Engineering Fracture Mechanics 77 (2010) 549–564

Contents lists available at ScienceDirect

Engineering Fracture Mechanics

journal homepage: www.elsevier.com/locate/engfracmech

An efficient fluid–solid coupled finite element scheme for weapon fragmentation simulations

O. Soto [a,*], J. Baum [a], R. Löhner [b]

[a] Science Applications International Corporation (SAIC), 1710 Saic Drive MS 2-6-9, McLean, VA 22102, USA
[b] Center for Computational Fluid Dynamics, College of Science, George Mason University, MS 6A2, Fairfax, VA 22030, USA

ARTICLE INFO

Article history:
Received 24 March 2009
Received in revised form 28 October 2009
Accepted 9 November 2009
Available online 13 November 2009

Keywords:
Finite elements
Fluid–solid coupling
Metal fragmentation

ABSTRACT

An efficient finite element (FE) scheme to deal with a class of coupled fluid–solid problems is presented. The main ingredients of such methodology are: an accurate Q1/P0 solid element (trilinear in velocities and constant piecewise-discontinuous pressures); a large deformation plasticity model; an algorithm to deal with material failure, cracking propagation and fragment formation; and a fragment rigidization methodology to avoid the possible numerical instabilities that may produce pieces of material flying away from the main solid body. All the mentioned schemes have been fully parallelized and coupled using a loose-embedded procedure with a well-established and validated computational fluid dynamics (CFD) code (FEFLO). A CSD and a CFD/CSD coupled case are presented and analyzed.

1. Introduction

In the analysis of complex structures subjected to blast loading and large deformations, Lagrangian FE codes with explicit time integration are a necessary tool [1,2]. This class of problems requires hundreds of thousands of time steps, and in some cases, millions of elements, which forces the use of computationally simple methodologies. Furthermore, the material failure and the possible changes of topology due to the crack advance has to be implemented in an efficient manner. Expensive remeshing procedures and/or cracking schemes could make the calculation of real problems an impossible task in terms of CPU time. Moreover, the possible numerical instabilities that may appear as a result of several fragment of elements flying away has to be taken into account. For most cases, i.e., for bomb fragmentation studies, it is not the internal stresses of the small fragments what needs to be computed in an accurate manner, but their size and shape. Hence, a fast rigidization algorithm that maintains the volume of the fragments must be available. All the mentioned schemes should be fully parallelized to obtain solutions in a realistic period of time.

The main ingredients of an efficient methodology to deal with real coupled fluid–solid blast problems will be described in this paper. First, a brief description of a large deformation plasticity constitutive material [17] is presented. Such model is motivated by a well-understood micromechanical picture of single-crystal metal plasticity. The resulting constitutive model relies on a hyperelastic characterization of the elastic material response, which avoids the drawbacks of the widely used hypoelastic models, i.e., the material isotropy, the nonzero residual strain in a closed "elastic" cycle, and in general, the lack of a stored-energy function to obtain the elastic stress tensor. In addition, the deviatoric and volumetric elastic stresses are function of tensors which are invariant against rigid rotations (tensors defined in the convective coordinates). This feature automatically makes the scheme to fulfill the objectivity [17] principle (no special objective stress rates are used).

* Corresponding author. Tel.: +1 703 676 4857; fax: +1 703 676 5323.
E-mail address: orlando.a.soto@saic.com (O. Soto).

0013-7944/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engfracmech.2009.11.003

From the numerical point of view, the element kinematic variables are fully integrated (eight integration points per element), which avoids spureos modes and its immediate consequence: the use of artificial hourglass viscosities [13]. On the other hand, the pressure field is interpolated using only one nodal point per element (piecewise constant pressures per element). Such numerical feature removes the volumetric locking and accurately approximates the material incompressibility in the plastic range [14]. Also, a pressure smoothing is applied in each time step, to avoid the appearance of check-board [14] pressure modes, which may be presented in some cases using the Q1/P0 element (this element does not strictly fulfill the incompressible BB condition) [14].

In Section 3, the material fracture model is exposed. For this purpose, the elements conforming the FE mesh are checked using some failure criteria, i.e., the maximum effective plastic strain, or some damage variable. The most failed node and face of the element are then chosen to define the failure plane. The new node is created by duplicating the most failed one, and by redefining the connectivity of the elements located at one side of the failure plane. Such methodology produces a cracking propagation algorithm that does not require expensive remeshing schemes. This last feature is highly desirable when dealing with real problems, where millions of elements and thousands of time steps may be required. However, it may fails to produce real fragmentation patterns for some cases, which motivates to implement semi-empirical agglomeration algorithms i.e., Mott's theory [27,28].

A simple rigidization algorithm based on the number of elements that a fragment of material contains has been implemented to avoid numerical instabilities and to save CPU time. All the mentioned schemes have been fully parallelized and coupled using a loose-embedded procedure with the well-established and validated computational fluid dynamics (CFD) code, FEFLO98. Finally, A CSD and a CFD/CSD coupled cases are presented.

2. Constitutive equation and numerical approximation

As mentioned in Section 1, the plasticity model used in this work is motivated by a well-understood micromechanical picture of single-crystal metal plasticity [17]. The resulting constitutive model relies on a hyperelastic characterization of the elastic material response, which avoids the drawbacks of the widely used hypoelastic models, i.e., the material isotropy, the nonzero residual strain in a closed "elastic" cycle, and in general, the lack of a stored-energy function to obtain the elastic stress tensor. Moreover, the deviatoric and volumetric elastic stresses are function of tensors which are invariant against rigid rotations (tensors defined in the convective coordinates). This feature automatically enforces objectivity [17] on the scheme.

