INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **42**:1189–1210 (DOI: 10.1002/fld.578)

A three-dimensional contact algorithm for sliding surfaces

A. S. Dawes*,†

Computational Physics Group, Department of Physics Design, AWE plc, Aldermaston, Berkshire, U.K.

SUMMARY

This paper describes a model for sliding contact between materials. This has been embodied in a contact algorithm which has been incorporated into a three-dimensional, time-dependent, staggered mesh, finite element Lagrangian code. We first present an overview of the code and then describe the model underpinning the contact algorithm. The stages of the algorithm are then described and finally, results are presented for a number of different problems. Copyright © 2003 John Wiley & Sons, Ltd.

1. INTRODUCTION

The physical interaction of adjacent materials at an interface is an important aspect of dynamical phenomena. For materials which move over each other shearing can occur if the relative motion is non-zero. The effect of shear is to deform the materials producing strain and internal stresses. The coupling between the stress and strain is dependent on the material properties. For a fluid the stress is proportional to the strain rate while for a perfectly elastic solid (say a metal) the stress is proportional to the strain (Hooke's law). For fluids, the stress can only be supported over a very small localized region, such as within a boundary layer [1]. Within the layer energy is dissipated through the action of viscosity. Solids can support stresses, over a larger region, because of their inherent physical strength. However, the strength of most metals is strain rate dependent. In many cases highly strain rate dependent. Solids that slide over each other will experience a frictional force tending to dissipate the kinetic and elastic distortional energy. The frictional coupling will also produce different levels of stress within each material and the magnitudes will be dependent on their respective strengths. As the level of the strain rate can effect the strength of a material the dynamics of sliding contact can be extremely complex. For small or moderate stresses, energy will be stored as recoverable elastic distortional energy; if the materials are allowed to slide back to their original positions this energy will be recovered. For increased stress levels one or both of the solids may reach their elastic limit, and yield, causing some of the energy to be dissipated as plastic work [2]. For extremely high stress levels many solid materials will behave like a fluid, and the fluid approximation is sufficient to describe their physical behaviour.

---

*Correspondence to: A. S. Dawes, Computational Physics Group, Department of Physics Design, AWE plc, Aldermaston RG74PR, Berkshire, U.K.
†E-mail: alan.dawes@awe.co.uk

Received 19 October 2002
Revised 20 March 2003

Within the computational physics group at AWE, dynamical problems are modelled using codes developed upon either the Lagrangian or Eulerian numerical methods. Both have their advantages and disadvantages. For the Lagrangian approach the mesh follows the flow field with material interfaces captured as part of the mesh. However, for highly distorted flows the mesh will tangle causing the numerical method to breakdown. For the Eulerian approach this does not happen because the mesh remains fixed. However, the method tends to diffuse results and smear material interfaces unless they are captured. For this we use the volume of fluid (VOF) technique based on Youngs [3]. A comprehensive review of this technique can be found in the paper by Rider and Kothe [4].

One way to overcome the Lagrangian mesh distortion is to remap the solution over to a regular mesh just before the distortion becomes too great. The calculation is then continued using an Eulerian code. However, this requires some form of human intervention, to monitor the degree of mesh distortion, and judge the appropriate point to remap. This has been the traditional approach but recently `CORVUS` [5, 6] has been developed in two-dimensions (2D) and `PEGASUS` [7, 8] in three-dimensions (3D) which both embrace the Arbitrary Lagrangian Eulerian (ALE) philosophy. This technique exploits the advantages of the Lagrangian and Eulerian methods while maintaining a smooth mesh topology using a dynamic mesh re-positioning strategy. A general review of this approach can be found in the paper by Benson [9].

The initial dynamics of the material interfaces (as they start to move) are important, and the pure Lagrangian approach is the natural choice to model this behaviour. Generally, adjacent materials in a Lagrangian mesh share a common numerical interface. However, as the flow field develops un-natural shearing can develop because of the rigidity of the mesh coupled with the differences in adjacent material properties. For materials which are known to slide over each other numerical shear is physically inaccurate. In this paper we describe a model for sliding contact that overcomes this limitation. In this paper we do not consider the tangential interaction, such as at a boundary layer or frictional interaction.

The implementation of the contact algorithm into the three-dimension code `PEGASUS` was briefly outlined earlier by the author [10] and was developed from the master and slave approach adopted in `CORVUS` [6]. The slide algorithm in `CORVUS` has some similarities to that used in the `TENSOR` code [11] but it does not use the notion of *phony* elements. The slide model described in this paper has similarities to the method developed by Wilkins [12]. However, it defers in a number of respects. First, it has been developed around the finite element approach used in `PEGASUS`. Second, it introduces the idea of an *effective* pressure force. Third, our model has been developed for purposes of modularity because `PEGASUS` has followed a modular development.

In this article we first present an overview of the code and then describe the model underpinning the contact algorithm. The stages of the algorithm are then described and finally, results are presented for a number of different problems.

2. HYDROCODE OVERVIEW

The governing equations to be solved in a Lagrangian frame are respectively conservation of mass, momentum and energy thus,

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} + \rho\nabla\cdot\underline{u} = 0 \tag{2.1}$$

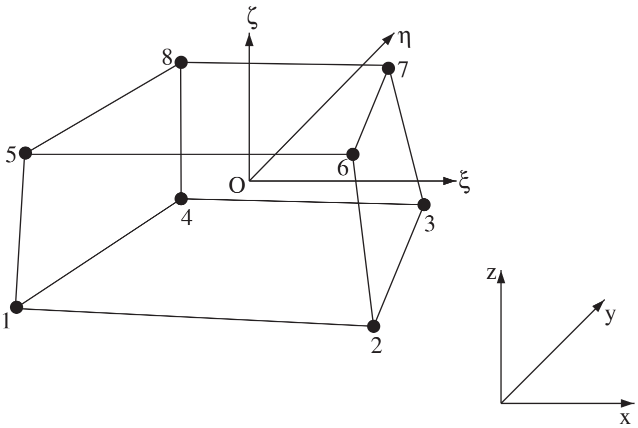

Figure 1. Hexahedral element definition.

$$\rho \frac{\mathrm{d}\underline{u}}{\mathrm{d}t} = \nabla \cdot \underline{\underline{\sigma}} \tag{2.2}$$

$$\rho \frac{\mathrm{d}\varepsilon}{\mathrm{d}t} = -(\underline{\underline{\sigma}} \cdot \nabla) \cdot \underline{u} \tag{2.3}$$

where ρ is the density, $\underline{u}$ is the velocity vector, ε is the specific internal energy and $\underline{\underline{\sigma}}$ is the stress tensor. As described above, for extremely high stress levels many solid materials behave approximately like a fluid. However, at lower stress levels, a solids ability to resist distortion becomes an important factor, and the material response cannot be characterised by an hydrodynamic equation of state (EoS) alone. It therefore becomes necessary to include the mechanical behaviour of the solid material in the modelling. To do this the stress tensor has been defined as the sum of the hydrostatic pressure p and the deviatoric stress tensor $\underline{\underline{s}} = -p\underline{\underline{i}} + \underline{\underline{s}}$, where $\underline{\underline{i}}$ is the unit tensor. For closure an EOS for the pressure p in terms of the other thermodynamic variables (usually ρ and ε) must also be supplied, as well as differential equations for the incremental changes in the deviatoric stress components and additional incremental energy changes due to elastic distortion and plastic work. The form of the elastic–plastic model employed is similar to the approach developed by Wilkins [2].

