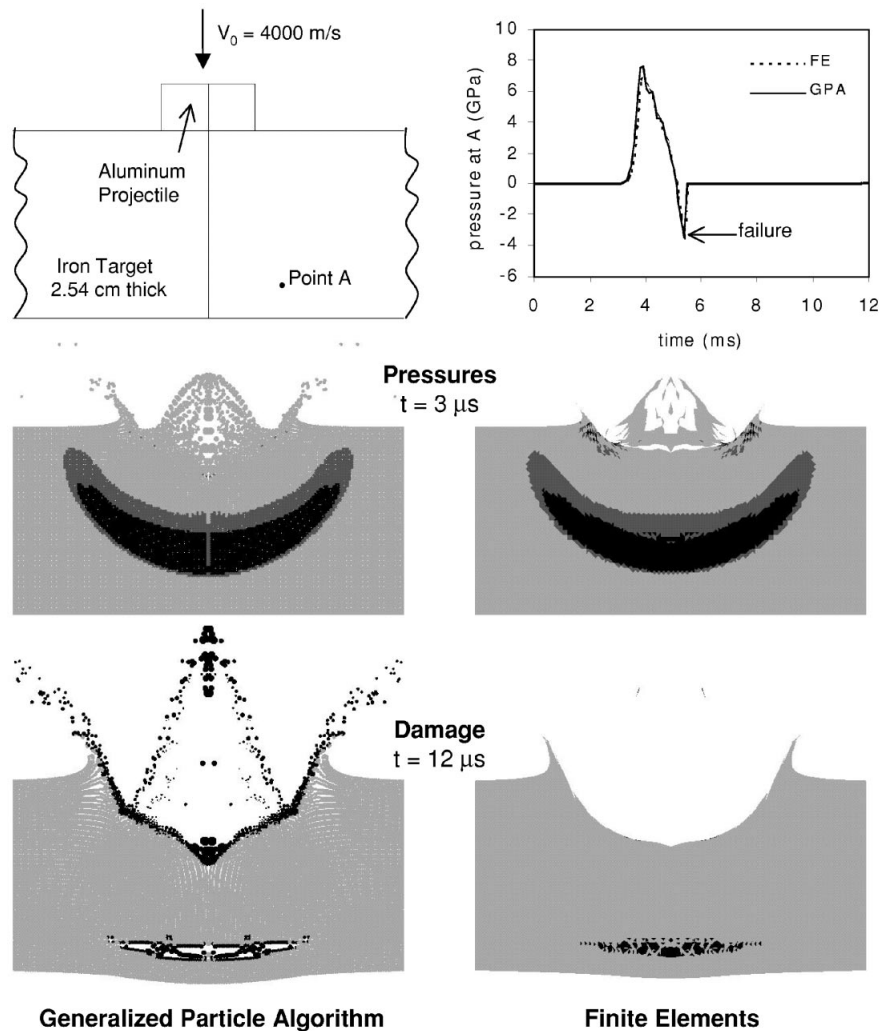


Figure 13. Hypervelocity impact computations.

and constants are presented in Reference [23]. This problem also computes damage for the spall failure, and the damage model and constants are provided by Johnson and Cook [24].

The GPA computation uses the particle contact interface algorithm as shown in Figure 8 and the lower right portion of Figure 6. The artificial viscosity coefficients are  $C_L = 0.2$  and  $C_Q = 4.0$ . Half of the viscosity is a nodal viscosity and half is a bond viscosity [17]. For this problem it is necessary to include some bond viscosity to maintain a stable solution. This same viscosity treatment is also used for the remaining examples. This example, and the remaining examples, use the improved (invariant) GPA algorithm (without the free boundary correction). The free boundary correction was shown to have little affect in the lower right of Figure 12, and it is very difficult to determine which nodes are free boundary nodes for the high distortion/debris response in Figure 13.

The finite element solution uses triangular elements (arranged in a pattern of four triangles within a quad to reduce volumetric locking), a sliding interface algorithm that is automatic, robust and symmetric, and an associated erosion algorithm [19, 20]. The erosion algorithm removes highly distorted elements on the sliding interfaces (when the equivalent plastic strains exceed 2.5) such that an adequate integration time increment and numerical stability can be maintained. When an element is eroded it cannot develop any stresses; it essentially disappears except that the mass is retained at the nodes. The sliding interfaces are updated in an automatic manner when an element is eroded.

