



An Eulerian shell formulation for fluid-structure interaction

David J Benson, Laurent Stainier

► To cite this version:

David J Benson, Laurent Stainier. An Eulerian shell formulation for fluid-structure interaction. Computer Methods in Applied Mechanics and Engineering, 2000, 187 (3-4), pp.571-590. 10.1016/S0045-7825(99)00340-0 . hal-01005231

HAL Id: hal-01005231

<https://hal.science/hal-01005231>

Submitted on 5 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

An Eulerian shell formulation for fluid–structure interaction

David J. Benson *, Laurent Stainier

*Division of Mechanical Engineering, Department of Applied Mechanics and Engineering Sciences, University of California, San Diego,
9500 Gilman Drive, Mail Code 0411, La Jolla, CA 92093-0411, USA*

Coupled Eulerian–Lagrangian formulations are currently used to analyze fluid–structure interaction. Extending these formulations to handle large structural deformations, structural failure, and perforation is challenging. An Eulerian structural formulation is attractive because it permits arbitrarily large deformations and permits the new surfaces associated with perforation to evolve without specialized algorithms. The theoretical basis for an Eulerian formulation for shells is developed and, in particular, the choice of a formulation for the rotations is discussed. Example calculations with large deformations, shell thinning, and weld failure are presented.

1. Introduction

Fluid–structure interaction problems are common in a wide variety of engineering disciplines. Most of these problems require no more than a linear model of the structure, perhaps coupled with large rotations. The traditional computational approach couples a Lagrangian structural model to an Eulerian fluid model through a specialized contact algorithm. This approach works well for many problems, but encounters difficulties when applied to problems where the structure is perforated or fails in a manner which permits the fluid to flow through it. A typical problem of this type is the detonation of an explosive near an underwater structure.

The strategies for coupling of Lagrangian structural models with Eulerian fluids either use a spatially fixed Eulerian mesh and calculate its intersection with the Lagrangian structure [1], or use an arbitrary Lagrangian Eulerian (ALE) fluid with a moving boundary that follows the motion of the structure. The first strategy permits the perforation of the structure by eroding or deleting selected elements from the structure, but it is computationally expensive due to the cost of the shell-fluid intersection calculation: a structural element may span one or more fluid elements, and a fluid element may contain an arbitrary number of structural elements. Mapping the fluid pressure onto the structural elements from the fluid, and calculating the strain rate in the fluid in elements that intersect the structure, in an accurate, consistent, energy and momentum conserving manner is very difficult. The primary advantage of the second strategy is its speed. Since it eliminates the overlap of Lagrangian elements with Eulerian elements, the complexity of the calculation is reduced, and by constraining the Lagrangian structure to the boundary of the fluid, the issues associated with load balancing on parallel computers are also simplified. Although the second strategy theoretically permits perforation via an element erosion algorithm, the distortion of the (nearly Lagrangian) fluid boundary as the fluid flows through the perforation will quickly stop the calculation.

Many issues similar to those in fluid–structure interaction arise when surface tension is an important effect, such as sloshing in fuel tanks. The surface tension is derived from a surface energy potential that is a

* Corresponding author.

function of the curvature, and behaves, in structural terms, like a membrane. Since the surface connectivity may change during the sloshing, an Eulerian formulation for the surface tension is attractive. In response to this class of problems, finite difference researchers in computational fluid dynamics have developed methods that do not require a Lagrangian model for the membrane forces [2]. These efforts have met with varying degrees of success with the cited finite volume approach appearing to be among the most successful. Many of the difficulties seem to be associated with the difficulty of obtaining an accurate approximation of the surface curvature on a spatially fixed mesh.

In this paper, a formulation for an Eulerian shell in two-dimensions is presented with example calculations. An extension of the formulation to three-dimensions is intended, and its development will follow the same methodology. Section 2 summarizes the computational methods that are used in multi-material Eulerian hydrodynamics [3,4]. In Section 3, the Eulerian shell element geometry and the kinematics are developed based on the shear deformable theory developed by Reissner [5] and Mindlin [6]. A detailed examination of two formulations for the rotational degrees of freedom results in a departure from the Eulerian surface tension formulations, which are formulated entirely in terms of the translational degrees of freedom. Constitutive equations and their numerical integration are also discussed in Section 3. The discrete finite element expressions are derived and discussed in Section 4. A new approach for estimating the stable time step [7] is applied to the shell formulation in Section 5. Preliminary linear and nonlinear example calculations demonstrate the potential of an Eulerian shell formulation.

2. The Eulerian formulation

The following brief discussion of the Eulerian formulation used in [3] provides an overview of the methods commonly used in multi-material Eulerian hydrocodes; a detailed review of explicit Lagrangian and Eulerian methods appears elsewhere [4]. The governing continuum equations in conservation form are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b}, \quad (2)$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} + \rho \mathbf{b} \cdot \mathbf{u}, \quad (3)$$

where ρ is the density, \mathbf{u} the velocity, \mathbf{x} the spatial coordinate, $\boldsymbol{\sigma}$ the Cauchy stress tensor, $\dot{\boldsymbol{\epsilon}}$ the strain rate tensor, \mathbf{b} the body force, and e is the internal energy.

The solution is advanced in time by splitting the governing equations [8] into two sets, Lagrangian and Eulerian, and solving them sequentially. Two meshes, one Lagrangian and the other, a spatially fixed Eulerian mesh, are initially superposed at the beginning of the time step. During the Lagrangian step, time is advanced and the Lagrangian mesh deforms to follow the material motion. Time is held constant in the Eulerian step, which projects the Lagrangian solution from the deformed mesh onto the spatially fixed mesh.

2.1. The Lagrangian step

The Lagrangian solution is advanced in time with an explicit finite element formulation similar to the ones found in DYNA3D [9], EPIC [10], HEMP [11], and PRONTO [12]. HEMP is a finite difference code developed prior to the finite element codes, but it is identical to a finite element formulation with one point integration [13].

Quadrilateral elements with one point integration are used for the spatial discretization. The Lagrangian step is summarized by Eq. (4), where \mathbf{F} is the force vector associated with the boundary conditions and the hourglass viscosity, \mathbf{M} the diagonal mass matrix, \mathbf{B} the discrete gradient operator, $\boldsymbol{\sigma}$ the Cauchy stress, \mathbf{x}

the vector of current coordinates, and \mathbf{u} is the velocity. The augmented stress, $\tilde{\sigma}$, is the sum of the Cauchy stress and the shock viscosity, q [14].

$$\begin{aligned}\mathbf{u}^{n+1/2} &= \mathbf{u}^{n-1/2} + \Delta t \mathbf{M}^{-1} \{ \mathbf{F} - \int \mathbf{B}^T \tilde{\sigma} \, d\Omega \}, \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \Delta t \mathbf{u}^{n+1/2}, \\ \tilde{\sigma} &= \sigma - q \mathbf{I}.\end{aligned}\tag{4}$$

Elements containing more than one material are commonly referred to as “mixed” elements. The only difference in the formulation of an element containing a single material and one containing several materials occurs in the mixture theory [15], a detail discussed in a later section.

2.2. The Eulerian Transport Step

The formulation of the Eulerian shell is influenced by two aspects of the Eulerian step, namely the material interface reconstruction algorithm, which defines the geometry of the shell within the Eulerian finite element, and the momentum transport, which constrains the form of mass matrix.

The second order accurate, monotonic MUSCL algorithm [16] projects the solution from the deformed Lagrangian mesh onto the Eulerian mesh. MUSCL was developed as a higher-order accurate Godunov method, which assumes all the state data (e.g., stress) is located at the element centroid. The MUSCL algorithm is one-dimensional, and it is extended to multi-dimensions by using sweeps in alternating directions [17].