Also, it can be demonstrated that the hypoelastic models based on commonly used objective stress rates, like the Jaumann–Zaremba and Green–McInnis–Naghdi stress rates, do not result in a straightforward generalization of the classical radial return algorithm when one tried to develop a J_2 plasticity theory for large deformation cases (see [17]). The only models which result in such a generalization are the ones based on the Lie derivate of the Kirchhoff stress. However, many authors copy as it is the structure of the radial return map for small deformation plasticity, to use it with hypoelastic models that are not based on the mentioned derivate. Our experience indicates that such algorithms violates in an unacceptable manner the isochoric (incompressible) character of the plastic flow when non-structured meshes are used: This is, the elements tend to have negative volume and their edges tend to cross at high explosive loads (pressures). However, the use of non-structured meshes is almost mandatory for this type of problems, not only because they allow the automatic generation of the computational grid by using a tetrahedra mesh generator (then each tetrahedra is divided in four hexahedral elements see Fig. 1), but also because the unstructured topology of the mesh automatically includes a random behavior to the fracture algorithm, which makes the simulations more realistic. This point is demonstrated below in the numerical simulations.

Ref. [17] presents a detailed description of the hyperelastic plasticity model that has been used in this work. However, a brief summary is shown below for completeness. The stored-energy function has the form:

$$W = U(J^e) + \overline{W}(\overline{\boldsymbol{b}^e}) \tag{1}$$

$$U(J^e) = \frac{1}{2}\kappa\left[\frac{1}{2}(J^{e2}-1) - \ln J^e\right] \tag{2}$$

$$\overline{W}(\overline{\boldsymbol{b}^e}) = \frac{1}{2}\mu\left(\mathrm{tr}[\overline{\boldsymbol{b}^e}] - 3\right) \tag{3}$$

which results in the following stress–strain relationships:

$$\boldsymbol{\tau} = J^e p\boldsymbol{I} + s \tag{4}$$

$$p = \frac{\kappa}{2}(J^{e2}-1)/J^e \tag{5}$$

$$\boldsymbol{s} = \mu\mathrm{dev}[\overline{\boldsymbol{b}^e}] \tag{6}$$

Above, τ is the Kirchhoff stress tensor, p the mechanical pressure and $\boldsymbol{s}$ is the deviatoric part of the stress tensor. J^e is the determinant of the elastic part of the deformation gradient tensor $\boldsymbol{F}^e$, κ the material bulk modulus and μ is the shear modulus. Finally, $\overline{\boldsymbol{b}^e}$ is the volume-preserving part of $\boldsymbol{b}^e$ (elastic left Cauchy–Green tensor), given by:

$$\overline{\boldsymbol{b}^e} = J^{e-2/3}\boldsymbol{F}^e\boldsymbol{F}^{eT}. \tag{7}$$

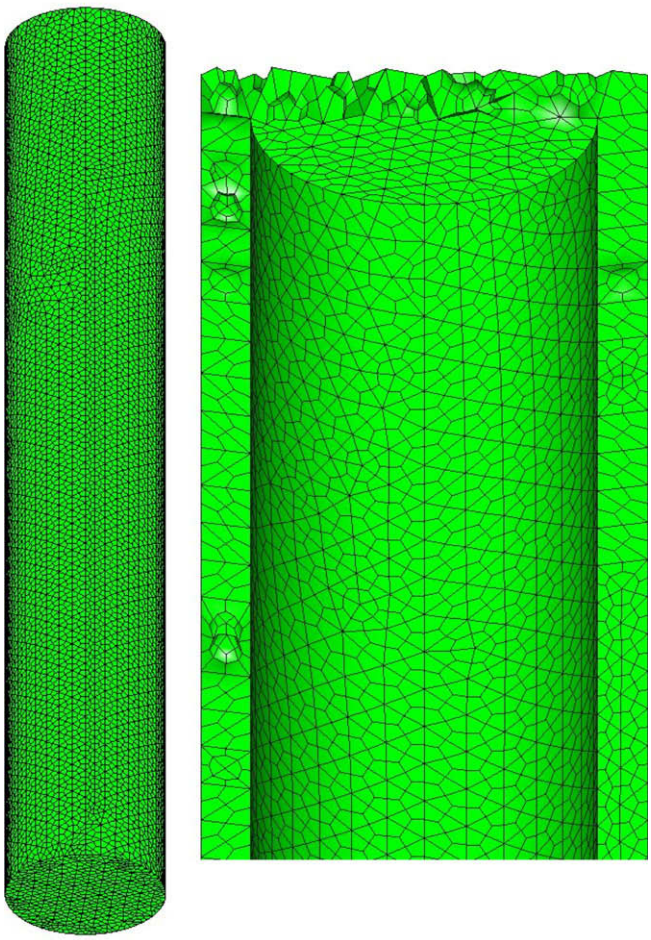

Fig. 1. From left to right: finite element mesh of ≈100,000 hexahedrals; vertical cut showing the interior top part of the weapon.

The yield condition is a classical Mises–Hubert type, but formulated in terms of the Kirchhoff stress tensor:

$$f(\tau,\alpha) := ||\boldsymbol{s}|| - \sqrt{\frac{2}{3}}[\sigma_Y + K\alpha] \leqslant 0 \tag{8}$$

where σ_Y denotes the flow stress, K denotes the isotropic hardening modulus, and α the hardening parameter. $||\boldsymbol{s}||$ is the norm of $\boldsymbol{s} = \sqrt{s_{ij}s_{ij}}$ (repeated index sum). It is clear that non-linear hardening laws $(K(\alpha))$, dynamic increase factors for σ_Y and empirical based models for the flow stress (σ_y) are easily accommodated in the formulation. As an example, the widely used Johnson–Cook [29,30] model for the flow stress in metals, which takes into account the effect of high strain rate hardening and temperature softening, was easily implemented in this plasticity framework.