The numerical approach was based on a staggered mesh using a 3D Cartesian computational grid with co-ordinates (x, y, z). Each element is hexahedral with eight vertices and the ordering of these nodes within each element are illustrated in Figure 1. The three velocity components (u, v, w) of the vector $\underline{u}$ are stored at the vertices while pressure, density, specific internal energy and (if present) deviatoric stresses are stored at the element centres. The mesh is divided into a number of blocks which are arranged to have structured rectangular connectivity between elements within a block but between blocks the connectivity is unstructured. The momentum, energy and (if present) deviatoric stress equations are solved using a Petrov–Galerkin weak integral finite element formulation (employing mass lumping) using an iso-parametric co-ordinate frame $\underline{\xi} = [\xi, \eta, \zeta]^{\mathrm{T}}$. In the new frame the element becomes a unit cube. The original Cartesian co-ordinate vector $\underline{x}$ is related to the iso-parametric co-ordinates

Table I. Iso-parametric nodal values.

k	ξ_k	η_k	ζ_k
1	$-1/2$	$-1/2$	$-1/2$
2	$1/2$	$-1/2$	$-1/2$
3	$1/2$	$1/2$	$-1/2$
4	$-1/2$	$1/2$	$-1/2$
5	$-1/2$	$-1/2$	$1/2$
6	$1/2$	$-1/2$	$1/2$
7	$1/2$	$1/2$	$1/2$
8	$-1/2$	$1/2$	$1/2$

through the relationship,

$$\underline{x} = \sum_{k=1}^{8} N_k(\underline{\xi})\underline{x}_k \tag{2.4}$$

where $\xi \in [-\frac{1}{2}, \frac{1}{2}]$, $\eta \in [-\frac{1}{2}, \frac{1}{2}]$, $\zeta \in [-\frac{1}{2}, \frac{1}{2}]$ and $N_k(\underline{\xi})$ is the basis function for the kth node in the element defined as

$$N_k(\underline{\xi}) = \tfrac{1}{8}(1 + 4\xi_k\xi)(1 + 4\eta_k\eta)(1 + 4\zeta_k\zeta) \tag{2.5}$$

where the vertice values (ξ_k, η_k, ζ_k) are defined in Table I. The velocity vector within an element is then just the variation of the velocities between the nodes thus,

$$\underline{u} = \sum_{k=1}^{8} N_k(\underline{\xi})\underline{u}_k \tag{2.6}$$

because the iso-parametric co-ordinates are time in-dependent.

The numerical solution method was based upon a predictor/corrector strategy and using an artificial viscosity term (added to the pressure) to maintain numerical stability about shock waves. To advance the solution at the ith node, jth element and at the nth time level to the next time level $n+1$ there are five stages.

1. The half time step node positions are calculated $\underline{x}_i^{n+1/2} = \underline{x}_i^n + \underline{u}_i^n \Delta t/2$, where the subscript refers to the node number and superscript refers to the time step level. At this stage no nodes are moved.
2. The finite element equations are solved during the *Predictor* stage at the half time step for density ρ, pressure p, specific internal energy ε, and (if present) deviatoric components for each element j and the accelerations at each node $\underline{\dot{u}}_i$.
3. The full time step velocities are calculated based on the nodal accelerations $\underline{u}_i^{n+1} = \underline{u}_i^n + \underline{\dot{u}}_i \Delta t$.
4. The nodes are moved to their new full time step positions $\underline{x}_i^{n+1} = \underline{x}_i^n + \frac{1}{2}(\underline{u}_i^n + \underline{u}_i^{n+1})\Delta t$.
5. The finite element equations are solved during the *Corrector* stage at the full time step for density ρ, pressure p, specific internal energy ε, and (if present) deviatoric stress components and associated energy changes for each element j.

This cycle is repeated until the desired temporal position is reached.

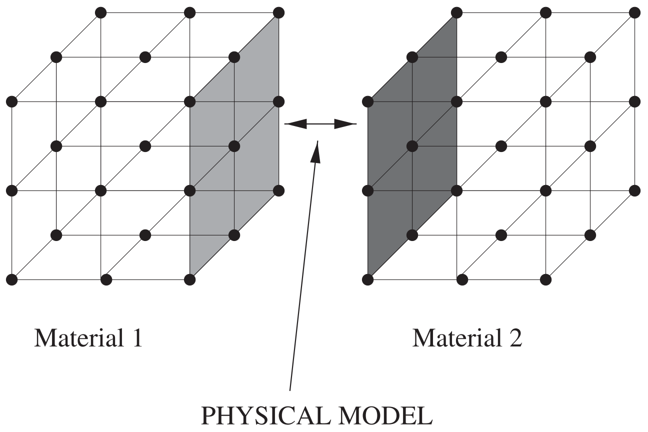

Figure 2. Split mesh definition.

3. MODEL FOR SURFACE INTERACTION

Consider for the moment a single mesh containing two different adjacent materials with the interface between them sharing the same set of nodes. The two materials will in general have different physical properties and as they distort, say through some sideways impulse, they will tend to slide or shear. However, slide will be impossible because they share a common set of nodes. This coupling will cause unnatural shearing. To overcome this the mesh can be split up into two separate regions, illustrated in Figure 2, each with its own set of nodes on the surface. The slide algorithm then defines the forces at each node and on each surface based on the interaction between them. We now go onto develop the model for the surface interaction.

3.1. Sliding interfaces

The *Rankine Hugoniot* relations (see for example, the book by Hirsch [13]) can be used to show that at an interface the normal stress σ_n and normal velocity u_n components must be continuous,

$$[\sigma_n]=0, \quad [u_n]=0 \tag{3.1}$$

while the tangential velocity u_t is generally discontinuous,

$$[u_t]\neq 0 \tag{3.2}$$

From these conditions any physical model for sliding contact must guarantee that Equation (3.1) is satisfied.

4. CONTACT ALGORITHM

For our purposes all materials have clearly defined boundaries and at some internal boundaries materials are allowed to slide over each other; it is assumed that there is no friction between them. Each sliding surface must be chosen to be either a master or slave. The master surface is chosen to be either the one which is the strongest or the most dense, or failing that, the one driving the flow. In this way, it is the slave surface which conforms most to the master.

The method was based on a predictor/corrector philosophy. During the *Predictor* stage nodal accelerations on the slide surfaces are calculated (see step 2 of the Lagrangian phase above) based on the assumption that they are free boundaries. The *Corrector* stage symmetrically modifies the normal accelerations, on both surfaces in contact, to account for the surface interaction. The new normal accelerations are vectorially added to the corresponding nodal unmodified tangential accelerations to give the new nodal accelerations. Nodes are then moved but numerical inaccuracies can cause the surfaces to penetrate. Therefore, the slave surface is asymmetrically *put-back-on* to the master surface and the slave normal velocity is forced to match the master; for normal velocity continuity.

To determine the normal forces of interaction, each surface must have knowledge of its opposite surface it is in contact with. For our purposes we shall use the terms *upper* and *lower* surface to represent respectively the surface where the forces are currently being computed and the surface which it is in contact. Generally nodal positions are not coincident. Therefore, the *lower* surface is where forces have to be interpolated to the point the *upper* point acts.

With the above modifications the original scheme now becomes:

1. The half time step node positions are calculated $\underline{x}_i^{n+1/2} = \underline{x}_i^n + \underline{u}_i^n \Delta t/2$, where the subscript refers to the node number and superscript refers to the time step level. At this stage no nodes are moved.
2. The finite element equations are solved during the *Predictor* stage at the half time step for density ρ, pressure p, specific internal energy ε, and (if present) deviatoric components for each element j and the accelerations at each node $\underline{\dot{u}}_i$. At a sliding surface it is assumed to have free boundary conditions.
   - The Contact Algorithm symmetrically updates the master and slave surface nodal accelerations due to the forces of interaction.
3. The full time step velocities are calculated thus $\underline{u}_i^{n+1} = \underline{u}_i^n + \underline{\dot{u}}_i \Delta t$.
4. The nodes are moved to their new full time step positions $\underline{x}_i^{n+1} = \underline{x}_i^n + \frac{1}{2}(\underline{u}_i^n + \underline{u}_i^{n+1})\Delta t$.
   - If penetration occurs the asymmetric put-back-on stage moves the slave nodal positions onto the adjacent master surface and adjusts the normal velocities to satisfy normal velocity continuity.
5. The finite element equations are solved during the *Corrector* stage at the full time step for density ρ, pressure p, specific internal energy ε, and (if present) deviatoric stress components and associated energy changes for each element j.

It is clear that there are two additional stages which only apply to the nodes on the sliding surfaces. We now go onto fully describe the stages of the contact algorithm.