Since the momentum is a function of the velocities, located at the nodes, special care must be taken to transport the momentum so that it is conserved and oscillations are not introduced into the solution. The simplest of several momentum transport methods [18], the staggered mesh method, is readily implemented on a logically regular mesh. For node c in one-dimension, with nodes l and r to the left and right of it respectively, the velocity is updated according to

$$\mathbf{u}_c^+ = \frac{1}{M_c^+} \{ M_c^- \mathbf{u}_c^- + \delta m_{lc} \mathbf{u}_l^- - \delta m_{cr} \mathbf{u}_r^- \},\tag{5}$$

$$\delta m_{lc} = \frac{1}{2} (\delta v_l \rho_l^- + \delta v_c \rho_c^-),\tag{6}$$

$$\delta m_{cr} = \frac{1}{2} (\delta v_r \rho_r^- + \delta v_c \rho_c^-),\tag{7}$$

where M is the nodal mass, δm_{ab} the mass transported from node a to node b , δv_a the volume transported between adjacent elements through node a , and the superscripts $-$ and $+$ indicate values before and after the transport calculation. The expressions for the δm were derived so that a uniform velocity distribution, with a variable density distribution, is uniform after transport.

When an element contains several materials, their individual contributions to the transport volumes must be calculated and their accurate calculation requires the geometry of the material interfaces within the element. Many different methods are available for tracking/reconstructing the material interfaces, but the most popular are the volume of fluid (VOF) methods [19], which calculate the material interfaces based on the volume of each material within an element and its neighbors. Instead of working directly with the material volume, VOF methods often use the volume fraction, V_m^f , which is the ratio of the volume of material m , V_m , to the total volume of the element, V^e . The interface reconstruction method used in many modern hydrocodes was developed by Youngs [20]. The version used here incorporates the extensions developed for CAVEAT [21] by researchers at Los Alamos National Laboratories [22]. Within each element, the interface between two materials is approximated as a straight line,

$$\mathbf{n} \cdot \mathbf{x} - d = 0,\tag{8}$$

where \mathbf{n} is the interface normal, \mathbf{x} the vector of the current coordinates, and d is a constant. In two-dimensions, the normal direction is calculated from the volume fractions of the material in the element and its surrounding eight neighbors using a finite difference stencil for the gradient,

7	8	9
4	5	6
1	2	3

Fig. 1. The numbering of the elements for evaluating the finite difference stencil for ∇V^f .

$$\mathbf{n} = \nabla V^f / |\nabla V^f|, \quad \nabla V^f = \tilde{\mathbf{B}}\{V^f\}, \quad (9)$$

and the value of d is determined from the required volume fraction. For a problem containing only two materials, the location and orientation of the interface will be independent of which material is chosen for Eq. (8)

The finite difference stencil for ∇V^f is:

$$\tilde{\mathbf{B}} = \mathbf{J}^{-1} \mathbf{B}_V, \quad (10)$$

$$\mathbf{J} = \frac{1}{2} \begin{bmatrix} -x_1 - x_2 + x_3 + x_4 & -y_1 - y_2 + y_3 + y_4 \\ -x_1 + x_2 + x_3 - x_4 & -y_1 + y_2 + y_3 - y_4 \end{bmatrix}, \quad (11)$$

$$\mathbf{B}_V = \frac{1}{4} \begin{bmatrix} -1 & 0 & +1 & -2 & 0 & +2 & -1 & 0 & 1 \\ -1 & -2 & -1 & 0 & 0 & 0 & +1 & +2 & +1 \end{bmatrix}, \quad (12)$$

where the elements are numbered as in Fig. 1. The interface reconstruction is for E_s , the center element.

When an element contains several materials, the interfaces are reconstructed sequentially using the “onion skin” model. Assume, for convenience, that material 1 is adjacent to material 2, and material 2 lies between materials 1 and 3, and so on, with material k located between materials $k - 1$ and $k + 1$. The interface between materials 1 and 2 is calculated by evaluating Eqs. (8) and (9) with the volume fractions of material 1. The second interface, between materials 2 and 3, is calculated using the sum of the volume fractions for materials 1 and 2. Continuing on through the materials, the interface between materials k and $k + 1$ is calculated using the sum of the volume fractions of materials 1 through k . For a specified ordering of materials, the volume fractions used in Eq. (9) for interface k is given by

$$V^f(k) = \sum_{j=1}^k V_{m(j)}^f, \quad (13)$$

where $m(j)$ is the j th material in the list of adjacent materials. Note that $m(j)$, in practice, may vary as a function of both space and time [23].

3. Shell geometry and kinematics

The geometry of the Eulerian shell element is defined by the upper and lower shell material interfaces calculated by the interface reconstruction algorithm [20,22]. In two-dimensions, the reference lamina is a line segment intersecting the boundaries of the element, and in three-dimensions, the lamina will be a plane (Fig. 2). Due to the assumptions used to construct the interfaces, the theoretical issues (and computational costs) associated with a nonplanar shell geometry are eliminated.

The Eulerian shell formulation is based on the shear deformable theory developed by Reissner [5] and Mindlin [6]. Its discretization follows Hughes and Liu [24,25], and Belytschko and Tsay [26]. A generic element in its local coordinate system, illustrated in Fig. 3, consists of n layers of independent materials. The shell material interfaces are forced to be parallel for simplicity, and future investigations may remove this restriction. The perpendicular to the material interfaces defines, to within a sign, the normal direction through the shell, and the exterior direction is determined from the ordering of the materials.

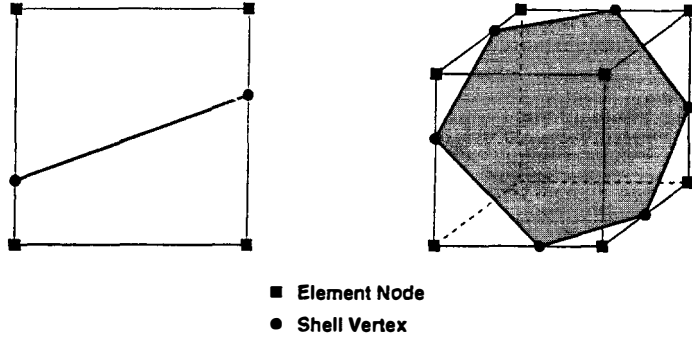


Fig. 2. Geometry of the Eulerian shell element in two- and three-dimensions.

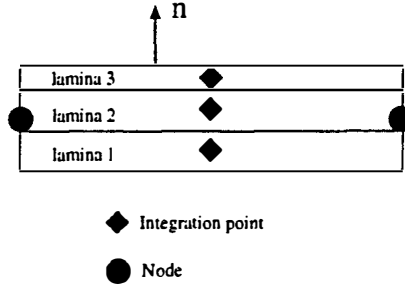


Fig. 3. Lamina structure of the constant thickness Eulerian shell element in two-dimensions.

The layered structure of independent materials was adopted to simplify the transport calculations. Each integration point in the shell requires the same amount of storage as an integration point for any other material. The introduction of the n shell integration points as independent materials is simpler than modifying the transport logic to handle special data blocks that are n times the standard length for the shell elements. Although the shells in the example calculations are homogeneous through the thickness, the generality of the formulation, and its implementation, permit composite shells of different materials.

3.1. Interpolation of the shell variables

As illustrated in Fig. 4, the η -coordinate lines define the shell lamina, while the ζ -coordinate defines the fiber direction. The position vector is defined in terms of a point on the reference laminae (e.g., at mid-thickness) and a fiber vector starting from this point (Fig. 5),

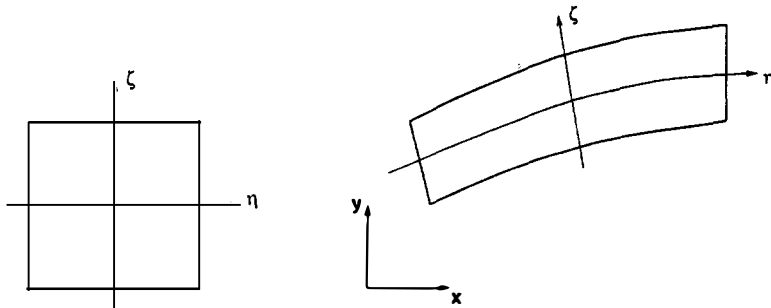


Fig. 4. Mapping from isoparametric coordinates to the physical shell domain.