Finally, the plasticity model is completed, as usual, with the following associative flow rule, hardening law, Kuhn–Tucker and consistency conditions:

$$\boldsymbol{L}_v\boldsymbol{b}^e = -\frac{2}{3}\gamma \mathrm{tr}[\boldsymbol{b}^e]\boldsymbol{n} \tag{9}$$

$$\boldsymbol{n} = \frac{\boldsymbol{s}}{||\boldsymbol{s}||} \tag{10}$$

$$\dot{\alpha} = \sqrt{\frac{2}{3}}\gamma \tag{11}$$

$$\gamma \geqslant 0, \quad f(\tau,\alpha) \leqslant 0, \quad \gamma f(\tau,\alpha) = 0 \tag{12}$$

$$\gamma \dot{f}(\tau,\alpha) = 0. \tag{13}$$

Above, $\boldsymbol{L}_v\boldsymbol{b}^e$ is the Lie derivative of $\boldsymbol{b}^e$ and γ is the consistency parameter. $\dot{A}$ is the material time derivative of A.

2.1. Discretization of the dynamic equation

The dynamic equilibrium equation written on the spatial configuration at time $t(\Omega_t)$ is given by:

$$\rho\ddot{\boldsymbol{u}} - \nabla\cdot\boldsymbol{\sigma} = \boldsymbol{F} \text{ in } \Omega_t \tag{14}$$

plus compatible initial and boundary condition. In Eq. (14) ρ is the density, $\boldsymbol{\sigma}$ the Cauchy stress tensor, $\boldsymbol{u}$ the displacement field, $\ddot{\boldsymbol{u}}$ the acceleration and $\boldsymbol{F}$ is the body forces. All the terms are evaluated in the actual configuration.

The equilibrium variational form at the spatial configuration is obtained by multiplying Eq. (14) times a test function $\boldsymbol{\delta u} \in \boldsymbol{V}$ that fulfills the imposed boundary condition and continuity requirements. After integrating by parts the internal stress term in the usual manner, the following is obtained:

$$\int_\Omega \rho\boldsymbol{\delta u}\cdot\ddot{\boldsymbol{u}}\mathrm{d}\Omega = -\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{\delta u}) : \boldsymbol{\sigma}\mathrm{d}\Omega + \int_\Omega \boldsymbol{\delta u}\cdot\boldsymbol{f}\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{\delta u}\cdot\boldsymbol{t}\mathrm{d}\Gamma, \quad \forall\boldsymbol{\delta u}\in\boldsymbol{V} \tag{15}$$

where $\boldsymbol{\varepsilon}(\cdot)$ is the spatial symmetric gradient operator, $\boldsymbol{\varepsilon}(\boldsymbol{\delta u})$ is nothing but the spatial strain rate tensor, which is obtained from the standard FE kinematic interpolation (see Ref. [14] for details), $\boldsymbol{t}$ are the surface tractions which are applied on the boundary of Ω ($\partial\Omega$) in the actual configuration, and Γ_N is the Neumann part of $\partial\Omega$.

It is widely known that by using standard FE functions and equal interpolation for the kinematic variables (displacements, velocities, and accelerations) and the pressure field to discretize Eq. (15) in space, a lack of stability may appear if the material becomes almost incompressible. This is the case for metals in the plastic range, where it is standard to assume that the plastic flow is isochoric (the material flows preserving its volume). The mentioned lack of stability (a pure numerical problem) is demonstrated by the well documented pressure check-board mode [13,14] and locking effect. It is also well-established that the both anomalies may be avoided by several numerical techniques. Among them are: uniform reduced integration [14,15], selective integration [14], mixed interpolation [13,14] (div-stable elements), and numerical or physical stabilizations [3–12]. A mixed interpolation has been used in this work: the kinematic variables are interpolated using the standard Q1 FE functions (trilinear 8-nodes hexahedral elements), and the pressure is interpolated using the constant piecewise-discontinuous (P0) FE function (constant pressure at elemental level). In addition, a filter of pressures (smoothing) is performed each time step to enforce a convergent pressure field (free of check-board modes). This element does not require artificial viscosity terms due to the fact that it is free of hourglass modes [13] (the deviatoric stress terms are fully integrated).

3. Fracture scheme

Most of the fracture schemes may be classified into two groups (or a combination of them): smeared crack models and discrete crack approaches [18,19]. The former, also called continuum approaches, are based on the classical continuum (strain localization, smeared cracking) or the enriched continuum [16,20,21] (gradient enrichment, Cosserat continuum, non-local models). In such models, the material softens (using some constitutive law) forming a band where the crack (or micro-cracks) grows. In general, the topology of the mesh does not change during the simulation.

In the discrete crack approaches, the mesh topology changes to follow the cracks [18,22]. These schemes explicitly represent the crack as a separation of nodes. When some failure criterion is fulfilled, the node is redefined as two nodes and the elements are allowed to separate. While this produces a realistic representation of the opening crack, a coarse discretization in the finite element model results in misrepresentation of the propagating crack tip. One way to avoid this is by using remeshing procedures in the failure zone [22].

In the present work a discrete crack approach has been adopted. The main reason for this choice is that in coupled blast simulations big changes of topology are expected. A smeared approach would not be capable of realistically simulating the hundreds of fragments flying away from the main solid body, and the possibility of impacting some other surfaces. Remeshing procedures have not been implemented yet in the code, and it is not mandatory as long as fine enough meshes are utilized. In addition, expensive remeshing procedures could render large scale cases impossible to simulate.