Pressure distributions are shown for both approaches at  $t = 3.0 \mu\text{s}$ . The black regions represent pressures above 6.0 GPa, the medium gray regions represent pressures in the range of 3.0–6.0 GPa, and the light gray regions represent pressures under 3.0 GPa. The pressure distributions are almost identical. For the finite element solution some of the elements in the projectile, and on the top surface of the target, have eroded.

Damage distributions are shown at  $t = 12.0 \mu\text{s}$ , where the black regions represent fully damaged material ( $D = 1.0$ ). Both approaches show similar damaged spall regions near the bottom of the target. Again, for the finite element solution, all of the projectile elements, and some of the target elements in the crater, have eroded. The GPA solution retains all of the material, and it can be seen that the highly distorted, ejected material is fully damaged.

The upper right portion of Figure 13 shows pressure versus time for the GPA node and finite element at point *A*, which is in the spall region. The responses are essentially identical. This includes the arrival times, magnitudes and shapes of both the compressive waves and tensile reflections, as well as the tensile spall failure. For this problem the GPA solution provides an accurate reflection off the free bottom surface, and it does not give any indication of a tensile instability.

#### 5.4. Explosive detonation in concrete

Figure 14 shows computations for a detonated explosive that is buried in a concrete target. The concrete target is semi-infinite and is modelled by a cylinder with a height of 2.0 m and a diameter of 4.0 m. Non-reflective elements [19] are used on the side and bottom boundaries to represent the extended concrete target. The TNT explosive is a cylinder with a height and diameter of 15.2 cm, and the top surface is 15.2 cm below the top surface of the concrete. The JWL Equation of State is used for the explosive [19], and it is detonated at the centre of the charge. The Holmquist–Johnson–Cook model is used for the concrete [25].



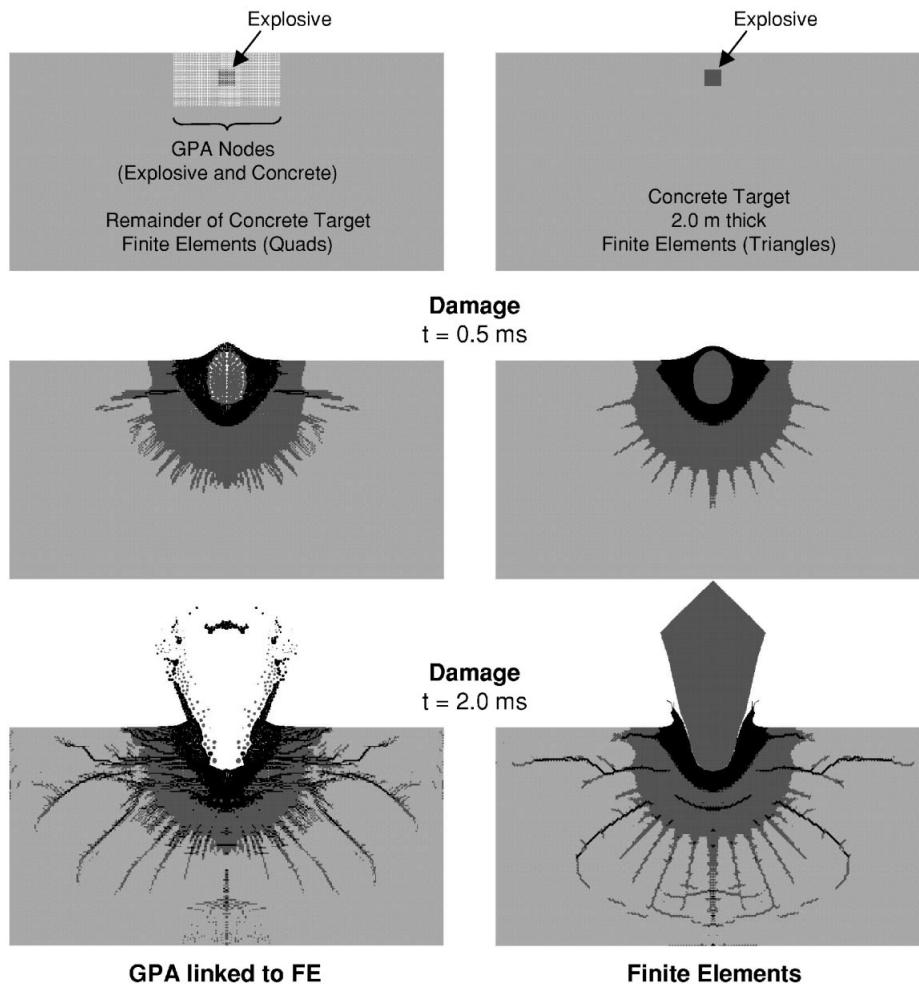


Figure 14. Buried explosive computations.