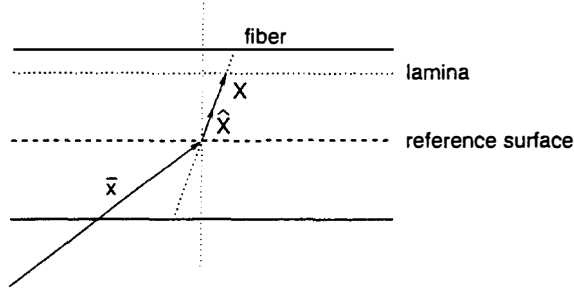


Fig. 5. Shell geometric description.

$$\mathbf{x}(\eta, \zeta) = \bar{\mathbf{x}}(\eta) + \mathbf{X}(\eta, \zeta) \quad (14)$$

$$= \bar{\mathbf{x}}(\eta) + z(\zeta) \hat{\mathbf{X}}(\eta) \quad (15)$$

$$= N_a(\eta) \bar{\mathbf{x}}_a + z(\zeta) N_a(\eta) \hat{\mathbf{X}}_a, \quad (16)$$

where $N_a(\eta)$ is the lamina shape function (the subscript refers to the shell vertices) and $z(\zeta)$ is the fiber coordinate.

Using this interpolation, two sets of coordinate systems are defined [24,25]. First, the lamina coordinate systems are defined by

$$\mathbf{e}_1^l = \begin{Bmatrix} e_{11}^l \\ e_{12}^l \end{Bmatrix} = \frac{\mathbf{x}_{,\eta}}{\|\mathbf{x}_{,\eta}\|}, \quad (17)$$

$$\mathbf{e}_2^l = \mathbf{e}_3 \times \mathbf{e}_1^l = e_{11}^l \mathbf{e}_2 - e_{12}^l \mathbf{e}_1 \quad (18)$$

with

$$\mathbf{e}_1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}, \quad \mathbf{e}_2 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \quad (19)$$

leading immediately to the global to lamina transformation matrix \mathbf{q} .

$$\mathbf{q} = [q_{ij}] = [\mathbf{e}_1^l \ \mathbf{e}_2^l]^t : \text{global} \rightarrow \text{lamina}. \quad (20)$$

The fiber basis vectors are defined as

$$\mathbf{e}_2^f = \hat{\mathbf{X}}, \quad (21)$$

$$\mathbf{e}_1^f = \mathbf{e}_2^f \times \mathbf{e}_3 = \hat{X}_2 \mathbf{e}_1 - \hat{X}_1 \mathbf{e}_2 \quad (22)$$

and result in the fiber to global transformation matrix \mathbf{s} .

$$\mathbf{s} = [s_{ij}] = [\mathbf{e}_1^f \ \mathbf{e}_2^f] : \text{fiber} \rightarrow \text{global}. \quad (23)$$

In the initial development of the Eulerian shell formulation, the material interfaces have been defined to be parallel, which implicitly results in $\mathbf{s}^T = \mathbf{q}$, a simplification that was explicitly introduced by Belytschko and Tsay [26].

The kinematics of the shell are derived from the isoparametric hypothesis (in the continuum quadrilateral), i.e., the velocity field can be expressed as

$$\mathbf{v}(\eta, \zeta) = \bar{\mathbf{v}}(\eta) + \mathbf{V}(\eta, \zeta) = N_a(\eta)\bar{\mathbf{v}}_a - z(\zeta)N_a(\eta)\dot{\theta}_a\mathbf{e}_{1a}^f, \quad (24)$$

where fiber inextensibility has been assumed. As noted previously, the thickness is defined by the interface reconstruction, and is therefore dynamically updated each time step to automatically account for large membrane strains.

3.1.1. Strain and spin rate tensors

The strain and spin rate tensors (\mathbf{D} and \mathbf{W} , respectively) are:

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}, \quad (25)$$

$$\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^t), \quad (26)$$

$$\mathbf{W} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^t) \quad (27)$$

and the discrete expressions for these tensors are obtained from Eq. (24). The matrix \mathbf{B} is defined by

$$\mathbf{D} = [\mathbf{B}_a] \left\{ \begin{array}{c} \bar{\mathbf{v}}_a \\ \dot{\theta}_a \end{array} \right\} = \mathbf{B}\bar{\mathbf{v}}, \quad (28)$$

which corresponds to \mathbf{B} that would be derived for small strains.

3.2. Choice of a formulation for the rotations

The degrees of freedom for a classical (Lagrangian) shell element are two (or three) displacements and one (or two) rotations in two- (or three-) dimensions. For the Eulerian shell elements, two formulations are possible for computing the shell rotational degrees of freedom from the Eulerian element nodes. The first, formulation I, uses only the translational velocities that are found in the standard Eulerian formulation, while the second, formulation II, augments the translational velocities with a set of angular velocities.

The two formulations have been compared in the two-dimensional case. It appears that if Formulation I has the advantage of a smaller number of degrees of freedom, it presents serious drawbacks that are avoided by formulation II.

3.2.1. Formulation I: translational velocities only

For the general two-dimensional case, shown in Fig. 6, $s1$ and $s2$ are the shell vertices, while $c1$ and $c2$ are the continuum element edges they intersect. These could be consecutive edges or not, but they cannot be the same. The position of the shell relative to the quadrangle (or to any polygon) is thus defined by two edge

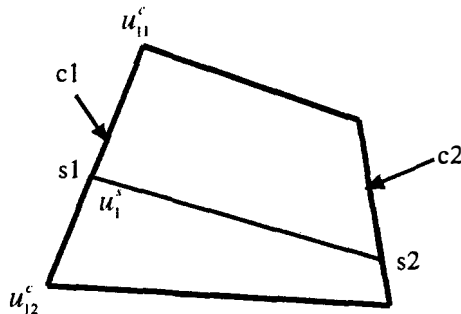


Fig. 6. General two-dimensional case.

numbers (or equivalently, two sets of two node numbers) and their two reduced coordinates η_1 and η_2 ($\in [-1, 1]$) for edges c1 and c2, respectively.

The shell vertex velocities are interpolated from the velocities of the edge nodes:

$$\mathbf{u}_a^s = N_1(\eta_a)\mathbf{u}_{a1}^c + N_2(\eta_a)\mathbf{u}_{a2}^c, \quad (29)$$

where $N_1(\eta)$ and $N_2(\eta)$ are the linear shape functions, and \mathbf{u} is the velocity vector. The superscripts s and c refer to the shell and Eulerian continuum elements and the subscripts a1 and a2 are the first and second nodes defining the continuum element edge a . The angular velocity at a shell vertex is given by the angular velocity of the continuum element edge:

$$\dot{\theta}_a^s \mathbf{e}_y = - \frac{(\mathbf{x}_{a2}^c - \mathbf{x}_{a1}^c) \times (\mathbf{u}_{a2}^c - \mathbf{u}_{a1}^c)}{\|\mathbf{x}_{a2}^c - \mathbf{x}_{a1}^c\|^2}, \quad (30)$$

which is derived from the kinematic relations of a rigidly spinning line segment in two-dimensions.

Special treatment of corners: With this approach, special consideration is required when a shell node coincides with a node of the continuum element, as illustrated in Fig. 7.

In this case, the challenge is to calculate a uniquely defined angular velocity from the continuum edges $a-$ and $a+$. One possible solution uses a linear combination of the two angular velocities based on the relative orientation of the shell and the edges,

$$\dot{\theta}_a^s = \frac{\phi_1}{\phi_1 + \phi_2} \dot{\theta}_{a-}^c + \frac{\phi_2}{\phi_1 + \phi_2} \dot{\theta}_{a+}^c, \quad (31)$$

which ensures that if the shell is aligned with one of the edges, the angular velocity is computed from the other edge.

Continuity problems: Even using Eq. (31) for the angular velocities, this formulation leads to a severe continuity problem. When two shells are connected through a continuum element edge, the C^0 -continuity of the velocity is guaranteed since the angular velocity is computed from the same nodal velocities on both sides (Fig. 8(a)). However, when they are connected through a continuum element node, the C^0 -continuity is no longer ensured because the rotations are computed from different nodal velocities on each side (Fig. 8(b)). In fact, the connection through a node introduces a spurious hinge mechanism, which can ruin the solution.

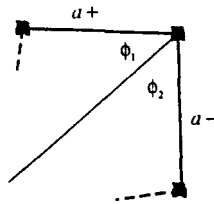


Fig. 7. Special case where a shell element node coincides with a continuum element node.