The discrete fragmentation scheme is implemented as follows: first, the adopted failure criterion is computed at the interior of the elements. The most failed element e (element that violates the failure criterion with the highest value) is then chosen. Among the faces that are attached to e, the one with maximum tensile stress will be detached. This is, the one that fulfills the following expression:

$$(\boldsymbol{n}_i\cdot\boldsymbol{\sigma}^e)\cdot\boldsymbol{n}_i \geqslant (\boldsymbol{n}_j\cdot\boldsymbol{\sigma}^e)\cdot\boldsymbol{n}_j, \quad \forall i,j\in\mathscr{F}^e \quad \text{and} \quad i\neq j \tag{16}$$

where $\boldsymbol{\sigma}^e$ is the stress tensor in the element e, $\boldsymbol{n}_i$ the unit exterior normal on the face i (exterior to the element e) and $\mathscr{F}^e$ is the set of faces attached to e.

To detach the face with the highest tensile stress $\mathscr{F}_i^e$, each point $ip\in\mathscr{F}_i^e$ (4 points for hexahedral elements) is duplicated in the elements containing the topological perpendicular edge to the face. In other words, the nodal point ip belonging to the

face $\mathscr{F}_i^e$ is duplicated in all the elements containing the edge (ip, jp), where jp is the point that fulfills the following requirements: (1) jp belongs to the set of nodes of element e; (2) jp is connected to point ip; (3) jp is not contained in the set of nodes of the face $\mathscr{F}_i^e$.

The algorithm sketched above produces the main failure scheme for weapon fragmentation applications: longitudinal cracks consisting of very long stripes of metal which, although theoretically correct (tangential stress is higher than the longitudinal one), does not yield the small fragment pattern observed in experimental tests [25,26]. For that reason, a secondary fragmentation algorithm based on a lot of experimental data collected for more than 50 years, has been introduced to the CSD code. Such algorithm is based on the Mott's theory of break-up of cylindrical "ring-bombs". In [27] a detailed description of the model can be consulted. Below, just a brief summary is presented for completeness.

According to Mott [28], the average circumferential length of the resulting fragment is given by:

$$x_0 = \left(\frac{2\sigma_F}{\rho\gamma'}\right)\frac{r}{V} \tag{17}$$

where ρ and σ_F are the density and strength of the bomb material, respectively, r the radius of the ring at the moment of fracture, V its radial velocity, and γ' denotes a semi-empirical statistical constant determining the dynamic fracture properties of the material. The value of this last constant has been calibrated [27] using theoretical assumptions and experimental data to a value of $\gamma' = 56$. This has been the value used in this work for all the numerical simulations. Two others important relationship has been used for relatively large fragments. These are: the representative values of average aspect ratios of fragment lengths to circumferential breadth l_0/x_0, and the aspect ratios of circumferential breadth to fragment thickness x_0/t_0. The reported values in the literature [27,28] for such relationships are between $l_0/x_0 \approx x_0/t_0 \approx 2.5$ and $l_0/x_0 \approx x_0/t_0 \approx 5.0$. In this job a value of 4.0 has been used. However, some additional random effects are introduced to the fragment sizes. These are: the unstructured character of the used meshes as the reader can observe in Fig. 1, and a uniformly distributed random variable which will be clarified later.

Therefore, for weapon fragmentation cases the final fracture algorithm is as follows:

(1) Compute the element with the highest value of the failure criteria $e_{\max}$ (i.e., the element with maximum effective plastic strain).
(2) Obtain the circumferential length of the possible resulting fragment using expression (17).
(3) Obtain the length of the possible resulting fragment using $l_0/x_0 = 4$.
(4) Compute the average failure criteria f_{top} in a ring of elements of size l_0 above $e_{\max}$.
(5) Compute the average failure criteria f_{bot} in a ring of elements of size l_0 below $e_{\max}$.
(6) If f_{top} greater than the value the material can stands, the top ring will be fragmented using the relations $l_0/x_0 = x_0/t_0 = 4 * r$, where r is a uniformly distributed random number between 0.625 and 1.25, that introduces a random effect to take into account, in some way, the spatial variability of the material strength.
(7) If f_{bot} greater than the value the material can stands, the bottom ring will be fragmented using the relations $l_0/x_0 = x_0/t_0 = 4 * r$.

The above fragmentation algorithm produces results that compares remarkably with the experimental data.

3.1. Rigidization scheme

After the failure algorithm is performed and some topology changes are detected, the possible detachment of fragments of elements from the main body is checked. The procedure is as follows: one element e is marked with a fragment identification number ID. After that, the elements surrounding e are marked with the same ID, the elements surrounding the elements that surrounds e are also marked, and the procedure is successively repeated until no more elements are left. So, the fragment number ID has been identified. Other unmarked elements are then chosen to attempt to identify other fragments by repeating the whole procedure. The fragment identification algorithm is finished when all the mesh elements have been marked as belonging to some fragment.

After that, all the fragments with a number of elements that is less than a user defined value are rigidized. This is, the internal forces of the elements belonging to such fragments are set to zero, and their external forces are averaged and applied as a constant force to all the fragment elements. The fragment will then behave as a rigid body.

Finally, it is important to remark that the rigidization algorithm is activated only when the detailed stress distribution into the small fragments is not required, but their velocities and masses. This is the case for most of the weapon fragmentation problems.

4. Fluid/Solid coupling

The fluid–structure interaction is simulated by coupling a CSD solver and a CFD solver in a staggered manner (see Box 1 for details), and by using an embedded approach for the fluid–solid interaction (FSI) treatment. In our scheme, the

computational mesh for the solid body (for the CSD solver) is totally embedded into the fluid mesh (mesh for the CFD code). This is, the fluid mesh covers the entire solid body (or bodies), and it is not conforming (not fitted) to the solid boundary faces (fluid mesh edges are intersected by solid boundary faces).