A uniform grid size is used throughout the explosive and the concrete. Each triangular element, quad element, and GPA particle node has a cross-sectional area of  $1.61 \text{ cm}^2$ . For the particle computations the GPA nodes in the top centre of the target represent a region with a height of  $0.5 \text{ m}$  and a diameter of  $1.0 \text{ m}$ . They are attached to the quad elements as shown in the lower left portion of Figure 6. Limiting the GPA nodes to the highly distorted region of the problems has two advantages. First, it reduces the CPU time by minimizing the number of GPA nodes in the problem (as they require more CPU time than finite elements). Second, the far field transient response can generally be modelled more accurately with finite elements.

For this problem the particle contact algorithm was not used for the explosive–concrete interface because the explosive GPA nodes tended to penetrate into the concrete. The lack

of a special interface algorithm probably gives some localized errors at this interface, but it does not significantly affect the propagation of the shock front.

At  $t = 0.5$  ms the explosive has expanded significantly, but it has not yet broken through the surface. The high damage regions in the concrete target (black shading) represent damage in the range of  $0.9 \leq D \leq 1.0$ , and the lower damage regions (dark gray shading) are for  $0.1 \leq D \leq 0.9$ . Both approaches give seemingly similar results at this stage.

At  $t = 2.0$  ms the explosive has broken through the concrete. For the finite element approach the highly distorted concrete above the explosive has eroded (based on an equivalent plastic strain of 2.5), although the mass of the eroded elements remains to retard the expansion of the explosive. For the GPA approach the explosive particles simply break through the concrete particles. For this problem the freely expanding explosive GPA nodes are limited to a maximum volumetric strain of 10.0. This provides for a more efficient solution and has a negligible affect on the results.

Although the damage distributions at  $t = 2.0$   $\mu$ s are not identical, they are very similar. Considering the many differences between the two computational approaches, the similarity of the results tends to indicate that both approaches are providing results with good numerical accuracy. The GPA solution has a GPA interface between the explosive and the concrete, an attached GPA-finite element interface in the concrete, and quad finite elements in the remainder of the target. The finite element solution has a sliding interface (with erosion) between the explosive and the concrete, and triangular finite elements.

### 5.5. Fluid–structure interaction

The next example is shown in Figure 15 and it demonstrates the capability to perform fluid–structure interaction computations. These computations are performed in plane strain geometry. A slab of water (76 cm width  $\times$  51 cm height) rests on a 4340 steel plate (5.1 cm thickness). The models for the water and steel are given in References [21, 23]. A thin layer of Detasheet explosive (12.7 cm width  $\times$  1.27 cm thickness) is inserted into the water at a height of 21 cm above the top of the steel plate. All of the water and explosive finite elements (triangles), and all of the water GPA nodes, have a cross-section area of 0.10 cm<sup>2</sup>. The explosive GPA nodes are smaller to adequately represent the thin layer of explosive. They have a cross-sectional area of 0.025 cm<sup>2</sup>.

The GPA computation, on the left side of Figure 15, has GPA explosive nodes surrounded by a region (30.5 cm  $\times$  21.6 cm) of GPA water nodes. As was the case with the previous example in Figure 14, there is no special interface treatment between the explosive and the surrounding material. The interface between the water GPA nodes and the water finite elements uses the flexible particle contact algorithm shown on the right side of Figure 7. The finite element sliding interface algorithm [19, 20] is used for the interaction between the water and the steel plate.

The finite element computation uses the sliding interface algorithm [19, 20] for the explosive–water interface and the water–steel plate interface. For this problem there is no erosion in either the water or explosive. Because these materials have no strength, there is no resistance to the deformation and the equivalent strains continue to grow, which would lead to excessive erosion.