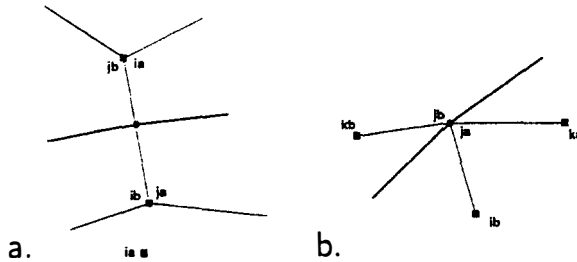


Fig. 8. Connection between two consecutive shells: (a) connection through an edge; (b) connection through a node.

3.2.2. Formulation II: adding angular velocities as solution variables

In this formulation, both the shell translational and rotational velocities are interpolated from the continuum element nodes in the same manner:

$$\mathbf{u}_a^s = N_1(\eta_a)\mathbf{u}_{a1}^c + N_2(\eta_a)\mathbf{u}_{a2}^c, \quad (32)$$

$$\dot{\theta}_a^s = N_1(\eta_a)\dot{\theta}_{a1}^c + N_2(\eta_a)\dot{\theta}_{a2}^c, \quad (33)$$

where $\dot{\theta}_{ai}^c$ are the additional rotational degrees of freedom added to the continuum element. All the problems encountered in the previous formulation vanish because there is no ambiguity about the shell angular velocity when a shell node is positioned on a continuum element node.

4. Stress update

4.1. Stress rate

The rotational contribution in the Jaumann rate is added to the stress before evaluating the constitutive model,

$$\hat{\sigma}^{n+1} = \sigma^n + (\mathbf{W}\sigma^n - \sigma^n\mathbf{W})\Delta t, \quad (34)$$

where the superscripts on the stress tensor refer to the updated $n + 1$ and reference n configurations, and Δt is the increment in time.

The rotational update is applied in the global coordinate system, where the stresses and other history variables are stored. Among other things, this allows for transport in the Eulerian step (following the Lagrangian step described here). After the rotational update, the stress tensor has to be rotated from the global coordinate system to the local (lamina) coordinate system,

$$\hat{\tau}^{n+1} = \mathbf{q}\hat{\sigma}^{n+1}\mathbf{q}^t, \quad (35)$$

where the material update can be performed according to the constitutive model,

$$\tau^{n+1} = \hat{\tau}^{n+1} + \Delta\tau, \quad (36)$$

and the zero normal stress condition is enforced. The updated stress tensor is then rotated back to the global coordinate system,

$$\sigma^{n+1} = \mathbf{q}^t\tau^{n+1}\mathbf{q} \quad (37)$$

before storing it for transport.

4.2. Plane stress update algorithm

The plane stress and plane strain implementations of a particular constitutive model are usually independent in finite element programs. To avoid the duplication of the coding during the development of constitutive models, the plane stress update algorithm for the shell element consists of an iteration loop that wraps around the plane strain material library. While this approach reduces the computational efficiency of the material library, the percentage increase in the total run time is negligible because the stress updates account for only a few percent of the total CPU time.

Two iteration methods are used to assure a converged solution. Both iterations methods require two initial values for the increment in the normal strain, $\Delta\epsilon_{nn}$. The first value assumes fully plastic flow:

$$\Delta\epsilon_{nn} = -\Delta\epsilon_{tt} - \Delta\epsilon_{33}, \quad (38)$$

where $\Delta\epsilon_{tt}$ is the strain increment in the lamina direction. The second initial value assumes the response is elastic:

$$\Delta\epsilon_{nn} = -\frac{1}{K + 4G/3}(\sigma_{nn} + (K - 2G/3)\{\Delta\epsilon_{uu} + \Delta\epsilon_{33}\}), \quad (39)$$

where K and G are the instantaneous moduli calculated by the equations of state and the strength model. The two values bound the solution, which is a requirement for the second iteration method.

Secant iteration, where i is the iteration counter,

$$\Delta\epsilon_{nn}^{i+1} = \Delta\epsilon_{nn}^i - \frac{\Delta\epsilon_{nn}^i - \Delta\epsilon_{nn}^{i-1}}{\sigma_{nn}^i - \sigma_{nn}^{i-1}} \Delta\sigma_{nn}^i \quad (40)$$

is performed for up to five iterations before resorting to the back-up strategy. In the majority of cases, less than three iterations are required for convergence. The superscript $n + 1$ has been eliminated to avoid confusion with the iteration counter.

A simple modification to the secant iteration, Regula Falsi [27], is guaranteed to converge if the initial two guesses bracket the solution. Instead of keeping the last two iterations for calculating the secant, Regula Falsi keeps two values that bracket the solution. Given the bounding interval $[\Delta\epsilon_{nn}^l, \Delta\epsilon_{nn}^r]$, and the associated function values σ_{nn}^l and σ_{nn}^r , the bounding interval is updated with $\Delta\epsilon_{nn}^l$ by

$$\text{if } \sigma_{nn}^l \cdot \sigma_{nn}^i > 0, \quad \text{then } \Delta\epsilon_{nn}^l = \Delta\epsilon_{nn}^i \quad (41)$$

$$\text{else } \Delta\epsilon_{nn}^r = \Delta\epsilon_{nn}^i \quad (42)$$

The rate of convergence is not guaranteed, however, and for the types of nonlinearities that occur in plasticity, the rate is often unacceptably slow (see Fig. 9), making it an unacceptable back-up strategy.

The back-up iteration strategy is Ridder's method [27], which replaces the linear approximation of the function on the interval with one scaled by an exponential function:

$$\sigma_{nn}(\Delta\epsilon_{nn}) = \exp(\Delta\epsilon_{nn})(m\Delta\epsilon_{nn} + b), \quad (43)$$

$$m = \frac{\sigma_{nn}^r - \sigma_{nn}^l}{\Delta\epsilon_{nn}^r - \Delta\epsilon_{nn}^l}, \quad (44)$$

$$b = \sigma_{nn}^l - m\Delta\epsilon_{nn}^l. \quad (45)$$

Ridder's method requires an extra evaluation of the function at the middle of the interval, $\Delta\epsilon_{nn}^m = (\Delta\epsilon_{nn}^l + \Delta\epsilon_{nn}^r)/2$, to provide the additional information needed to solve the third unknown, a . A small amount of algebraic manipulation yields the update formula:

$$\Delta\epsilon_{nn} = \Delta\epsilon_{nn}^m + \text{sign}(\sigma_{nn}^r - \sigma_{nn}^l)(\Delta\epsilon_{nn}^m - \Delta\epsilon_{nn}^r) / \sqrt{|(\sigma_{nn}^m)^2 - \sigma_{nn}^l \sigma_{nn}^r|}, \quad (46)$$

which does not require the evaluation of the exponential function.

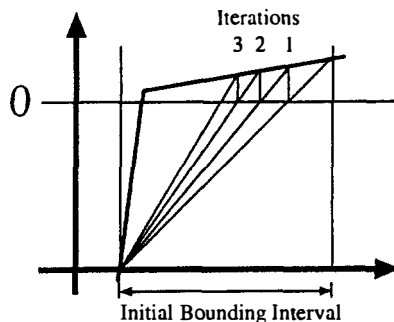


Fig. 9. The bilinear response of plasticity exhibits slow convergence with Regula Falsi iteration.

To reduce the number of iterations required for Ridder's method, the original bounding interval is reduced whenever possible during the second iteration by using the logic of Eq. (41).

5. Implementation

5.1. Transformation matrices

The transformation matrix \mathbf{T} links the Eulerian element velocities \mathbf{v}^e to the shell velocities \mathbf{v}^s :

$$\mathbf{v}^s = \mathbf{T}\mathbf{v}^e \quad (47)$$

and its transpose links their respective forces.

In two-dimensions, \mathbf{T} is 6×8 with formulation I and 6×12 with formulation II. They are easily obtained from Eqs. (29) and (30), or from Eqs. (32) and (33), respectively.

5.2. Internal forces and the stiffness matrix

The internal forces vector is obtained by

$$\mathbf{f}^{\text{int}} = \mathbf{T}^t \int_{\text{shell}} \mathbf{B}^t \boldsymbol{\sigma} dV_{\text{shell}}. \quad (48)$$

and for the geometrically linear framework, which is used later in the eigenvalue analyses, the stiffness matrix is

$$\mathbf{K} = \mathbf{T}^t \int_{\text{shell}} \mathbf{B}^t \mathbf{C} \mathbf{B} dV_{\text{shell}} \mathbf{T} \quad (49)$$

$$= \mathbf{T}^t \mathbf{K}_{\text{shell}} \mathbf{T}, \quad (50)$$

where \mathbf{C} is the constitutive tangent matrix in the local coordinate system.