4.1. Search procedure

Each surface is represented by a set of nodes and connectivity to their associated elements. The initial search will determine the connectivity between nodes on the *upper* surface to those on the *lower* surface. The search will be symmetric and will be the most computationally expensive because the connectivity has to be created. To determine this connectivity it is only necessary to determine the *lower* node which is closest to a given *upper* node because

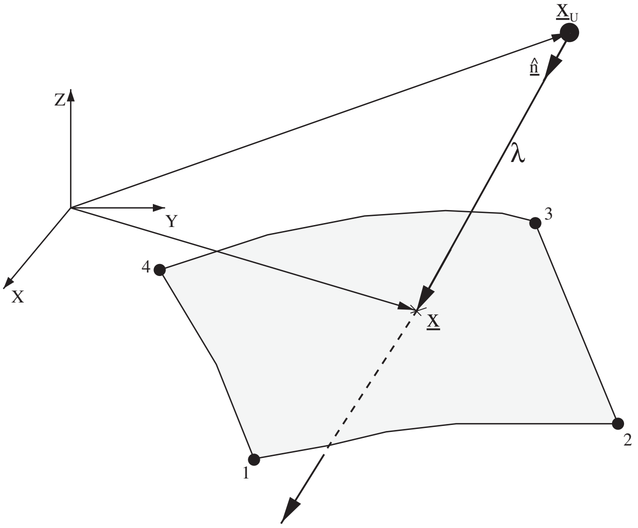

Figure 3. Normal and element face intersection.

the surfaces are assumed to be smooth and touching; we do not consider void opening or closure.

Once the *lower* node has been found the attached element faces are searched to determine the element face the projected normal vector penetrates, illustrated in Figure 3. To do this the surface intersection point must be determined based on the unit normal vector (see Section 4.1.1) and the distance λ to it from the *upper* node, with position vector $\underline{x}_u$, and the point (ξ,η) through which the unit normal vector $\hat{\underline{n}}$ passes. The position vector of the intersection point is given by

$$\underline{x}=\underline{x}_u+\lambda\hat{\underline{n}} \tag{4.1}$$

and by using the variation of the position vector in terms of the finite element basis functions the iso-parametric equation of the intersection point can be shown (see Appendix A) to have the form

$$\underline{x}_u+\lambda\hat{\underline{n}}=\underline{a}_0+\underline{a}_1\xi+\underline{a}_2\eta+\underline{a}_3\xi\eta \tag{4.2}$$

where $\underline{a}_0$, $\underline{a}_1$, $\underline{a}_2$ and $\underline{a}_3$ are vectors which are linear functions of the position vectors for the four nodes defining an element face. This is a vector equation for the three unknowns (ξ,η,λ) and can be solved using vector identities. The desired intersection point passes through the face satisfying the criteria $\xi\in[-\frac{1}{2},\frac{1}{2}]$ and $\eta\in[-\frac{1}{2},\frac{1}{2}]$. If the point found falls outside the region then the other element faces are searched until the desired element is found.

As part of the acceleration update stage(see Section 4.2) the connectivity information between the sliding surfaces must be kept up to date by checking it, at every time level, to determine whether it has changed. To do this, for a given *upper* node the intersection point on the *lower* surface is re-calculated and if it still lies within the *lower* element face, stored earlier, then it is not necessary to change the connectivity for that node. However, if this

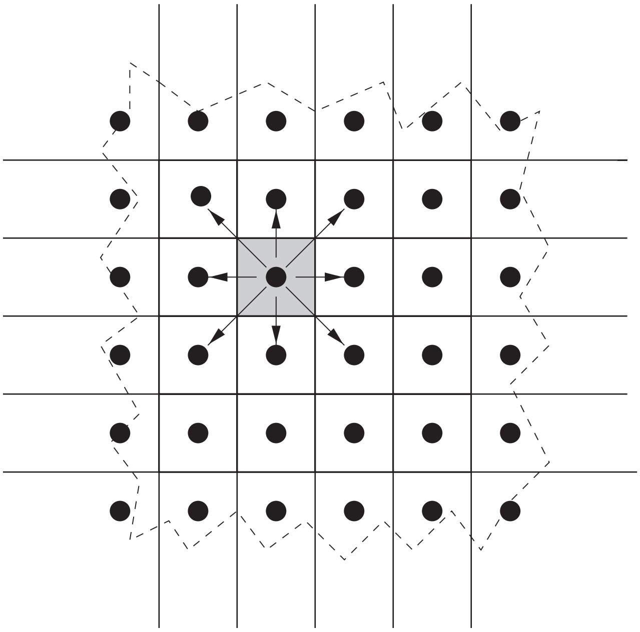

Figure 4. Eight possible search directions.

Table II. Movement criteria.

No.	$\xi > 1/2$	
1	$\eta < -1/2$	South East
2	$\eta \in [-1/2, 1/2]$	East
3	$\eta > 1/2$	North East
	$\xi < 1/2$	
4	$\eta < -1/2$	South West
5	$\eta \in [-1/2, 1/2]$	West
6	$\eta > 1/2$	North West
	$\xi \in [-1/2, 1/2]$	
7	$\eta < -1/2$	South
8	$\eta > 1/2$	North

is false the connectivity must be updated dynamically by searching outward from the stored element face. The search moves one element at a time with upto eight possible directions illustrated in Figure 4. The direction of motion is controlled by the calculated intersection point (ξ^*, η^*) on the *lower* element face and the criteria given in Table II. By moving to the adjacent element the direction of movement is recalculated and the search continues until the correct element face has been found.

4.1.1. Normal vector evaluation. Normal vectors are calculated at each node on the master and slave surfaces at the half time step positions. At each node there will be several elements attached to it. In general for hexahedral elements the normal vector calculated at the node in question will be different for the different elements attached to it. One has to calculate an *average* value $\underline{\hat{n}}$ and, for our purposes, was based on an area weight of the normal vectors from each of the attached elements calculated at the nodal position thus,

$$A\underline{\hat{n}} = \sum_{k=1}^{N} A_k \underline{\hat{n}}_k \tag{4.3}$$

where N is the number of elements attached to the node, A_k and $\underline{\hat{n}}_k$ are respectively the area and the unit normal vector associated with the kth element and A is the total area thus,

$$A = \left| \sum_{k=1}^{N} A_k \underline{\hat{n}}_k \right| \tag{4.4}$$

Furthermore, if the node lies on a boundary the normal vector is adjusted to satisfy the boundary condition.

4.2. Symmetric force update

The accelerations of all nodes on the sliding surfaces are initially calculated based on the internal body forces and free surface boundary condition. For a given node on the *upper* and *lower* surfaces momentum conservation requires that,

$$M_\mathrm{u}\underline{\dot{u}}_\mathrm{u} = \underline{F}_\mathrm{u}, \quad M_\mathrm{l}\underline{\dot{u}}_\mathrm{l} = \underline{F}_\mathrm{l} \tag{4.5}$$

where $\underline{\dot{u}}_\mathrm{u}$ and $\underline{\dot{u}}_\mathrm{l}$ are the surface accelerations based on a free boundary assumption, M_u is the mass associated with the *upper* node, M_l is the mass associated with the *lower* node, $\underline{F}_\mathrm{u}$ is the *upper* nodal force vector and $\underline{F}_\mathrm{l}$ is the *lower* nodal force vector. In all cases M_l, M_u, $\underline{F}_\mathrm{l}$ and $\underline{F}_\mathrm{u}$ are extracted as part of the finite element solution. The total force due to the interaction between the materials is given by

$$M\underline{\dot{u}} = \underline{F} \tag{4.6}$$

where M and $\underline{F}$ are respectively the total mass and vectorial force associated with the node. By Newton's Third Law the force can be split up into *upper* and *lower* components thus,

$$M_\mathrm{u}\underline{\dot{u}} = \underline{F}_\mathrm{u} + \underline{r}_\mathrm{u}, \quad M_\mathrm{l}\underline{\dot{u}} = \underline{F}_\mathrm{l} + \underline{r}_\mathrm{l} \tag{4.7}$$

where $\underline{F} = \underline{F}_\mathrm{u} + \underline{F}_\mathrm{l}$, $\underline{r}_\mathrm{u} + \underline{r}_\mathrm{l} = 0$ and where $\underline{r}_\mathrm{u}$ and $\underline{r}_\mathrm{l}$ are reaction forces due to the surface interaction. The sum of the reaction forces is an equivalent statement to the continuity of the normal stress components Equation (3.1) over a given area thus

$$\underline{r}_\mathrm{u} + \underline{r}_\mathrm{l} = [\sigma_n]\underline{\hat{n}}\Delta\Sigma = 0 \tag{4.8}$$

where $\Delta\Sigma$ is an elemental area at the slide interface. Since there are no frictional forces between the surfaces there is no coupling between the tangential acceleration components. The only surface interaction force is normal to it and the normal acceleration component must be modified to reflect this. Therefore, if A_{u} is the area over which the *upper* nodal force acts and A_{l} is the area over which the *lower* nodal force acts then for normal stress continuity,

$$\frac{(M_{\mathrm{u}}\dot{u} - \underline{F}_{\mathrm{u}}) \cdot \underline{\hat{n}}}{A_{\mathrm{u}}} + \frac{(M_{\mathrm{l}}\dot{u} - \underline{F}_{\mathrm{l}}) \cdot \underline{\hat{n}}}{A_{\mathrm{l}}} = 0 \tag{4.9}$$