Box. 1. Description of the CFD/CSD coupling algorithm.

```
Read meshes; read initial and boundary conditions for both solvers;
CSD code sends the external solid faces (𝒮) and initial velocities (v_𝒮) on 𝒮 to CFD solver;
t_CFD ← 0; t_CSD ← 0;
DO WHILE (t_CSD < t_final and t_CFD < t_final)
   CFD code computes the fluid elements intersected by 𝒮;
   CFD code solves the fluid from t to t + δt_CFD using an embedded technique;
   CFD code sends the pressures P on 𝒮 to CSD code;
   t_CFD ← t + δt_CFD;
   DO WHILE (t_CSD ⩽ t_CFD)
      CSD code solves the structure (updates v_𝒮) loading 𝒮 with P;
      t_CSD ← t + δt_CSD;
      Update the external faces 𝒮;
   END DO
   CSD code sends the new external solid faces (𝒮) and velocities
   (v_𝒮) to CFD solver;
END DO
```

The staggered algorithm and FSI treatment may be read as follows: The simulation starts and all the data for both solvers are read. The CSD solver computes the external faces mesh of the solid body, and sends its coordinates, connectivities, and velocities to the CFD solver. Then, the CFD solver computes the fluid mesh edges that are intersected by the solid faces, and imposes the right flow boundary and initial (previous time step) conditions at the intersecting points. This is, the velocities are set to the solid faces velocities (or only to their normal components for Euler type flows), and, in general, all the variables are interpolated to the mentioned intersecting points, and to the fluid nodes which were inside the solid body at the previous time step, if it is required. This last procedure may be performed by a simple first order approach, or by a second order scheme which uses point mirroring and unknowns interpolation with gradient reconstruction. A detailed description of the FSI (fluid–solid interaction) treatment may be consulted in Refs. [25,26].

After the flow field is solved taking into account the blocking effect of the solid faces (boundary and initial conditions described above), the flow pressures are interpolated (applied) over such a solid faces. Then, the CSD solver updates the structure kinematic variables, internal stresses, strains and geometry with the new loads (new pressures over the solid faces). Finally, the solid faces are updated and sent back to the flow solver to repeat the whole procedure for a new time step. A detailed algorithm is shown in Box 1.

The stability of the CSD and CFD solvers is guaranteed by choosing different time step sizes for each code. The time step size for the explicit CSD code (δt_{CSD}) is computed in a standard manner, based on the element types and sizes [2]. A standard second-order explicit Newmark scheme (the widely called central difference scheme) is used to integrate the dynamic equilibrium equation in time [14]. The explicit CFD code also computes its own time step size (δt_{CFD}) using a standard Courant criterion [23].

Finally, it is important to remark that both the CFD and the CSD solvers are parallelized for shared memory architectures using the Open MP libraries. The CSD solver uses a coloring algorithm [23] to compute all the operations over the elements in a parallel manner, and a renumbering technique to avoid cache misses. Most of the cracking algorithm is also parallelized: the failure criterion computation over the elements, its smoothing from the element to the nodes, and the computation of i_{max} (see Section 3) are operations that might be perfectly parallelized. On the other hand, the node duplication and element reconnection are performed in a scalar way. This drawback does not have much impact in the total cost of the simulation because such operations involve a very small group of elements. Therefore, its CPU time can be almost neglected in comparison to other elemental tasks: internal force computations, constitutive law evaluation, etc.

5. Numerical examples

Several numerical examples for elastic and plastic materials were performed to validate the implemented solid element [24]. Among them were: the bar impacting on a rigid wall (typical benchmark) and the necking of a circular bar [17]. All the numerical results showed great agreement with the ones reported by other authors.

Due to the scope of this work, only real coupled test cases are of interest. Hence, the fragmentation of a generic weapon inside a chamber with very strong walls is presented to demonstrated the described methodology.