5.3. Mass matrix

The consistent mass matrix is given by

$$\mathbf{M} = \mathbf{T}^t \int_{\text{shell}} \rho \mathbf{N}^t \mathbf{N} dV_{\text{shell}} \mathbf{T} \quad (51)$$

$$= \mathbf{T}^t \mathbf{M}_{\text{shell}} \mathbf{T}, \quad (52)$$

where ρ is the density and \mathbf{N} is the shape function matrix, and describes the interpolation of the shell kinematics. The rotational terms are scaled so that the critical time step is controlled by element length and not by element thickness [28,29].

The mass matrix is lumped for use with an explicit time integration scheme. Instead of lumping $\mathbf{M}_{\text{shell}}$ then using the transformation given in Eq. (52), one fourth of the total shell mass is distributed to the four Eulerian element nodes in the same manner as any other material contained within the element, and similarly for the rotational inertias. This mass distribution is required by the momentum advection algorithm. The staggered transport masses, δm , used in the momentum transport were derived on the assumption that the element mass was distributed equally to each of its four nodes [30]. If a lumped $\mathbf{M}_{\text{shell}}$ is introduced into Eq. (52), the momentum transport becomes unstable, as illustrated in Section 7. Consistent, conservative momentum transport on a staggered mesh is a nontrivial issue [18], and its extension to problems with mass matrices of the type defined by Eq. (52) will be discussed in a future paper.

5.4. Area weighting for axisymmetric problems

The axisymmetric version of the Eulerian shell element uses an area-weighted Petrov–Galerkin formulation to preserve spherical symmetry. This method is commonly used for continuum elements in hydrocodes [31], but has never been implemented for shell elements to our knowledge. The area weighting leads to an expression for the internal force vector, \mathbf{f}^{int} :

$$\mathbf{f}^{\text{int}} = \mathbf{T}^t \left(\int_{\text{shell}} \mathbf{B}^t \boldsymbol{\sigma} \, dA_{\text{shell}} - \int_{\text{shell}} \mathbf{N}^t \left\{ \begin{matrix} \sigma_{rr} - \sigma_{\theta\theta} \\ \sigma_{rz} \end{matrix} \right\} dA_{\text{shell}} \right), \quad (53)$$

where \mathbf{B} is the same matrix used in plane strain, and $\sigma_{\theta\theta}$ does not appear in the vector $\boldsymbol{\sigma}$ in Eq. (53). This axisymmetric shell formulation is completely consistent with the area-weighted formulation of the Eulerian continuum axisymmetric elements.

The consistent axisymmetric mass matrix is

$$\mathbf{M} = \mathbf{T}^t \int_{\text{shell}} \rho \mathbf{N}^t \mathbf{N} \, dA_{\text{shell}} \mathbf{T} \quad (54)$$

and is thus time dependent (as is the area-weighted lumped mass matrix); however, both are inexpensive to evaluate.

5.5. Rank deficiency problems

The equivalent classical (Lagrangian) shell element has six degrees of freedom, its null-space is composed of three rigid-body modes, and therefore, the rank of its stiffness matrix is three. The rank of the Eulerian shell stiffness is also three, but its dimension is eight or twelve, depending on the rotational formulation. Therefore, in addition to the three rigid-body modes, there are spurious zero energy (hourglass) modes linked to the nature of the transformation matrix \mathbf{T} . Formulation I has two spurious modes, while formulation II has four more, for a total of six. These modes are illustrated in Fig. 10, where the first two (on the left) are common to both formulations. In practice, the modes did not occur in the example calculations, perhaps partially due to the monotonic momentum transport, which tends to filter out zero energy modes.

The treatment of the zero energy modes is simplified by recognizing that they originate in the interpolation of the continuum velocities to the shell nodes. For a shell node with an isoparametric coordinate s

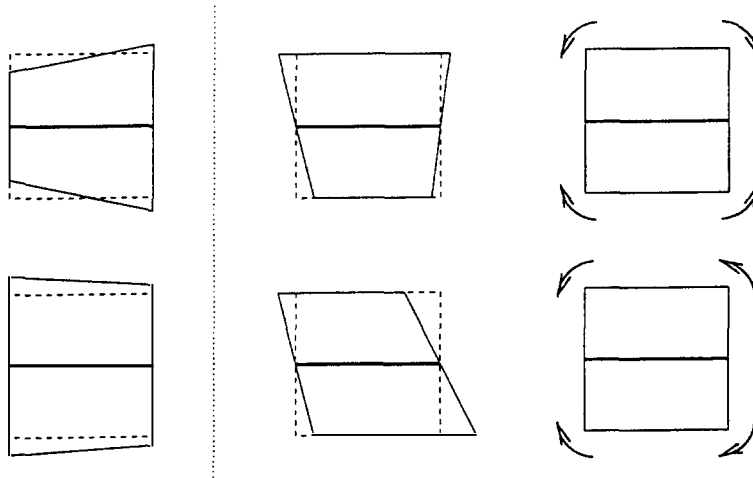


Fig. 10. Spurious modes of the two-dimensional shell element.

on an edge defined by continuum nodes a and b , the zero energy mode, which is localized to that edge, has the form:

$$\mathbf{h} = \begin{Bmatrix} q_a \\ q_b \end{Bmatrix} = \frac{\alpha}{\sqrt{2(s^2 + 1)}} \begin{Bmatrix} s + 1 \\ s - 1 \end{Bmatrix}, \quad (55)$$

where α is the amplitude of the zero energy mode and q may be either the translational or rotational velocities. A rank one matrix, \mathbf{H} ,

$$\mathbf{H} = \beta \mathbf{h} \otimes \mathbf{h}, \quad (56)$$

$$\mathbf{F}_H = -\mathbf{H}\mathbf{q}, \quad (57)$$

is multiplied by the individual the translational and rotational velocity components to produce a viscous force opposing the zero energy modes, where β is the damping coefficient [32,31].

6. Computation of a stable time step

The stable time step in both Lagrangian and Eulerian hydrocodes is typically based on local (element level) time step bounds that can be cast in the form:

$$\Delta t_{\text{crit}} = \mathcal{S} \min_{i=1, \text{NEL}} \left(\frac{\ell_i}{c_i} \right), \quad (58)$$

where \mathcal{S} is a constant that depends on the element formulation, c_i is the acoustic wave velocity and ℓ_i is a characteristic length of element i . This approach is problematic with the Eulerian shell formulation because the minimum characteristic length of the Eulerian shells is likely to become zero somewhere as the shell structure is transported through the mesh.

A stable time step size estimate has been derived by iteratively solving the global eigenvalue problem for the highest natural frequency [7]. This leads to a more accurate, larger time step size, and thus decreases the cost of the computation. Of particular importance for the Eulerian shell formulation, the stable time step size does not go to zero as the shell structure moves through the mesh, and as shown below, the time step size is nearly independent of the structure's location within the mesh.

7. Example calculations

7.1. Eigenvalues analysis

The following eigenvalues analyses illustrate the problems inherent to formulation I, and how formulation II avoids them.

7.1.1. Plate in planar coordinates

An eigenvalue analysis of a plate of unit depth, fixed on one end and free on the other, illustrates the relative performances of the two rotational formulations. The plate has a width L , a thickness $0.1L$, and it is made of an elastic material having a density of 1.0, a bulk modulus of 833.0 and a shear modulus of 385.0.

Horizontal model: In the horizontal model, the plate is discretized by one row of elements, as illustrated in Fig. 11. The boundary conditions are enforced by fixing the left edge. In this case, the connection between the shell elements is correct for both of the rotational formulations. The results obtained for both formulations match those from an equivalent Lagrangian model, as illustrated in Fig. 11.

Diagonal model: The plate lies on the diagonal of a square mesh, as illustrated in Fig. 12. The boundary conditions for formulation I require the left and bottom edges of the element in the lower, left-hand corner to be fixed. As shown in Fig. 12, the accuracy of formulation I is poor, while II maintains its accuracy.