From this equation the modified nodal normal acceleration component on the surface can be shown to be equal to

$$\underline{\dot{u}} \cdot \underline{\hat{n}} = \dot{u}_n = \frac{p_{\mathrm{u}}A_{\mathrm{u}} + p_{\mathrm{l}}A_{\mathrm{u}}}{M_{\mathrm{u}} + \overline{M}_{\mathrm{l}}A_{\mathrm{u}}} \tag{4.10}$$

where we have defined $\overline{M}_{\mathrm{l}} = M_{\mathrm{l}}/A_{\mathrm{l}}$ and the *effective* normal pressure forces to be,

$$p_{\mathrm{u}} = \frac{\underline{F}_{\mathrm{u}} \cdot \underline{\hat{n}}}{A_{\mathrm{u}}}; \quad p_{\mathrm{l}} = \frac{\underline{F}_{\mathrm{l}} \cdot \underline{\hat{n}}}{A_{\mathrm{l}}} \tag{4.11}$$

One will notice that when there is no strength (fluid dynamic problem) the *effective* pressure forces are just the hydrodynamic pressures as expected.

4.3. Asymmetric put-back-on method

With the updated accelerations the nodes are moved, but some nodes may penetrate due to numerical inaccuracies. For a node that does penetrate the distance between the slave to master surfaces is calculated and the slave node is moved onto the master surface. For velocity conservation the normal component of the slave node velocity is forced to equal the interpolated master velocity resolved in the slave normal direction.

4.4. Programming philosophy

Like PEGASUS the contact algorithm was written in FORTRAN 90 [14]. It was designed to be modular so that there was a clear interface between PEGASUS and the contact algorithm. The first stage of the slide algorithm was an initialization step. Surface data was gathered from PEGASUS and this was used to create the internal slide data structure which keeps a record of the nodes, both in terms of global and local information, on the slide surfaces and stores the slide node and element surface connectivity. With this done it was never repeated. After each Lagrangian *Predictor* step the slide algorithm gathers the required slide nodal information from PEGASUS and stores this in the slide internal data structure. The slide algorithm computes all velocities, accelerations and position vectors for the slide surface nodes. Before each Lagrangian *Corrector* step the slide surface nodal information is scattered back to the PEGASUS data structures.

5. RESULTS AND DISCUSSION

5.1. Sod shock tube problem

Sod's shock tube problem [15] was used to test the planar symmetry of the contact algorithm. The problem, illustrated in Figure 5, consists of two ideal fluid regions separated by a diaphragm. With one region at a higher pressure with respect to the other. The diaphragm

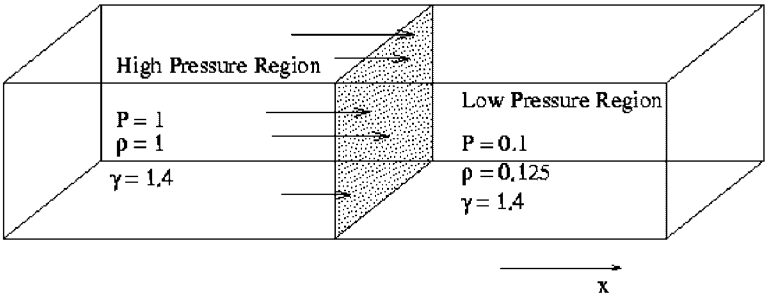

Figure 5. Sod's shock tube.

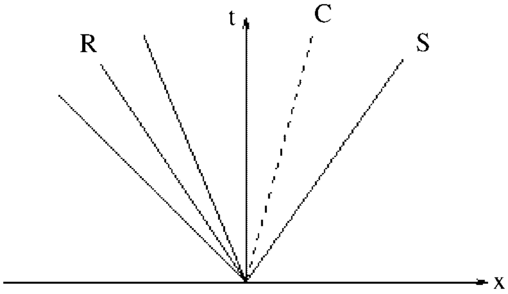

Figure 6. Wave structure. (S) shock wave, (R) rarefaction wave and (C) contact.

is broken producing a shock wave that propagates outward from left to right and a rarefaction wave from right to left. The two waves are separated by a contact discontinuity, which propagates at a constant velocity, and represents the moving interface between the two fluids. The arrangement is illustrated in Figure 6.

Numerically the problem can be setup as a single mesh, with a common set of nodes on the interface, or as two separate meshes with the inclusion of the contact algorithm. In both cases the behaviour of the interface should be the same. This was studied by placing a probe on the interface at the start of the calculation and its position and velocity measured with respect to time; the master surface was assumed to be the high pressure region. The results are illustrated in Figures 7 and 8. One can see that the agreement between the positions is excellent. However, the results for the velocity oscillate early on during the transient, or start up phase, but settle down to the same approximate value later on.

5.2. Driven spherical shell

A driven spherical shell was used to test the spherical symmetry of the contact algorithm. The driving force was radial with no shear between the materials. The configuration consisted of an outer 10 cm thick lead (Pb) shell in contact with an inner 3 mm thick copper (Cu) shell with an inner radii of 9.7 cm. The initial configuration is illustrated in Figure 9. At time

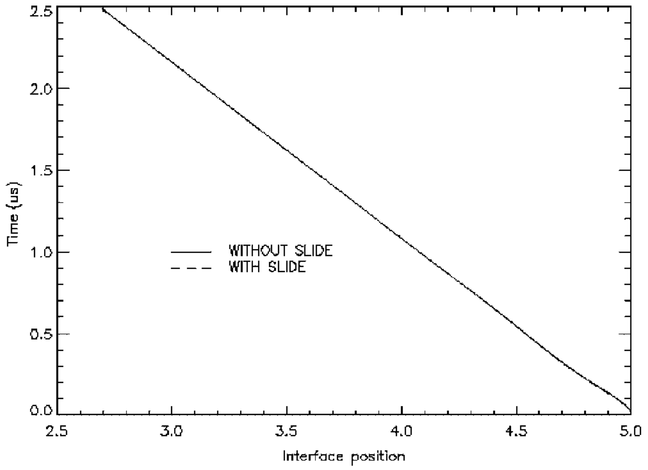

Figure 7. Spatial variation at interface.

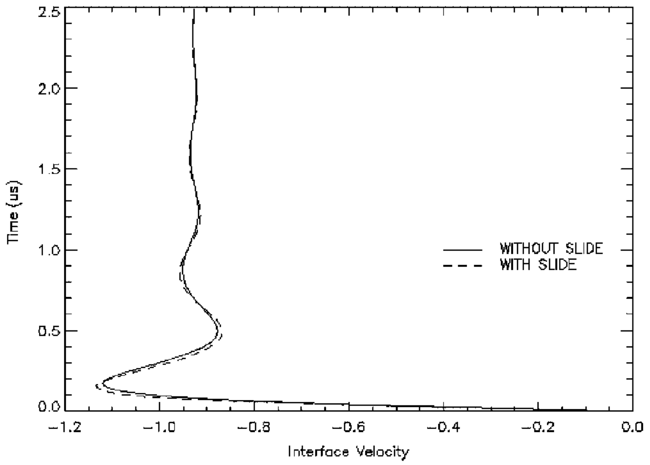

Figure 8. Velocity variation of interface.

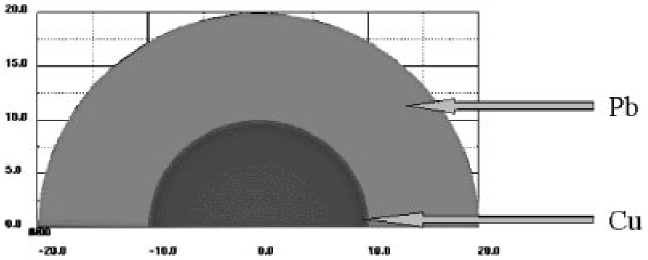

Figure 9. Driven spherical shells.