At  $t = 0.1$  ms the pressure distributions for the two approaches are almost identical. The black regions represent pressures above 200 MPa, the medium gray regions represent pressures

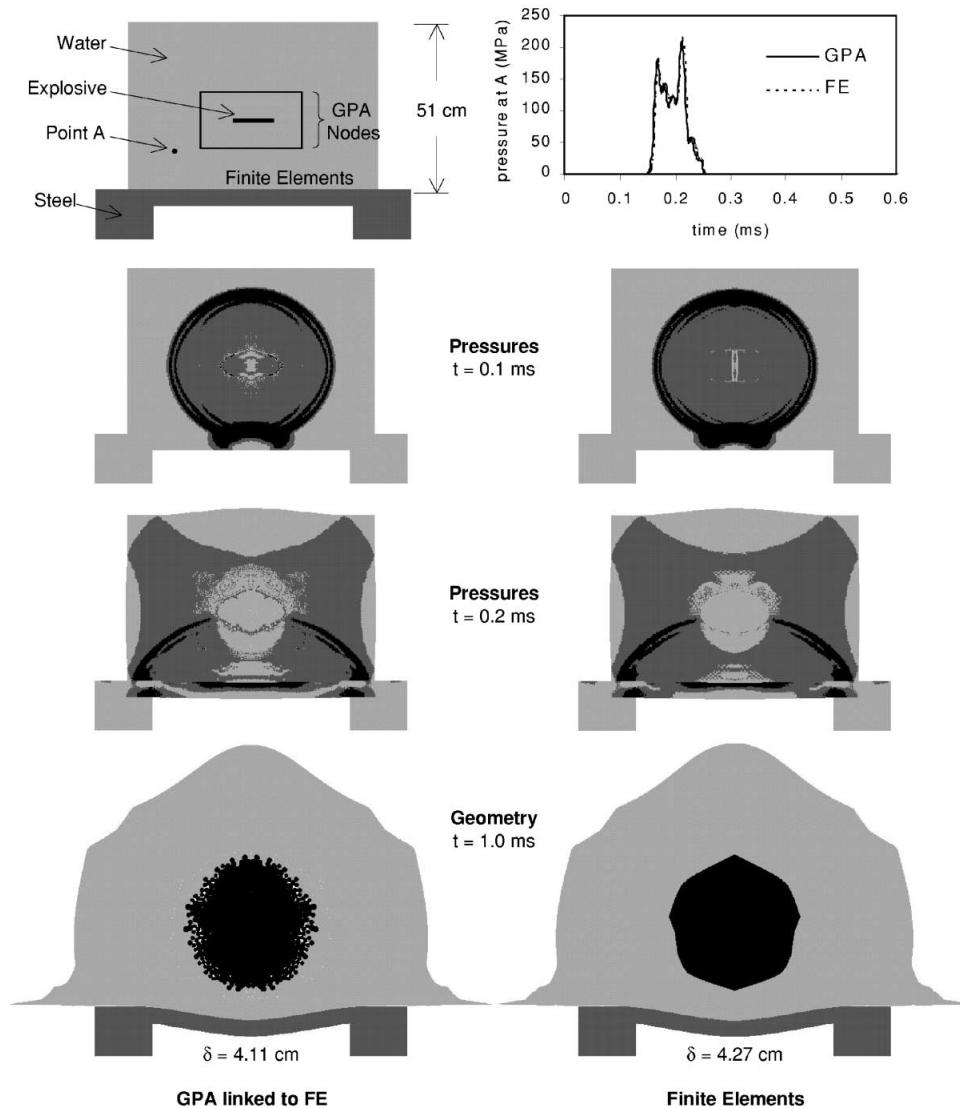


Figure 15. Explosive-water-metal interaction computations.

in the range of 50–200 MPa, and the light gray regions represent pressures under 50 MPa. At  $t = 0.2$  ms the shock waves have reflected off the boundaries and interfaces, and again the pressure distributions are almost identical. At a much later time of  $t = 1.0$  ms the geometric shapes are shown to be very similar. The plate deflection for the GPA computation is 4.11 cm and the deflection for the finite element computation is 4.27 cm (a difference of only 3.8 per cent).