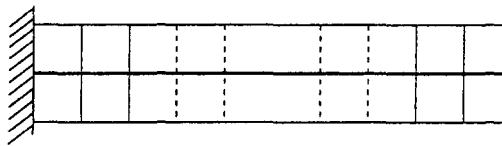
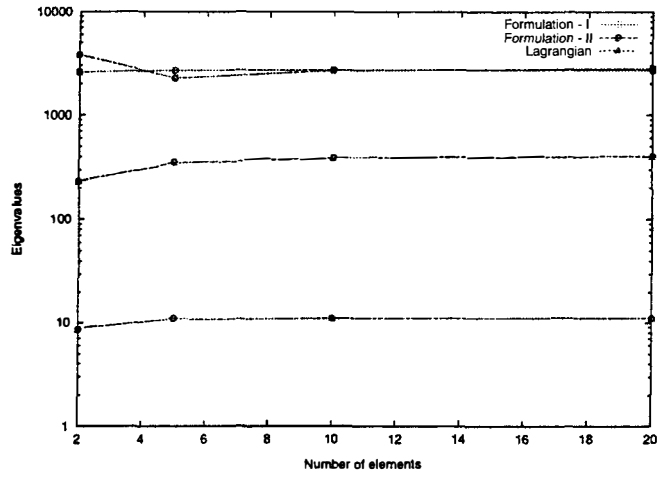


Fig. 11. First four eigenvalues for horizontal model.

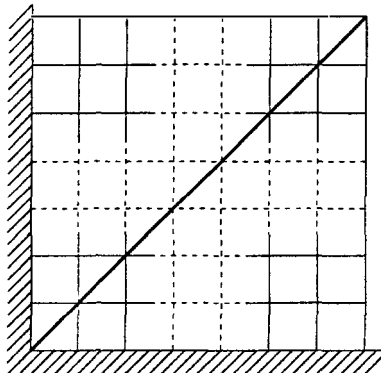
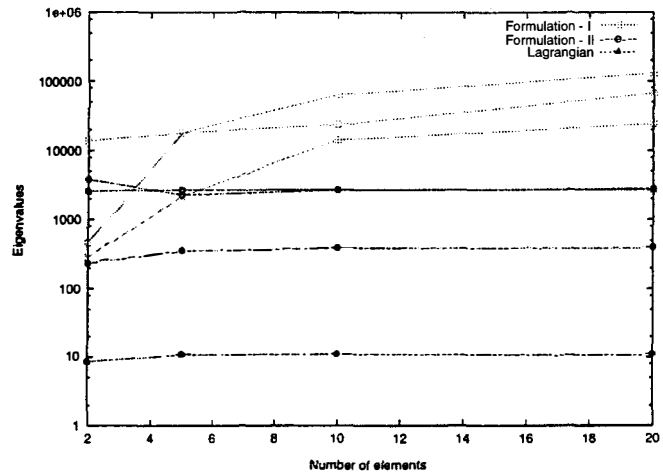


Fig. 12. First four eigenvalues for diagonal model.

7.1.2. Cylindrical shell moving through an Eulerian grid

This example demonstrates how the eigenvalues of a cylindrical shell are influenced by a changing discretization due to a rigid body translation of the cylinder relative to the Eulerian mesh. Ideally, they should be independent of the cylinder's position. The eigenvalues for formulation II, but not I, are independent of the the six positions as indicated in Fig. 13. For the positions where formulation I exhibits large errors, some shell nodes coincide with continuum nodes, illustrating the formulation's deficiencies that were discussed in a previous section.

7.2. Nonlinear analysis

7.2.1. Explosively loaded plate in plane strain

This example demonstrates large deformation fluid–structure interaction. The initial configuration, shown in Fig. 14, consists of a 1.0 cm thick steel plate that spans the 1.0 m wide spatial domain. The shear modulus, Poisson's ratio, yield stress, and hardening modulus for the plate are 0.77 Mbar, 0.3, 7.7, and 20.5 Kbar, respectively. Water fills the region above the plate and the region below the plate is empty. A bubble of ideal gas, with $\gamma = 5/3$, occupies a small region 5.0 cm in diameter in the water-filled region, and provides the dynamic loading. Its initial density is 2.0 g/cm³ and its initial internal energy is 0.3 Mbar cm³/cm³.

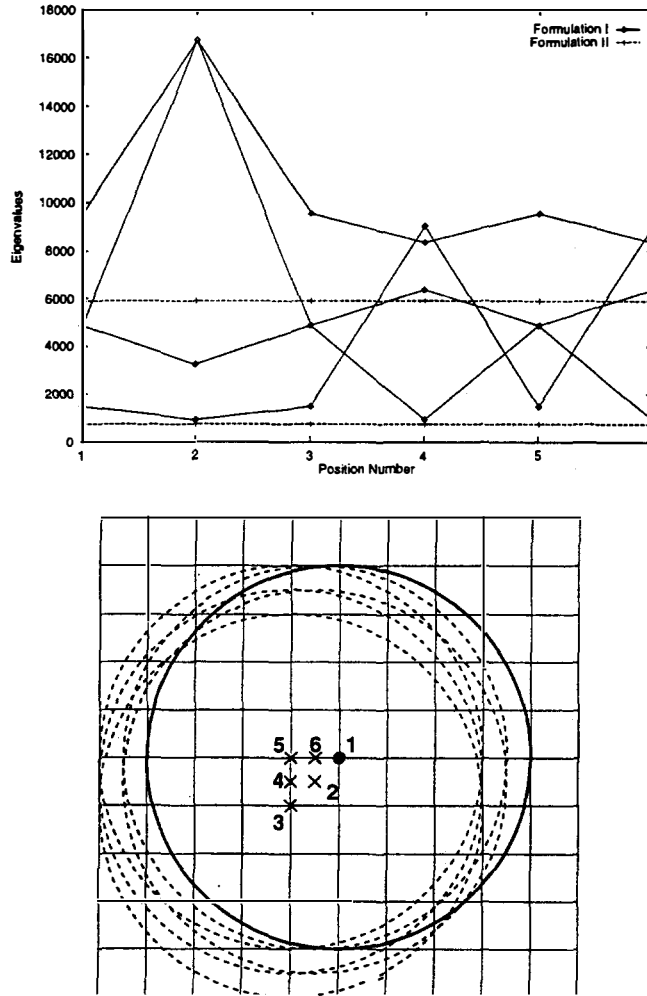


Fig. 13. Evolution of the first four eigenvalues of a moving shell.

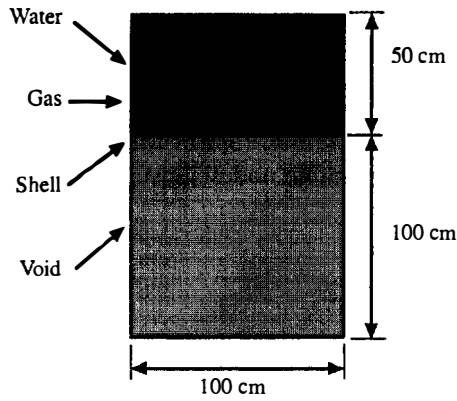


Fig. 14. Initial configuration of the explosively loaded shell.

The solution was obtained using the new shell formulation with four integration points through the thickness on uniform meshes of 20×30 and 40×60 elements covering a domain 1.0 m wide by 1.5 m deep. For comparison, the solutions were also calculated by modeling the steel as a solid on meshes of 100×100 and 200×150 elements covering a domain $1.0 \times 1.0 \text{ m}^2$. In the second mesh, the water was resolved with 50 elements in the vertical direction and the region below the plate (which the plate traverses during the calculation) was resolved with 100 elements. As shown in Fig. 15 for plane strain, the coarse solid model is unable to resolve the shell structural response at 1.0 ms , and the solutions obtained by the fine solid model and the coarse Eulerian shell model are in close agreement. The solution on the coarse mesh obtained with the mass matrix in Eq. 52 exhibits severe instabilities, as shown in Fig. 16. A sequence showing the evolution of solution out to 2.0 ms for the fine Eulerian shell model is shown in Fig. 17.

The coarse and fine mesh shell element solutions required 6 and 39 CPU seconds, respectively on a DEC Alpha workstation to reach a simulation time of 1.0 ms , while the fine mesh solution obtained with the continuum elements required 1447 CPU seconds.