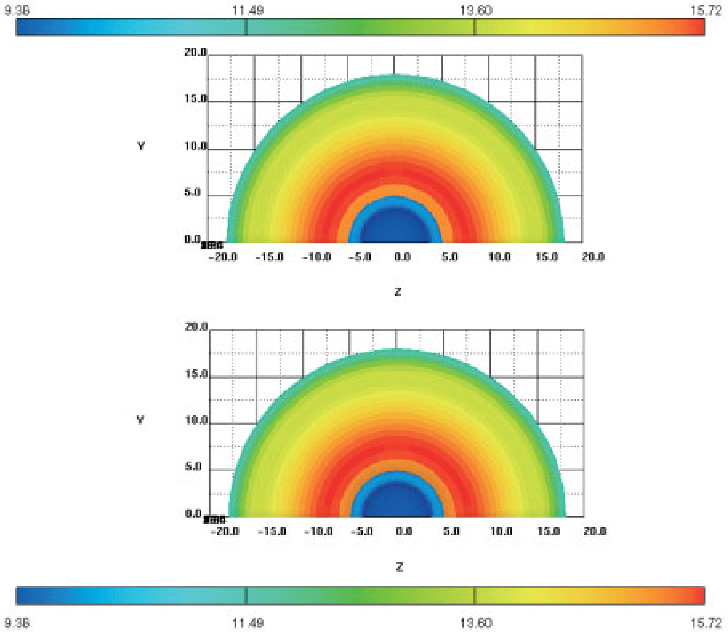

Plate 1. Density contours at $t = 30\ \mu s$ from PEGASUS. (Top) single mesh, (Bottom) split mesh.

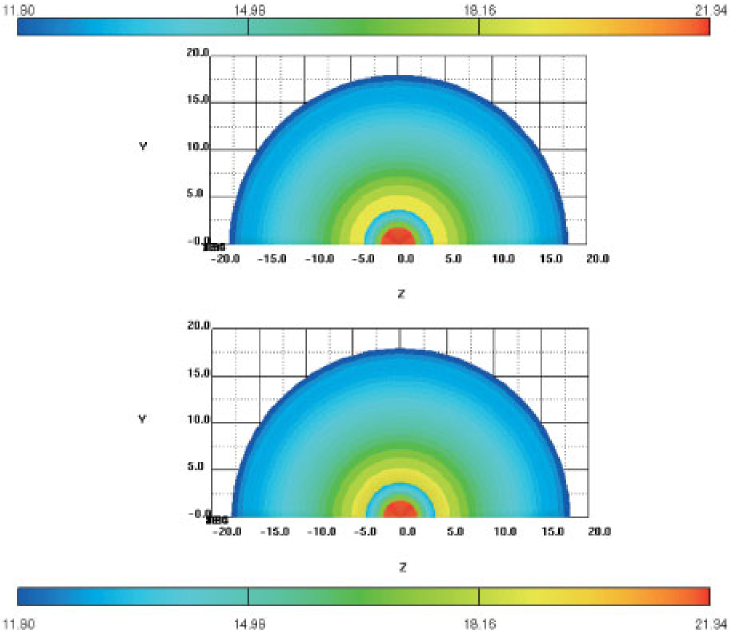

Plate 2. Density contours at $t = 35\ \mu s$ from PEGASUS. (Top) single mesh, (Bottom) split mesh.

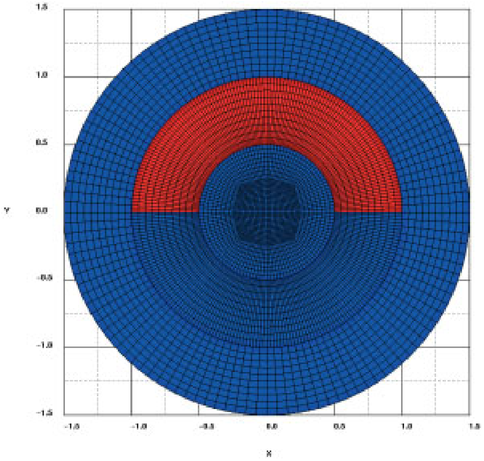

Plate 3. Arrangement of rotating discs at 0 μs. Middle disc has been colour coded for clarity. Red (Top) and blue (bottom).

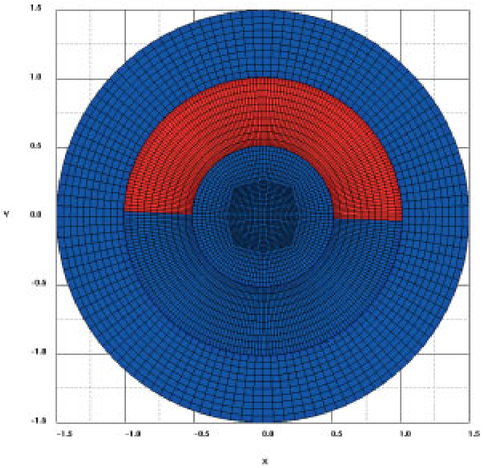

Plate 4. Low strength results after one rotation at 50 μs.

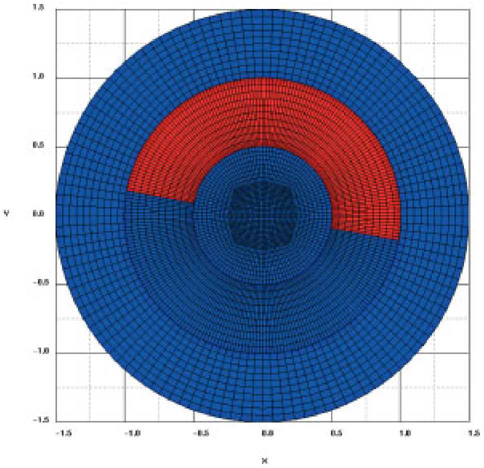

Plate 5. High strength results after one rotation at 50 μs.

zero the outer shell is given an inward radial velocity of -1.0 mm/μs. The Pb shell then compresses the Cu shell until it has collapsed.

Numerically the problem can be setup as a single mesh, with a common set of nodes on the interface, or as two separate meshes with the inclusion of the contact algorithm. In both cases the behaviour of the collapsing radial interface will be the same.

Two calculates were performed with (Pb being the master and Cu the slave) and without the contact algorithm. Equivalent axi-symmetric calculations were also performed in `CORVUS`. Density contours are illustrated for $t = 30$ μs in Plate 1 and at $t = 35$ μs in Plate 2. Visually there are appears to be no differences. To gain a greater understanding of the effect of splitting the interface, probes were placed in the calculation at radii 9, 8, 7, 6, 5, 4.5, 4, 3.5, 3, 2.5 and 2 cm. In addition, the polar angle (θ) between the probes was varied in steps of 5° from 90° at the equator to the pole at 180° and the azimuthal angle (ϕ) was varied from 0 to 90° in 5° steps. In order to compare the time the probe gets crossed by the inner Cu interface with `CORVUS` a single time was required for each radial level and polar angle. This was obtained by averaging the arrival times over the 19 azimuthal angles. Timings from the probes were compared for the two different `PEGASUS` calculations. It was found that the maximum absolute timing difference was $t = 0.0023$ μs. Equivalent `CORVUS` calculations using an axi-symmetric mesh were performed and the maximum absolute timing difference was found to be $t = 0.0008$ μs. Both timing differences are very small but the `CORVUS` calculation appears to be more accurate than the `PEGASUS` calculation because of the symmetry inherent in the axi-symmetric mesh.

5.3. Rotating disc

The rotation of a hollow disc between materials was used to test slide in the contact algorithm. The problem consisted of three concentric discs with respective outer radii 0.5, 1.0 and 1.5 cm. Both the inner disc and outer ring were fixed while the middle ring was allowed to rotate clockwise with angular velocity ω ($= 0.13$ rad/μs). The full arrangement is illustrated in Figure 10 where the labels M and S represent the position of the master and slave surfaces. The `PEGASUS` mesh was setup using 120 angular zones in the inner disc and outer ring. In the middle ring 240 angular zones where used. Due to the use of hexahedral cells non-uniform zoning had to be used around the origin within the inner disc. For illustration purposes the middle ring was coloured coded to highlight the top and bottom of the disc at the start and after one rotation, and is illustrated in Plate 3.