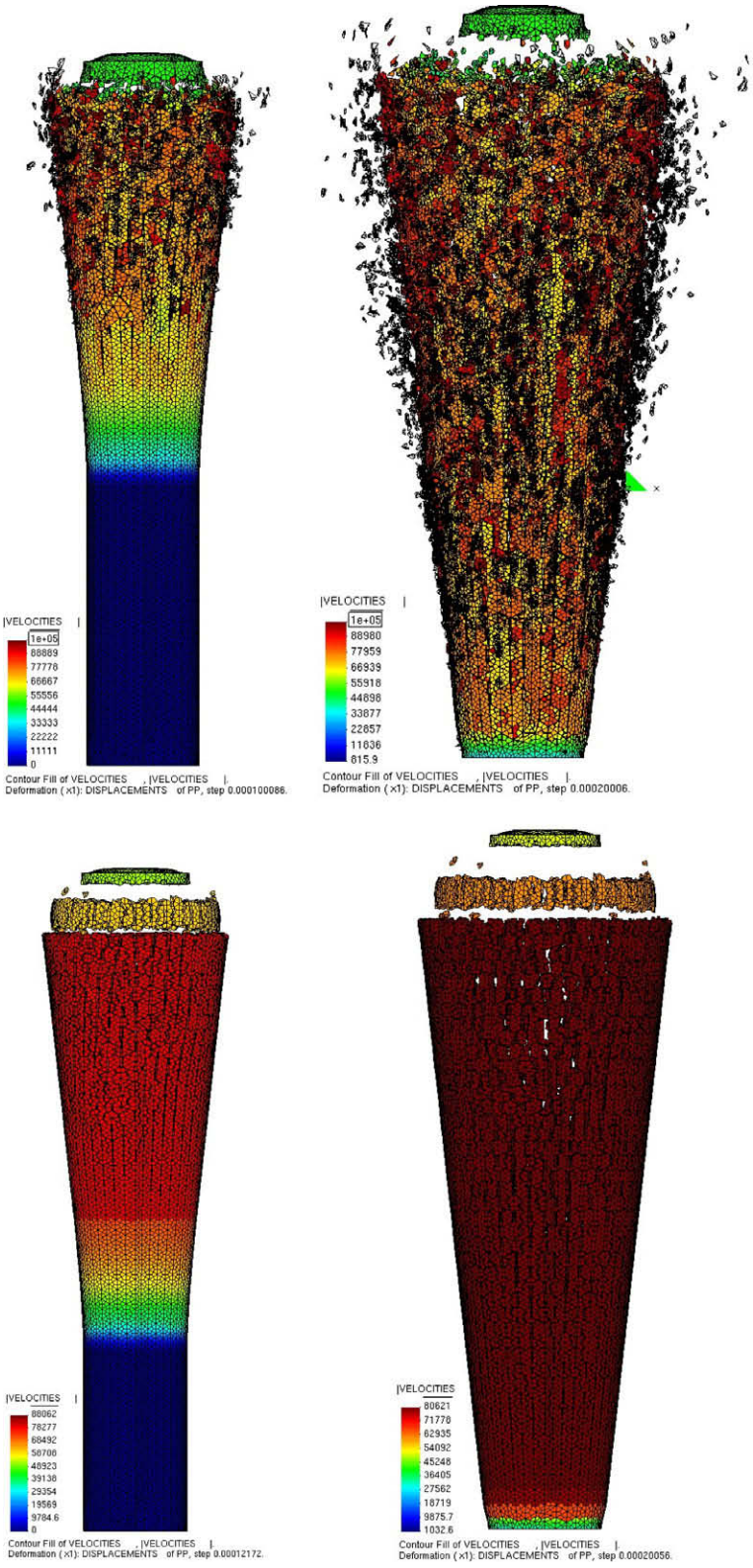

Fig. 2. Comparison of agglomeration schemes. From left to right and top to bottom: long strips and one-element fragments at 100.0 μs using the node disconnection scheme as it is; same result at 200.0 μs using the node disconnection scheme; fragmentation using the Mott's approach at 100.0 μs; fragments at 200.0 μs using the Mott's approach.

The geometry and some details of the finite element mesh for the weapon can be observed in Fig. 1. The walls of the chamber were assumed rigid: all its kinematic and dynamic variables were prescribed to zero. The adopted weapon material properties are presented as follows: Density $\rho = 7.76\ \mathrm{g/cm^3}$, Young modulus $E = 6.415 \times 10^{12}\ \mathrm{dy/cm^2}$, Poisson ratio $v = 0.3$, yield stress $\sigma_y = 1.283 \times 10^{10}\ \mathrm{dy/cm^2}$, isotropic hardening modulus $K = 8.657 \times 10^{10}\ \mathrm{dy/cm^2}$ and maximum effective plastic strain for failure $\overline{\varepsilon}^p_{\max} = 0.3$.