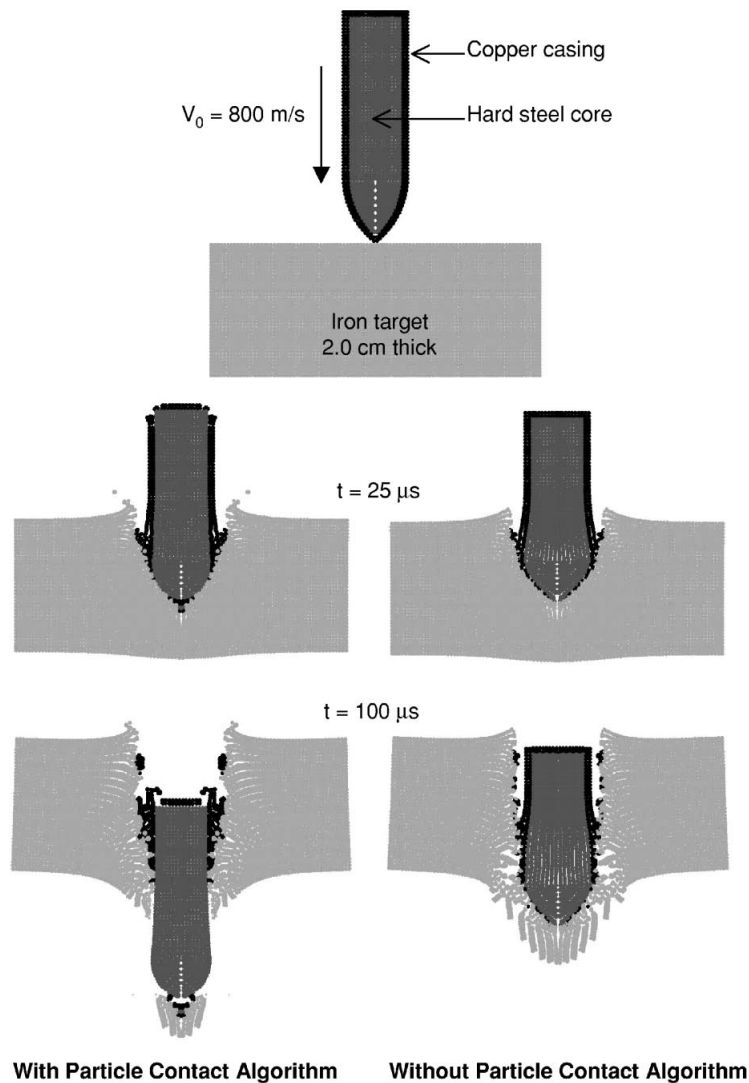


Figure 16. Projectile penetration computations with and without the particle contact algorithm.

The upper right portion of Figure 15 shows pressures versus time at point *A*, for both of the computations. The first peak is from the expanding pressure wave and the second peak occurs when the reflected wave from the steel plate reaches point *A*. Once again there is excellent agreement between the two approaches. For the GPA computation the shock wave must travel through an explosive-water interface (between GPA nodes), an interface between water GPA nodes and water finite elements, and a fluid-structure interface between water finite elements

and steel finite elements. Because fluids do not develop tensile stresses there is no concern for tensile instabilities in the GPA nodes.

### 5.6. Projectile penetration with contact/sliding interfaces

The final example is shown in Figure 16. It involves a projectile (1.0 cm diameter  $\times$  3.5 cm length) impacting an Armco iron target (2.0 cm thickness). The hard steel core (S-7 tool steel) is covered with a copper casing. The problem is composed entirely of GPA nodes. The strength and fracture models are given in References [23, 24].

The computation on the left side of Figure 16 uses the particle contact algorithm shown in Figure 8. Here the copper casing can slide and/or become detached from the steel core. When this algorithm is not used, as shown on the right side of Figure 16, the three materials tend to stick together and the response is much different. These computations are shown to only illustrate the capability for sliding and detachment. The accuracy of this capability, for a variety of applications, will be evaluated in the future.

## 6. SUMMARY AND CONCLUSIONS

This paper has presented an improved GPA algorithm that is invariant with respect to the system co-ordinates, and which improves the accuracy at boundaries and interfaces. Several interface algorithms are also presented. Included are an attached interface algorithm (where GPA nodes are attached to a finite element grid), two contact/sliding interface algorithms (where a flexible or rigid GPA node interacts with the surface of a finite element grid), and a particle contact algorithm (where GPA nodes of different materials interact with one another). A wide range of example computations (with GPA nodes only and with GPA nodes linked to finite elements) show excellent agreement with analytic and finite element results.