Unlike the other problems, which were hydrodynamic in nature, strength was used to limit the radial expansion of the middle disc; without it the disc would fly apart. The effect of this is demonstrated in Figures 11 and 12 where the *magnitude* of material strength directly effects the increase in the volume of the inner and middle discs. The change in volume of the inner disc is a direct consequence of the put-back-on step because the outer surface of the inner disc must conform to the inner surface of the middle disc; which has a tendency to fly outward. Similarly for the outer surface of the middle disc which tends to fly outward. Therefore, the degree of expansion of the inner and middle discs is directly dependent on the magnitude of material strength.

The effect of the different strength levels are illustrated in Plates 4 and 5. After one revolution (at 50 μs), and without friction, the middle disc would have moved by 6.5 radians. However, for low strength the discs have a tendency to expand causing them to slow down,

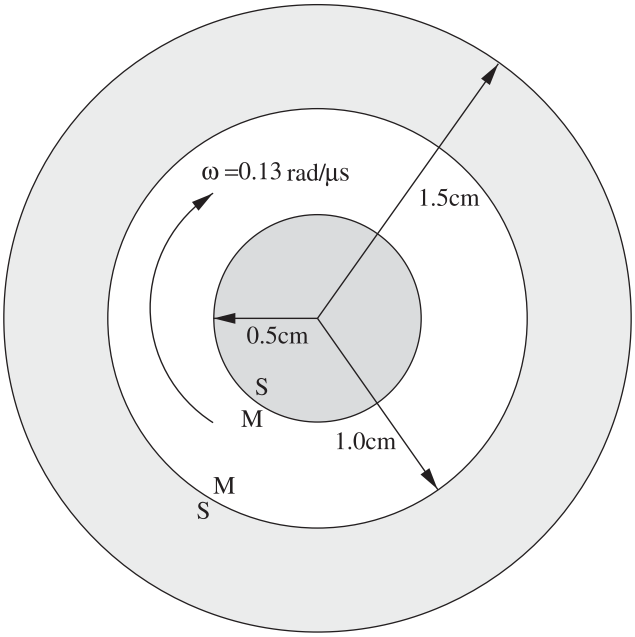

Figure 10. Rotating disc. Shaded regions are stationary.

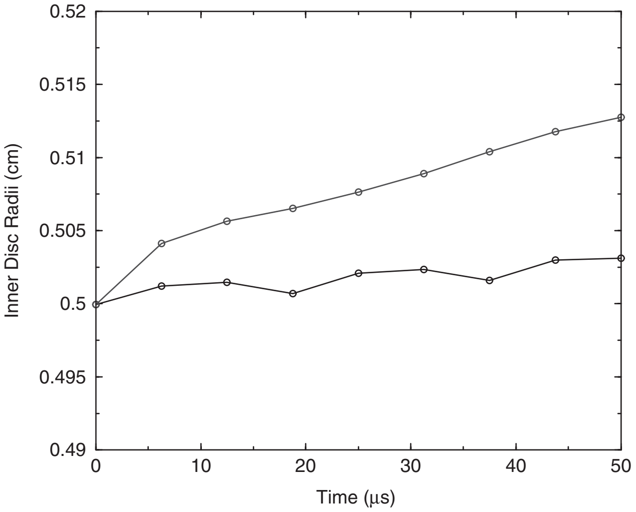

Figure 11. Effect of strength on the rigidity of inner disc radii. Lower curve, high strength; upper curve, low strength.

which is illustrated in Plate 4. However, by significantly increasing the strength contribution the discs become less able to expand and less likely to slow down, which is illustrated in Plate 5. This latter result being very close to the analytic value.

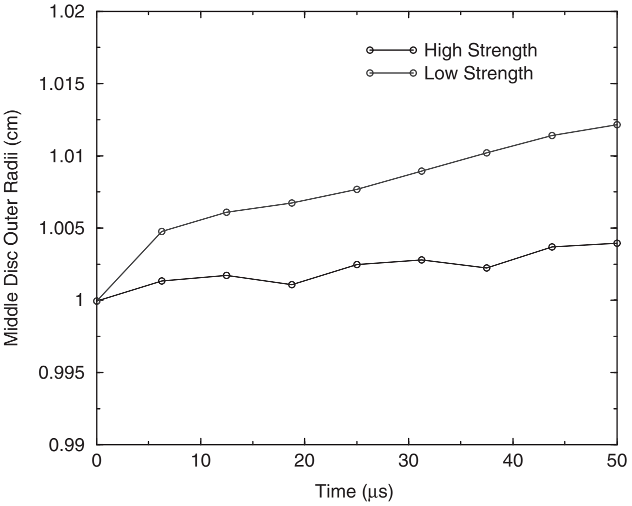

Figure 12. Effect of strength on the rigidity of middle disc outer radii. Lower curve, high strength; upper curve, low strength.

6. CONCLUSIONS

In this paper the development of an algorithm for sliding contact between materials was presented. The algorithm was described in the context of the Lagrangian, explicit, staggered mesh, finite-element, predictor–corrector code `PEGASUS` and results for a number of different problems presented. From them it is observed that the algorithm is accurate and robust. The modularity of the algorithm makes it suitable for it to be incorporated into any explicit Lagrangian code based on the predictor–corrector philosophy independent of the choice of numerical method. The only requirement being that hydrodynamic values can be defined (such as an *effective* pressure force and velocities) on the surfaces in contact with each other.

APPENDIX A: PROPERTIES OF ISO-PARAMETRIC SURFACE

For convenience we consider the element face defined by $\zeta_k = \frac{1}{2}$ and $\zeta = \frac{1}{2}$ illustrated in Figure A1.

Whence,

$$N_k(\xi,\eta) = \tfrac{1}{4}(1 + 4\xi_k\xi)(1 + 4\eta_k\eta) \tag{A1}$$

The position vector in terms of the iso-parametric co-ordinate frame is given by

$$\underline{x} = \underline{a}_0 + \underline{a}_1\xi + \underline{a}_2\eta + \underline{a}_3\xi\eta \tag{A2a}$$

where

$$\begin{aligned} \underline{a}_0 &= \tfrac{1}{4}(\underline{x}_1 + \underline{x}_2 + \underline{x}_3 + \underline{x}_4), & \underline{a}_1 &= \tfrac{1}{2}(-\underline{x}_1 + \underline{x}_2 + \underline{x}_3 - \underline{x}_4) \\ \underline{a}_2 &= \tfrac{1}{2}(-\underline{x}_1 - \underline{x}_2 + \underline{x}_3 + \underline{x}_4), & \underline{a}_3 &= \underline{x}_1 - \underline{x}_2 + \underline{x}_3 - \underline{x}_4 \end{aligned} \tag{A2b}$$

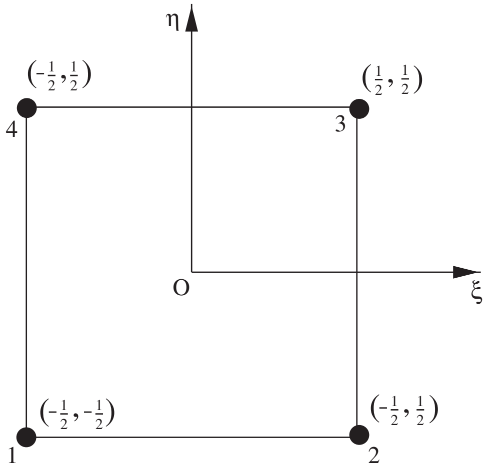

Figure A1. Element face in iso-parametric space.

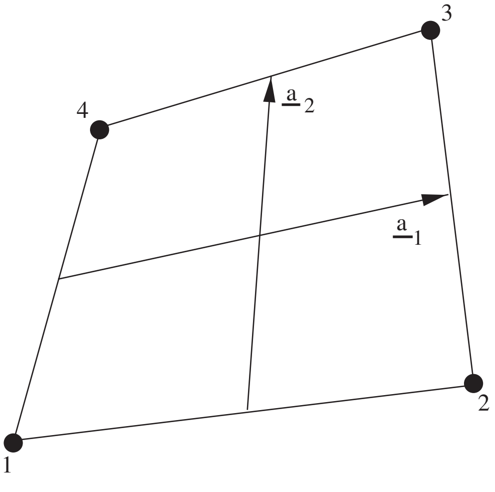

Figure A2. Characteristic vectors.