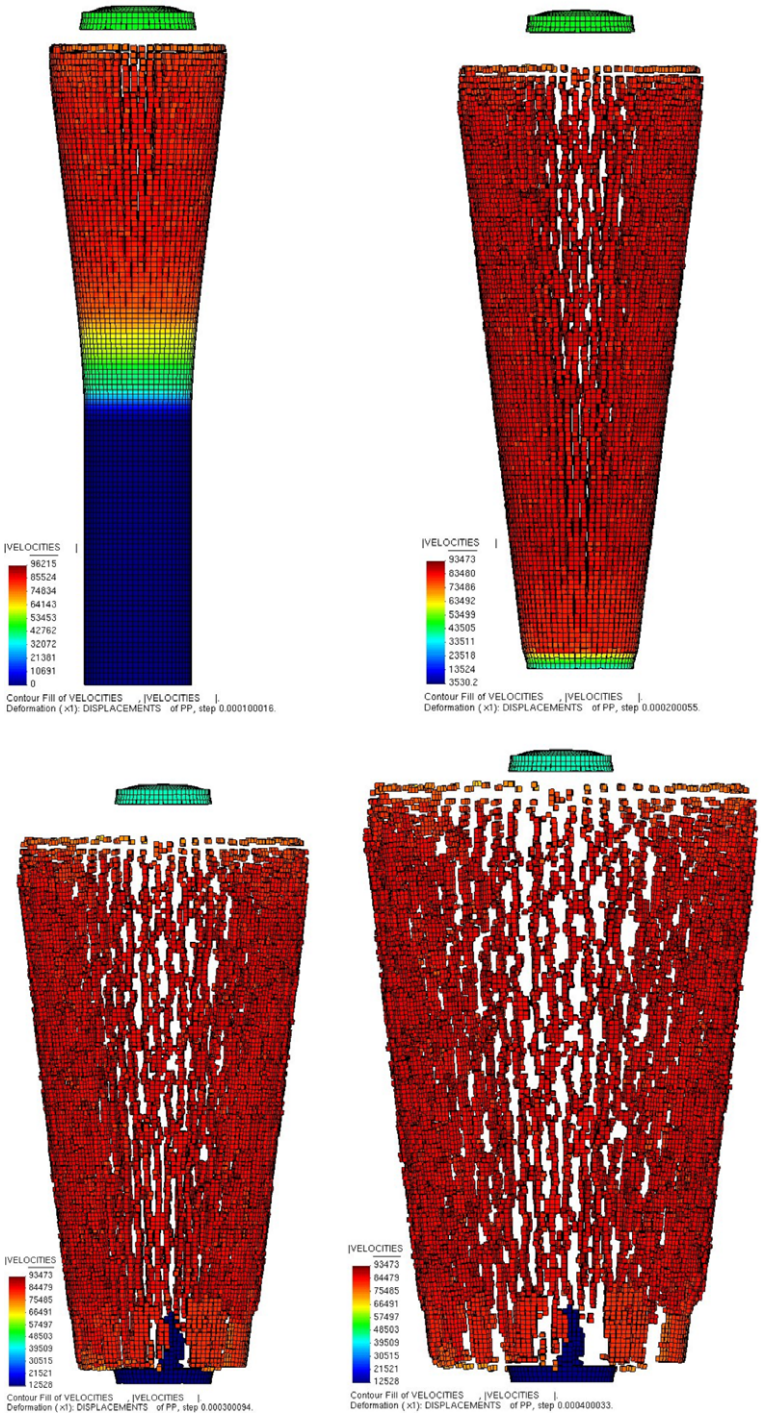

Fig. 3. Failure evolution and fragment formation using the node disconnection scheme and a structured mesh. From left to right and top to bottom: fracture at 100, 200, 300, and 400 μs.

To compare the straightforward failure scheme (node disconnection as it is) and the one used the empirical agglomeration algorithm given by Mott's expression (17), a cheap stand-alone simulation were performed. Basically, the flow field around and inside the weapon was not solved, and just a pressure front traveling inside the bomb from the top to the bottom were prescribed. The pressure peak was set to 10^{11} dy/cm^2 and the front velocity to 7.2×10^5 cm/s. The decay of the pressure front was simulated with an exponential function of the form:

$$p = p_{\text{peak}} e^{-0.1d} \tag{18}$$

where d is the distance from the faces above the pressure front (the front is traveling from the top to the bottom of the weapon), to the position of the peak pressure. The pressure of the faces below the peak pressure were set to zero.

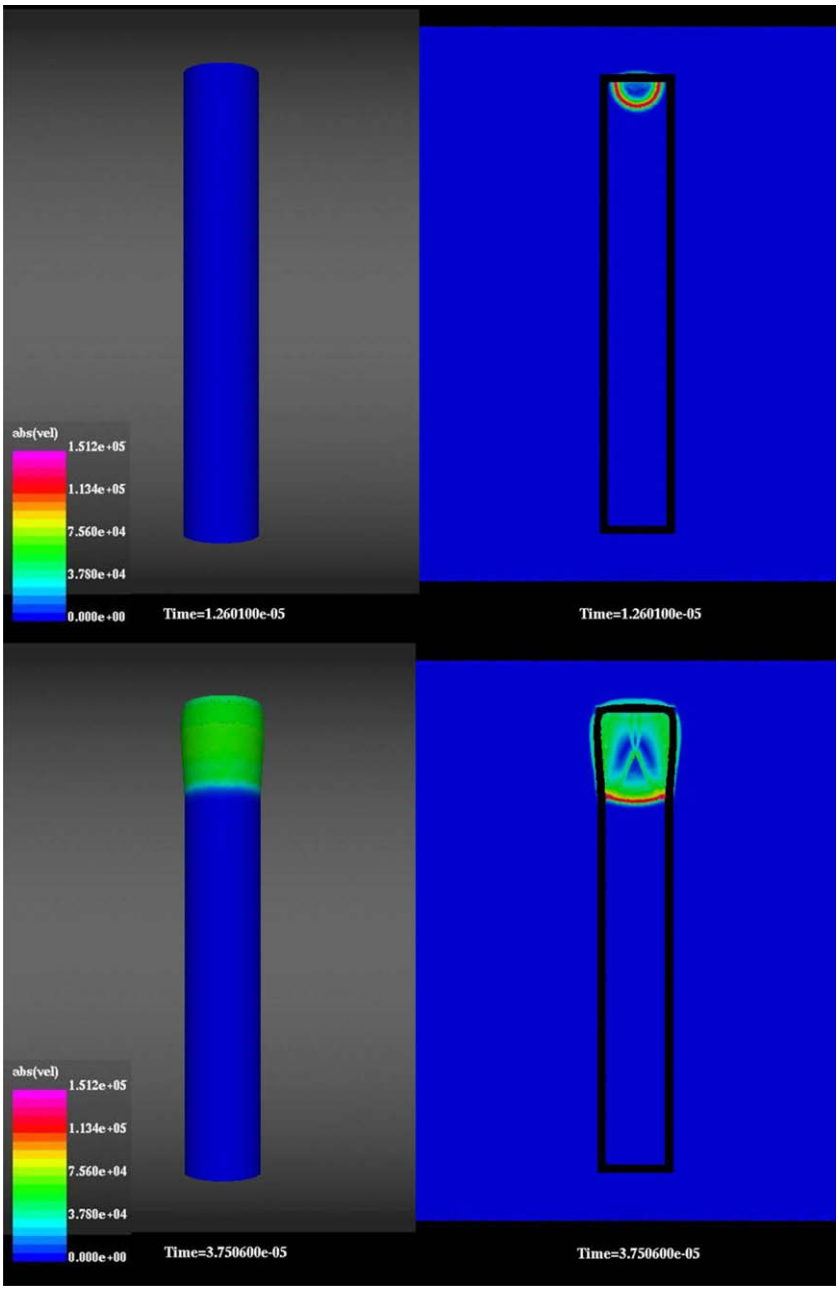

Fig. 4. From left to right and top to bottom: CSD (solid metal) velocities at 12.6 μs, vertical cut of fluid velocities at 12.6 μs, CSD velocities at 37.5 μs, vertical cut of fluid velocities at 37.5 μs (CFD velocity values are hidden for clearance reasons).