7.2.2. Axisymmetric explosively loaded plate

An axisymmetric solution, shown in Fig. 18, demonstrates the thinning and perforation of the Eulerian shell. The initial density and internal energy of the gas are increased to 16.0 g/cm^3 and $0.6 \text{ Mbar cm}^3/\text{cm}^3$, respectively to overcome the stiffening of the plate due to the hoop stress.

7.2.3. Shell contact

This example shows that contact between shells is adequately handled by the standard mixture theory (i.e., no special treatment is required). This problem is identical to the first calculation of an explosively loaded plate except that two plates, separated by a gap, are used to model a double-hulled vessel. The mesh is 30×45 elements (Fig. 19).

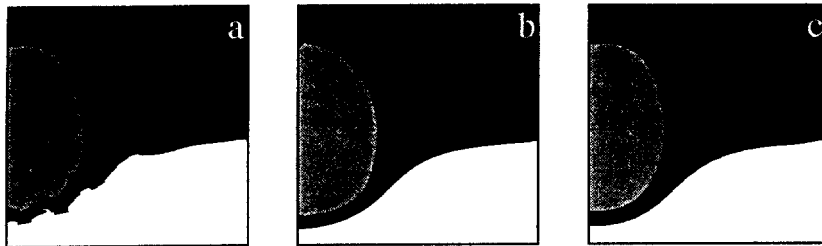


Fig. 15. Explosively loaded plate at 1.0 ms using a continuum model and the new shell formulation: (a) coarse mesh continuum solution; (b) fine mesh continuum solution; (c) coarse mesh shell solution.

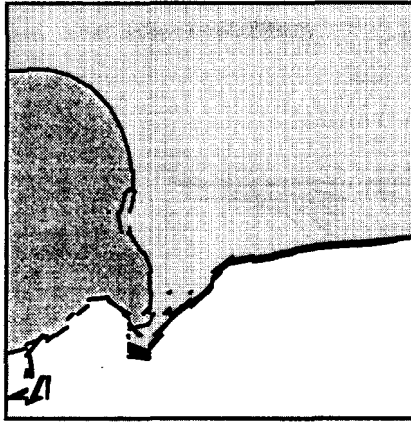


Fig. 16. Explosively loaded plate using the new shell formulation with a consistent mass matrix at 1.0 ms.

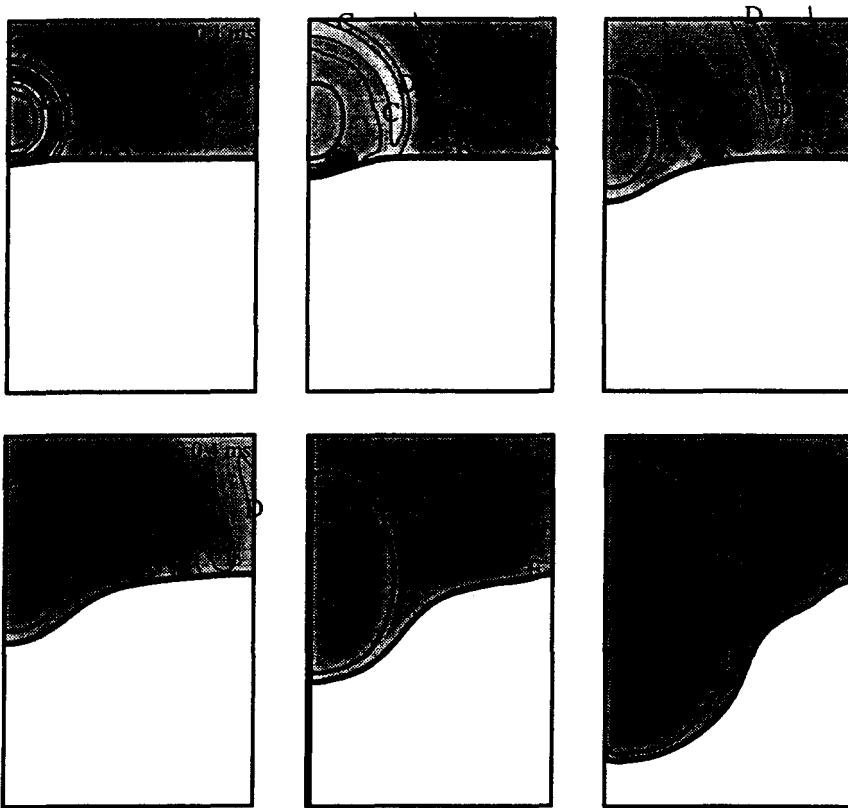


Fig. 17. Sequence of the evolution of the explosively loaded plate in plane strain using the new shell formulation on the fine mesh.

This calculation uses the mean strain rate mixture theory with void collapse, the simplest mixture theory in the code. The mean strain rate mixture theory sets the strain rate in each material within an element to the mean strain rate of the element. For elements which are partially empty (containing void), the void is compressed the same amount as the other materials, and therefore its volume in the element is reduced only until the volumetric compression in the other materials is sufficient to resist additional compression. Without any modifications, the two shells would never touch. A modified version of the mean strain rate mixture theory preferentially collapses, or squeezes out, the void before any compression is applied to the other materials.

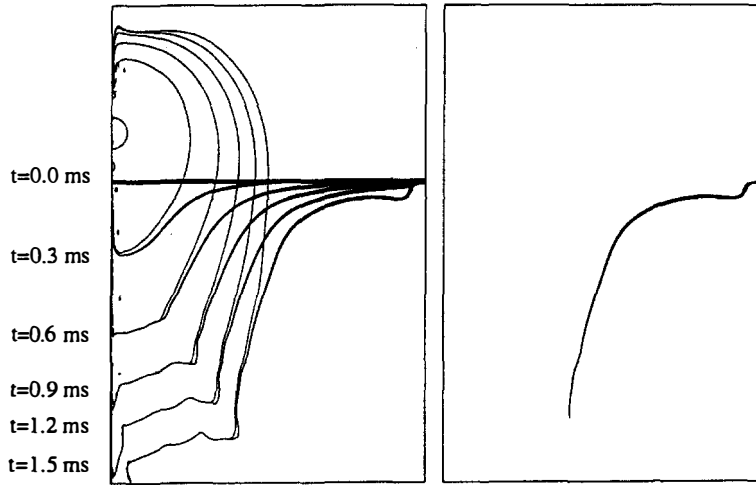


Fig. 18. Sequence of the evolution of the axisymmetric plate and gas bubble on a fine mesh. The right-hand figure shows the final state of the perforated plate.

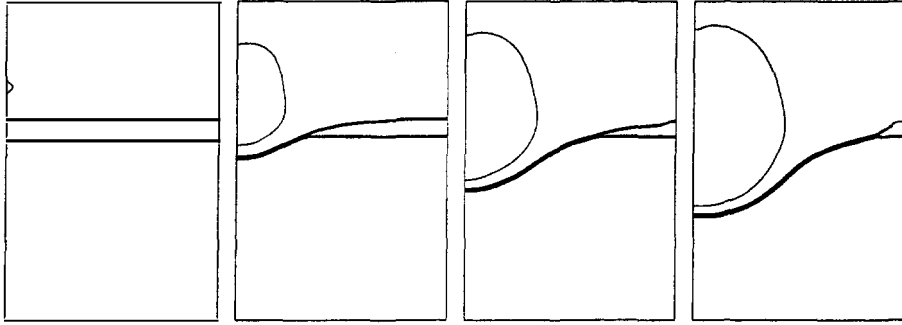


Fig. 19. Evolution of a double hull structure and a gas bubble.