Geometrically $\underline{a}_0$ represents an average position vector, $\underline{a}_3$ is a measure of the face skewness and the vectors $\underline{a}_1$ and $\underline{a}_2$ are represented schematically in Figure A2. The normal vector at any point on the surface is given by

$$\underline{n}(\xi,\eta)=\frac{\partial \underline{x}}{\partial \xi}\wedge\frac{\partial \underline{x}}{\partial \eta}=(\underline{a}_1\wedge\underline{a}_2)+(\underline{a}_3\wedge\underline{a}_2)\eta+(\underline{a}_1\wedge\underline{a}_3)\xi \qquad \text{(A3)}$$

APPENDIX B: DETERMINATION OF ISO-PARAMETRIC CO-ORDINATES

It is trivial to calculate the position vector $\underline{x}$ from Equation (A2a) when the iso-parametric co-ordinates (ξ, η) are known. However, to determine the surface intersection point we must be able to solve the inverse problem i.e. given the position vector what are the iso-parametric co-ordinates and this is far from trivial. As discussed in the main text the position vector $\underline{x}$ is represented by,

$$\underline{x} = \underline{x}_o + \lambda \underline{n} \tag{B1}$$

where $\underline{x}_o$ is the *upper* position vector and λ is the distance to the surface.

To determine the co-ordinates there are different cases to consider. We first define the normal vector $\underline{m}$ to be,

$$\underline{m} = \underline{a}_1 \wedge \underline{a}_2 \tag{B2}$$

and any point on the plane will satisfy the equation $\underline{m} \cdot \underline{x} = h$ where h is a constant for the plane. By defining the parameter Δ by

$$\Delta = \underline{a}_3 \cdot \underline{m} \tag{B3}$$

it is trivial to prove that all four points $(\underline{x}_k)$ lie on the plane when $\Delta = 0$. We now go on to investigate solutions to Equation (A2a) for non-planar and planar surfaces.

B.1. Planar case

For $\Delta = 0$ all four points lie on a plane. Therefore,

$$(\underline{x}_o - \underline{a}_0 + \lambda \underline{n}) \cdot (\underline{a}_1 \wedge \underline{a}_2) = 0 \tag{B4}$$

which yields a single value for λ thus,

$$\lambda = \frac{(\underline{a}_0 - \underline{x}_o) \cdot (\underline{a}_1 \wedge \underline{a}_2)}{\underline{n} \cdot (\underline{a}_1 \wedge \underline{a}_2)} \tag{B5}$$

This equation becomes invalid when $\underline{n}$ is perpendicular to the vector $\underline{a}_1 \wedge \underline{a}_2$ i.e. when $\underline{n}$ is parallel to the surface and does not intersect it.

B.1.1. Linear situation. When $|\underline{a}_3| = 0$, Equation (A2a) reduces to the simpler form

$$\underline{x}_o - \underline{a}_0 + \lambda \underline{n} = \underline{a}_1 \xi + \underline{a}_2 \eta \tag{B6}$$

By taking the dot product with respect to either $\underline{a}_1$ or $\underline{a}_2$ yields two simultaneous equations which can be solved simultaneously for the (ξ, η).

B.1.2. Non-linear situation. When $|\underline{a}_3| \neq 0$ all vectors are co-planar and any vector in the plane can be written as a linear combination of $\underline{a}_1$ and $\underline{a}_2$ thus,

$$\underline{a}_3 = A\underline{a}_1 + B\underline{a}_2 \tag{B7}$$

and because $\underline{x}$ and $\underline{a}_0$ are in the plane,

$$\underline{x}_o - \underline{a}_0 + \lambda \underline{n} = C\underline{a}_1 + D\underline{a}_2 \tag{B8}$$

The coefficients are derived in the same way as in Section B.1.1. Equation (A2a) reduces to

$$C\underline{a}_1 + D\underline{a}_2 = \underline{a}_1\xi(1+A\eta) + \underline{a}_2\eta(1+B\xi) \tag{B9}$$

Equating coefficients yields,

$$\begin{aligned} C &= \xi(1+A\eta) \\ D &= \eta(1+B\xi) \end{aligned} \tag{B10}$$

Upon elimination of η yields the quadratic equation,

$$B\xi^2 + (1+\delta)\xi - C = 0 \tag{B11}$$

where $\delta = AD - BC$. The roots of the equation are given by

$$\xi_{\pm} = -\frac{1+\delta}{2B} \pm \frac{1}{2B}\sqrt{(1+\delta)^2 + 4BC} \tag{B12}$$

For $(1+\delta)^2 + 4BC > 0$ there are two roots (i.e. two sets of iso-parametric co-ordinates) for one value of λ. The surface in (ξ,η) space must fold back onto itself at the cusp. The choice of root must be selected carefully because both give the same positional vector! However, a planar surface based on positional vectors will not in general produce an equivalent planar surface based on the velocity vectors and because of this the different roots will give different velocities! Provided the cusp does not pass through the cell face then the root is chosen in the following way. For a point $\underline{x}_o$ lying on the surface at $\underline{a}_o$ then (ξ,η) will be $(0,0)$ and $\lambda = 0$. To guarantee this the sign of the root in Equation (B12) must be selected such that,

$$\begin{aligned} \tfrac{1+\delta}{2B} > 0, &\quad \xi = \xi_+ \\ \tfrac{1+\delta}{2B} < 0, &\quad \xi = \xi_- \end{aligned} \tag{B13}$$

The cusp locus can be found by considering the curve along which $(1+\delta)^2 + 4BC = 0$ is true. Thus,

$$\xi = -\frac{1+\delta}{2B}, \quad \eta = \frac{\delta - 1}{2A} \tag{B14}$$

Therefore, the locus of points where (ξ,η) are single valued is given by the linear equation,

$$B\xi + A\eta + 1 = 0 \tag{B15}$$

The boundaries where the cusp just skims the edges of the iso-parametric face are illustrated in Figure B1. From this the cusp can be shown to pass through the face region if it lies in the shaded region depicted in Figure B2. When the cusp does pass through the square the element face can become non-convex, i.e. boomerang, and is illustrated in Figure B3.

B.2. Non-planar case

For $\Delta \neq 0$ the vectors are non-co-planar. Let,

$$\underline{x}_o - \underline{a}_0 = \alpha\underline{a}_1 + \beta\underline{a}_2 + \gamma\underline{a}_3 \tag{B16}$$

$$\underline{n} = p\underline{a}_1 + q\underline{a}_2 + r\underline{a}_3 \tag{B17}$$

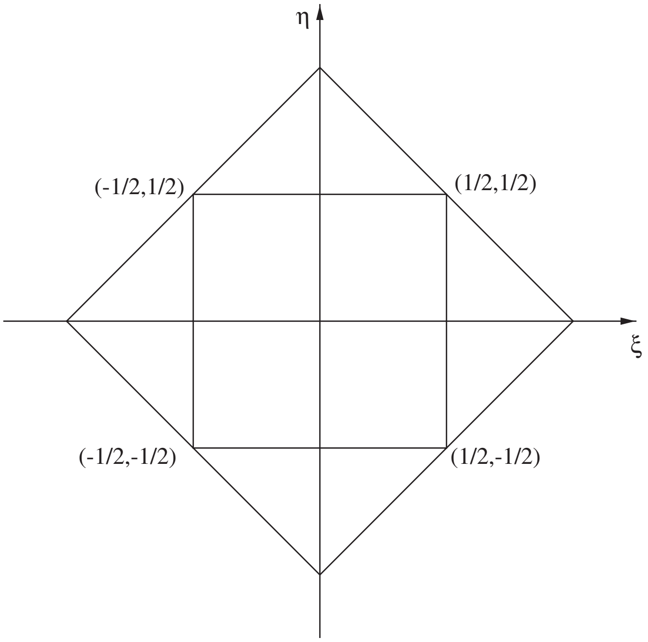

Figure B1. Diamond represents the boundaries where cusp just skims edges.