## ACKNOWLEDGEMENTS

The work conducted by Alliant Techsystems was funded by Contract DACA39-98-K-0015 from the U.S. Army Engineer Research and Development Center (ERDC) at Vicksburg, MS. The authors also appreciate the contributions of J. D. Cargile and S. A. Akers at ERDC.

## REFERENCES

1. Johnson GR, Beissel SR, Stryk RA. A generalized particle algorithm for high velocity impact computations. *Computational Mechanics* 2000; **25**(2/3):245–256.
2. *Computer Methods in Applied Mechanics and Engineering*. vol. 139. 1996.
3. *Computational Mechanics*. vol. 25. 2000.
4. Lucy LB. A numerical approach to the testing of fusion process. *The Astronomical Journal* 1977; **88**:1013–1024.
5. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices Royal Astronomy Society* 1977; **181**:375–389.
6. Benz W. Smooth particle hydrodynamics: a review. Harvard-Smithsonian Center for Astrophysics (Preprint 2884), 1989.
7. Monaghan JJ. Smoothed particle hydrodynamics. *Annual Review Astronomics and Astrophysics* 1992; **30**: 543–574.
8. Libersky LD, Petschek AG. Smooth particle hydrodynamics with strength of materials. *Advances in the Free Lagrange Method, Lecture Notes in Physics*, vol. 395. Springer: Berlin, 1990, pp. 248–257.
9. Johnson GR, Beissel SR. Normalized smoothing functions for SPH impact computations. *International Journal for Numerical Methods in Engineering* 1996; **39**:2725–2741.

10. Johnson GR. Status of EPIC codes, material characterization and new computing concepts at Honeywell. *Computational Aspects of Penetration Mechanics, Lecture Notes in Engineering*, vol. 3, Springer: Berlin, 1983; pp. 24–35.
11. Johnson GR, Stryk RA, Dodd JG. Dynamic Lagrangian computations for solids with variable connectivity for severe distortions. *International Journal for Numerical Methods in Engineering* 1986; **23**:509–522.
12. Johnson GR, Stryk RA, Beissel SR. SPH for high velocity impact computations. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:347–373.
13. Swegle JW, Attaway SW. On the feasibility of using smoothed particle hydrodynamics for underwater explosion calculations. *Computational Mechanics* 1995; **17**:151–168.
14. Swegle JW, Attaway SW, Heinstein MW, Mello FJ, Hicks DL. An analysis of smoothed particle hydrodynamics. SAND93–2513, Sandia National Laboratories, Albuquerque, NM, March 1994.
15. Johnson GR. Dynamic response of axisymmetric solids subjected to impact and spin. *AIAA Journal* 1979; **17**(9):975–979.
16. Monaghan JJ, Gingold RA. Shock simulation by the particle method SPH. *Journal of Computational Physics* 1983; **52**:374–389.
17. Johnson GR. Artificial viscosity effects for SPH impact computations. *International Journal of Impact Engineering* 1996; **18**:477–488.
18. Randles PW, Libersky LD. Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:375–408.
19. Johnson GR, Stryk RA, Holmquist TJ, Beissel SR. Numerical algorithms in a Lagrangian hydrocode. *Report WL-TR-1997-7039*, Wright Laboratory, U.S. Air Force, July 1997.
20. Johnson GR, Stryk RA. Symmetric contact and sliding interface algorithms for intense impulsive loading computations. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:4531–4549.
21. Huang YC, Hammitt FG, Yang WJ. Hydrodynamic phenomena during high speed collision between liquid droplet and rigid plane. *Journal of Fluids Engineering, ASME*, 1973; **95**(2):276–294.
22. Johnson GR. Liquid–solid impact calculations with triangular elements. *Journal of Fluids Engineering, ASME* 1977; **99**(3):598–600.
23. Johnson GR, Cook WH. A constitutive model and data for metals subjected to large strains, high strain rates, and high temperatures. *Proceedings of Seventh International Symposium on Ballistics*, The Hague, The Netherlands, 1983; 541–547.
24. Johnson GR, Cook WH. Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures. *Engineering Fracture Mechanics* 1985; **21**:31–48.
25. Holmquist TJ, Johnson GR, Cook WH. A computational model for concrete subjected to large strains, high strain rates and high pressures. *Proceedings of Fourteenth International Symposium in Ballistics*, Quebec City, Canada, 1993.