7.3. Weld failure

Underwater structures can fail at their welds, and this example demonstrates the ability of the Eulerian shell formulation to model weld failure. The structure again consists of an assembly of 1.0 cm thick plates with the same material properties as in the previous examples. The three welds, with their locations defined by tracer particles, fail when the material reaches an equivalent plastic strain of 0.25, see Fig. 20. This

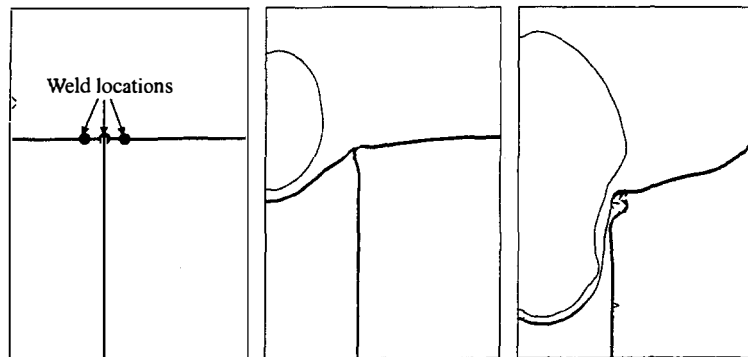


Fig. 20. Welded structure sequence: 0, 0.5 and 1.5 ms.

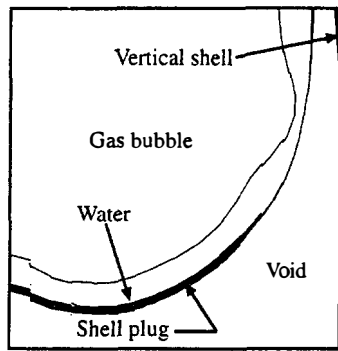


Fig. 21. Close up of the failed shell at 1.5 ms.

failure criteria was chosen for this example calculation for its simplicity, and better criteria can be substituted.

The T intersection of the two plates is not resolved perfectly because of the limitations of the interface reconstruction algorithm. There are methods for improving the resolution of T intersections [22], but they have not been implemented at this stage of our research.

As shown in Fig. 20, the weld which is subjected to the highest moment fails while the others remain intact. A large piece of the horizontal plate is pushed down the channel formed by the vertical plate by the expanding gas bubble. A close-up of the final state shows the detached section of the structure (Fig. 21). Small discontinuities in the shells are visible near the left edge. The right edge of the plate tapers in thickness to zero over the width of one element due to the interface reconstruction, which assumes that each material layer spans the the element. For the types of large deformation calculations shown, resolving the end of the failed material in less than one element probably is not important.

8. Summary

The theoretical basis of an Eulerian formulation for two-dimensional shells has been presented. In particular, the effects of two different formulations for rotations have been discussed and illustrated with examples in the linear range. Further applications in the nonlinear range have demonstrated the ability of the new formulation to model large deformation fluid–structure interaction, shell thinning, perforation, and weld failure.

Acknowledgements

This research was supported by the Office of Naval Research (grant No. N00014-96-1-0638).

References

- [1] S. Hancock, PISCES-2DELK theoretical manual, Technical report, Physics International, 1985.
- [2] J.U. Brackbill, B. Kothe, C. Zemach, A continuum method for modeling surface tension, *Journal of Computational Physics* 100 (2) (1992) 335–354.
- [3] D.J. Benson, A Multi-material Eulerian formulation for the efficient solution of impact and penetration problems, *Computational Mechanics* 15 (1995) 558–571.
- [4] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, *Computer Methods in Applied Mechanics and Engineering* 99 (1992) 235–394.
- [5] E. Reissner, The effect of transverse shear deformation on the bending of elastic plates, *Journal of Applied Mechanics* 12 (1945) A69–A77.
- [6] R.D. Mindlin, Influence of rotary inertia and shear on flexural motion of isotropic elastic plates, *Journal of Applied Mechanics* 18 (1951) 31–38.

- [7] D.J. Benson, Stable time step estimation for multi-material Eulerian hydrocodes, *Computer Methods in Applied Mechanics and Engineering* 167 (1998) 191–205.
- [8] A. Chorin, T.J.R. Hughes, M.F. McCracken, J.E. Marsden, Product formulas and numerical algorithms, *Commun. Pure Appl. Math.* 31 (1978) 205–256.
- [9] J.O. Hallquist, D.J. Benson, *Nonlinear Dynamic Analysis of Structures in Three-Dimensions*, DYNA3D User's Manual, Lawrence Livermore National Laboratory UCID-19592 Rev. 3, 1987.
- [10] G.R. Johnson, R.A. Stryk, User Instructions for the EPIC-2 Code, Air Force Armament Laboratory AFATL-TR-86-51, 1986.
- [11] M. Wilkins, Calculation of Elastic–Plastic Flow, *Methods in Computational Physics*, vol. 3, Academic Press, New York, 1964, pp. 211–263.
- [12] L.M. Taylor, D.P. Flanagan, PRONTO 2D – a Two-Dimensional Transient Solid Dynamics Program, Sandia National Laboratories, Technical report SAND86-0594, 1987.
- [13] T. Belytschko, J.M. Kennedy, D.F. Schoeberle, On finite element and difference formulations of transient fluid–structure problems, in: *Proceedings of Computational Methods in Nuclear Engineering*, Charleston, South Carolina, 1975, pp. IV39–IV54.
- [14] D.J. Benson, A new two-dimensional flux-limited viscosity for impact calculations, *Computer Methods in Applied Mechanics and Engineering* 93 (1991) 39–95.
- [15] D.J. Benson, A mixture theory for contact in multi-material Eulerian formulations, *Computer Methods in Applied Mechanics and Engineering* 140 (1997) 59–86.
- [16] B. Van Leer, Towards the ultimate conservative difference scheme IV: A new approach to numerical convection, *Journal of Computational Physics* 23 (1977) 276–299.
- [17] P.J. Roache, *Computational Fluid Dynamics*, Hermosa Publishers, 1976.
- [18] D.J. Benson, Momentum advection on a staggered mesh, *Journal of Computational Physics* 100 (1) (1992) 143–162.
- [19] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics* 39 (1981) 201–225.
- [20] D.L. Youngs, Time dependent multi-material flow with large fluid distortion, In: K.W. Morton, M.J. Baines (Eds.), *Numerical Methods for Fluid Dynamics*, 1982, pp. 273–285.
- [21] F.L. Addessio, J.R. Baumgardner, J.K. Dukowicz, N.L. Johnson, B.A. Kashiwa, R.M. Rauenzahn, C. Zemach, CAVEAT: A Computer Code for Fluid Dynamics Problems with Large Distortion and Internal Slip, revised edition, Los Alamos National Laboratory, 1990.
- [22] N. Johnson, Personal communication, Los Alamos National Laboratory, 1990.
- [23] D.J. Benson, Eulerian finite element methods for the micromechanics of heterogeneous materials: Dynamic prioritization of material interfaces, *Computer Methods in Applied Mechanics and Engineering* 151 (1998) 343–360.
- [24] T.J.R. Hughes, W.K. Liu, Nonlinear finite element analysis of shells: Part I. Three-dimensional shells, *Computer Methods in Applied Mechanics and Engineering* 26 (1981) 331–362.
- [25] T.J.R. Hughes, W.K. Liu, Nonlinear finite element analysis of shells: Part II. Two-dimensional shells, *Computer Methods in Applied Mechanics and Engineering* 27 (1981) 167–181.
- [26] T.B. Belytschko, C.S. Tsay, Explicit algorithms for nonlinear dynamics of shells, *ASME AMD-48* (1981) 209–231.
- [27] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, New York, 1992.
- [28] T.J.R. Hughes, W.K. Liu, I. Levit, Nonlinear dynamic finite element analysis of shells, in: W. Wunderlich, E. Stein, K.-J. Bathe (Eds.), *Nonlinear Finite Element Analysis in Structural Mechanics*, 1981, pp. 151–168.
- [29] S.W. Key, Z.E. Beisinger, The transient dynamic analysis of thin shells in the finite element method, in: *Proceedings of the Third Conference on Matrix Methods in Structural Mechanics*, Wright-Patterson Air Force Base, 1971.
- [30] A.A. Amsden, C.W. Hirt, YAQUI: An Arbitrary Lagrangian–Eulerian Computer Program for Fluid Flow at All Speeds, LA-5100, Los Alamos Scientific Laboratory, 1973.
- [31] G.L. Goudreau, J.O. Hallquist, Recent developments in large-scale finite element lagrangian hydrocode technology, *Computer Methods in Applied Mechanics and Engineering* 33 (1/3) (1982) 725–757.
- [32] D.P. Flanagan, T. Belytschko, A uniform strain hexahedron and quadrilateral with orthogonal hourglass control, *International Journal for Numerical Methods in Engineering* 17 (1981) 679–706.