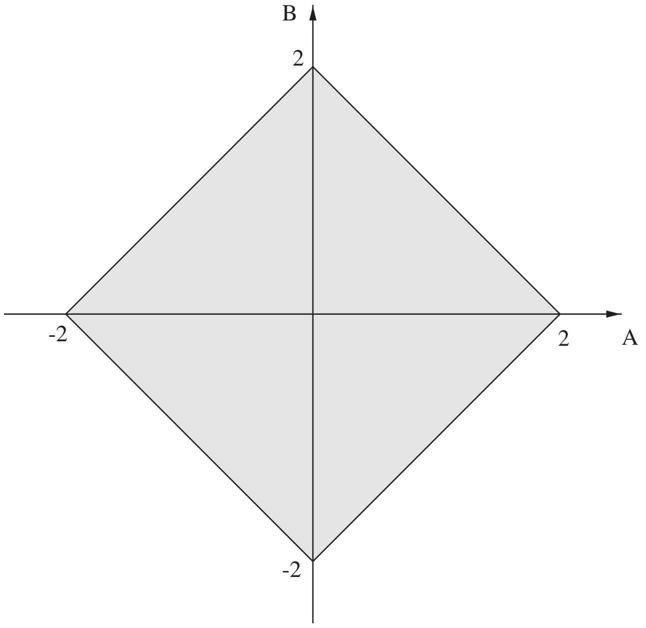

Figure B2. Cusp does not pass through iso-parametric surface when (A,B) lie in the shaded region.

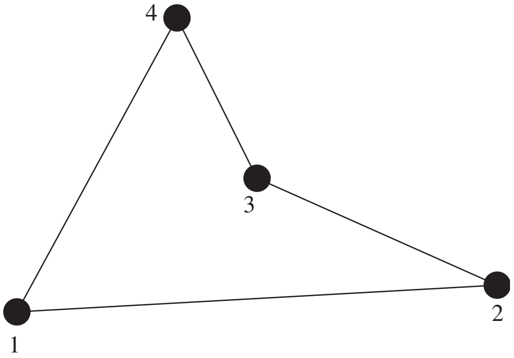

Figure B3. Element face that has boomeranged.

Hence

$$\alpha = \frac{\delta\underline{x}_o \cdot (\underline{a}_2 \wedge \underline{a}_3)}{\Delta}, \quad \beta = \frac{\delta\underline{x}_o \cdot (\underline{a}_3 \wedge \underline{a}_1)}{\Delta}, \quad \gamma = \frac{\delta\underline{x}_o \cdot (\underline{a}_1 \wedge \underline{a}_2)}{\Delta} \tag{B18}$$

where $\delta\underline{x}_o = \underline{x}_o - \underline{a}_o$ and

$$p = \frac{\underline{n} \cdot (\underline{a}_2 \wedge \underline{a}_3)}{\Delta}, \quad q = \frac{\underline{n} \cdot (\underline{a}_3 \wedge \underline{a}_1)}{\Delta}, \quad r = \frac{\underline{n} \cdot (\underline{a}_1 \wedge \underline{a}_2)}{\Delta} \tag{B19}$$

Equating terms yields,

$$A(\xi,\lambda)\underline{a}_1 + B(\eta,\lambda)\underline{a}_2 + C(\xi,\eta,\lambda)\underline{a}_3 = 0 \tag{B20}$$

where

$$\begin{aligned} A(\xi,\lambda) &= \alpha + \lambda p - \xi \\ B(\eta,\lambda) &= \beta + \lambda q - \eta \\ C(\xi,\eta,\lambda) &= \gamma + \lambda r - \xi\eta \end{aligned} \tag{B21}$$

Now because the vectors in Equation (B20) are not linearly dependent it can only be satisfied when $A = 0$, $B = 0$ and $C = 0$. Whence

$$\gamma + \lambda r = (\alpha + \lambda p)(\beta + \lambda q) \tag{B22}$$

Which upon expansion yields a quadratic equation for the distance to the surface λ thus,

$$pq\lambda^2 + (p\beta + \alpha q - r)\lambda + \alpha\beta - \gamma = 0 \tag{B23}$$

Hence for $pq \neq 0$ it has the roots given by,

$$\lambda_{\pm} = \frac{(r - p\beta - \alpha q) \pm \sqrt{(p\beta + \alpha q - r)^2 + 4pq(\gamma - \alpha\beta)}}{2pq} \tag{B24}$$

Provided $(p\beta + \alpha q - r)^2 + 4pq(\gamma - \alpha\beta) \geqslant 0$ the vector $\hat{\underline{n}}$ will intersect the surface either once or twice. When it intersects twice the surface must curve back on itself with co-ordinates

$(\xi_\pm, \eta_\pm)$ given by,

$$\begin{aligned} \xi_\pm &= \alpha + p\lambda_\pm \\ \eta_\pm &= \beta + q\lambda_\pm \end{aligned} \tag{B25}$$

The sign of the root must be chosen to satisfy the condition when the point $\underline{x}_o$ lies on the surface at $\underline{a}_o$ then $(\xi = 0, \eta = 0)$ and $\lambda = 0$; this latter result is equivalent to setting $\gamma = \alpha\beta$. To guarantee this the sign of the root in Equation (B24) must be selected such that

$$\begin{aligned} r - p\beta - \alpha q &\geqslant 0, \quad \lambda = \lambda_- \\ r - p\beta - \alpha q &< 0, \quad \lambda = \lambda_+ \end{aligned} \tag{B26}$$

The locus of points where (ξ, η, λ) are single valued and along which the surface curves back, in physical space, can be found by considering the curve along which $(p\beta + \alpha q - r)^2 + 4pq(\gamma - \alpha\beta) = 0$ is true. Hence,

$$\begin{aligned} \lambda_0 &= (r - p\beta - \alpha q)/(2pq) \\ \xi_0 &= \alpha + p\lambda_0 \\ \eta_0 &= \beta + q\lambda_0 \end{aligned} \tag{B27}$$

Elimination of λ_0 yields a linear equation (i.e. straight line) for the cusp thus,

$$q\xi_0 - p\eta_0 + (p\beta - q\alpha) = 0 \tag{B28}$$

The analysis of this equation follows the same procedure as the planar situation discussed earlier.

ACKNOWLEDGEMENTS

I would like to thank Ian Macdonald and Andy Barlow for their help over the course of this work.

REFERENCES

1. Hughes WF, Brighton JA. *Schaum's Outline of Theory and Problems of Fluid Dynamics*. McGraw-Hill Book Company: New York, 1967.
2. Wilkins ML. *Calculation of Elastic-Plastic Flow*, vol. 3. *Methods in Computational Physics*. Academic Press: New York, 1964.
3. Youngs DL. Time-dependent multi-material flow with large fluid distortions. In *Numerical Methods for Fluid*, Morton KW, Baines MJ (eds). Academic Press: New York, 1982.
4. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
5. Barlow A. ALE in `CORVUS`. *New Models and Numerical Codes for Shockwave Processes in Condensed Media*. St. Catherines College: Oxford, 1997.
6. Barlow AJ, Whittle J. Mesh adaptivity and material interface algorithms in a two dimensional Lagrangian hydrocode. *Chemical and Physical Reports* 2000; **19**(2):233–258.
7. MacDonald I. Development of a 3D Lagrangian hydrocode. *New Models and Numerical codes for Shockwave Processes in Condensed Media*. St. Catherines College: Oxford, 1997.
8. MacDonald I. 3D Simple ALE in `PEGASUS`. *New Models and Predictive Methods for Shock Waves/Dynamic Processes in Energetic Materials and Related Solids*, Maryland, USA, 1999.
9. Benson DJ. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:235–394.
10. Dawes AS. 3D Slide in `PEGASUS`. *New Models and Predictive Methods for Shock Waves/Dynamic Processes in Energetic Materials and Related Solids*, Maryland, USA, 1999.

11. Hallquist JO, Goudreau GL, Benson DJ. Sliding interfaces with contact-impact in large-scale Lagrangian computations. *Computer Methods in Applied Mechanics and Engineering* 1985; **51**:107–137.
12. Wilkins ML. *Computer Simulation of Dynamic Phenomena*. Springer: Berlin, 1999.
13. Hirsch C. *Numerical Computation of Internal and External Flows*, vol. 1. Wiley: New York, 1988.
14. Brained WS, Goldberg CH, Adams JC. *Programmers Guide to Fortran 90*. McGraw-Hill Book Company: New York, 1990.
15. Sod GA. A numerical study of a converging cylindrical shock. *Journal of Fluid Mechanics* 1977; **83**:785–794.