In Fig. 2 (top) the fragmentation pattern using the un-modified failure scheme is shown. One can observe longitudinal long strips of material plus some small one-element fragments (elements that has been disconnected) which, as it was previously mentioned, are theoretically correct. However, the final sizes and mass distribution of the fragments does not agree with the experience for this type of weapons (see Fig. 2 top right).

Using the agglomeration technique based on the Mott's expression (17), the weapon fragmentation seems to show a better agreement with the observed experimental data. In Fig. 2 (bottom) the evolution of the metal failure using this scheme is presented.

For additional tests, a structured mesh was generated to check the un-modified (simple node disconnection) failure scheme for this type of meshes. In Fig. 3 the fracture evolution for the structure grid is shown. Even though the fragment formation improved with respect to the un-structured grid, the final fragments does not look real. They are too well shaped, due to the fact that the randomness introduced by the unstructured grids is lost.

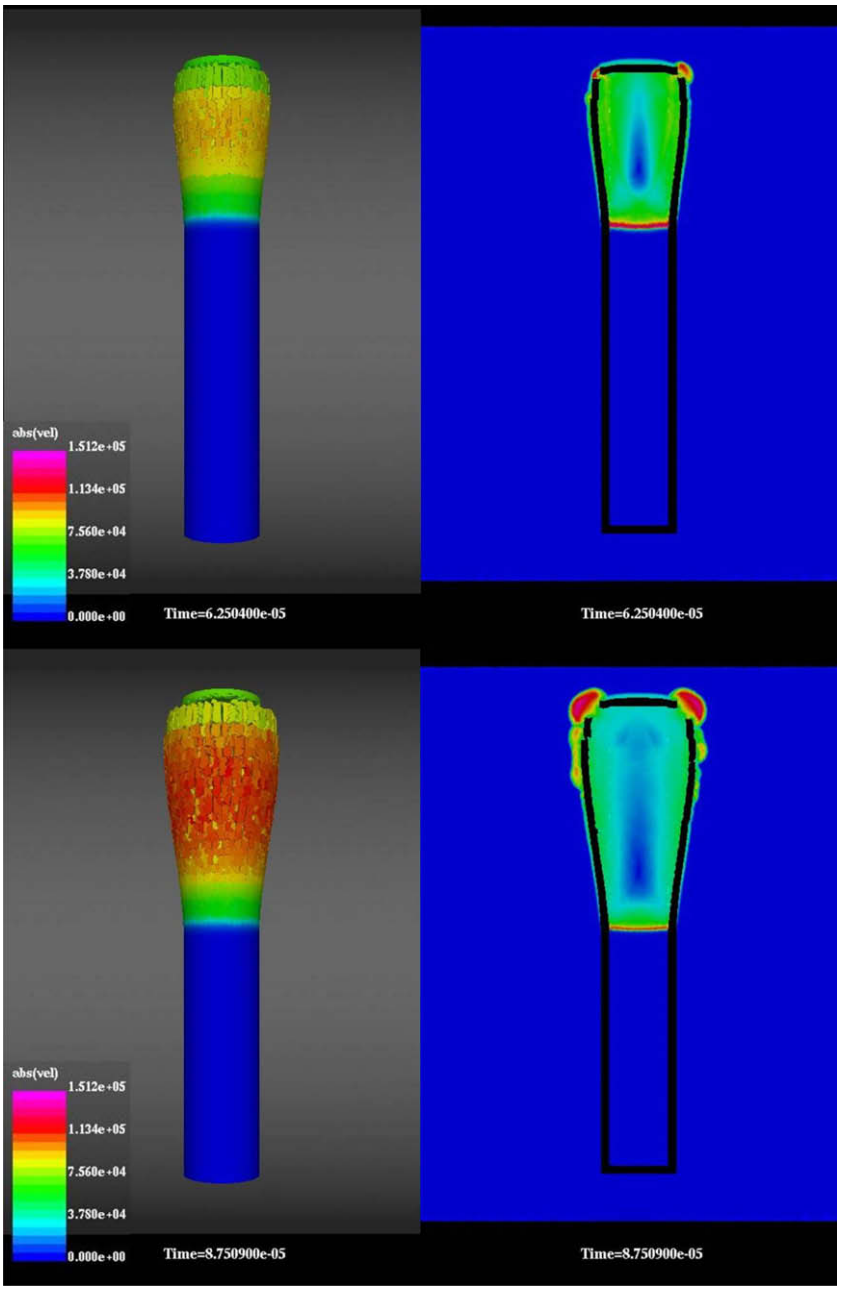

Fig. 5. From left to right and top to bottom: CSD velocities at 62.5 μs, vertical cut of fluid velocities at 62.5 μs, CSD velocities at 87.5 μs, vertical cut of fluid velocities at 87.5 μs (CFD velocity values are hidden for clearance reasons).

Based on the previous results, a coupled simulation was performed using the Mott's scheme for the fragmentation process. The simulation was performed using the algorithm presented in Box 1. An explosive was ignited (point detonated) inside the weapon (at the top), generating a pressure front that travels along the case, and producing the fragmentation evolution showed in Figs. 4–8. The results can be observed to be in excellent qualitative agreement with experimental data. The fragment velocities, sizes and shapes are very similar to those reported by real tests. It can also be noticed that the top part of the weapon is detached from the body at an early fragmentation stage (something commonly observed in experiments). In the same Figs. 4–8 a vertical cut of the fluid velocities evolution may also be observed. The effect of the embedded fragments on the fluid velocities seems to be well captured, which validates the used coupled approach. Fig. 9 presents a cut of the fluid pressures and of the pressures over the solid faces at different steps. This Figure demonstrates the extrapolation of such a variable from the flow field to the embedded solid structure.

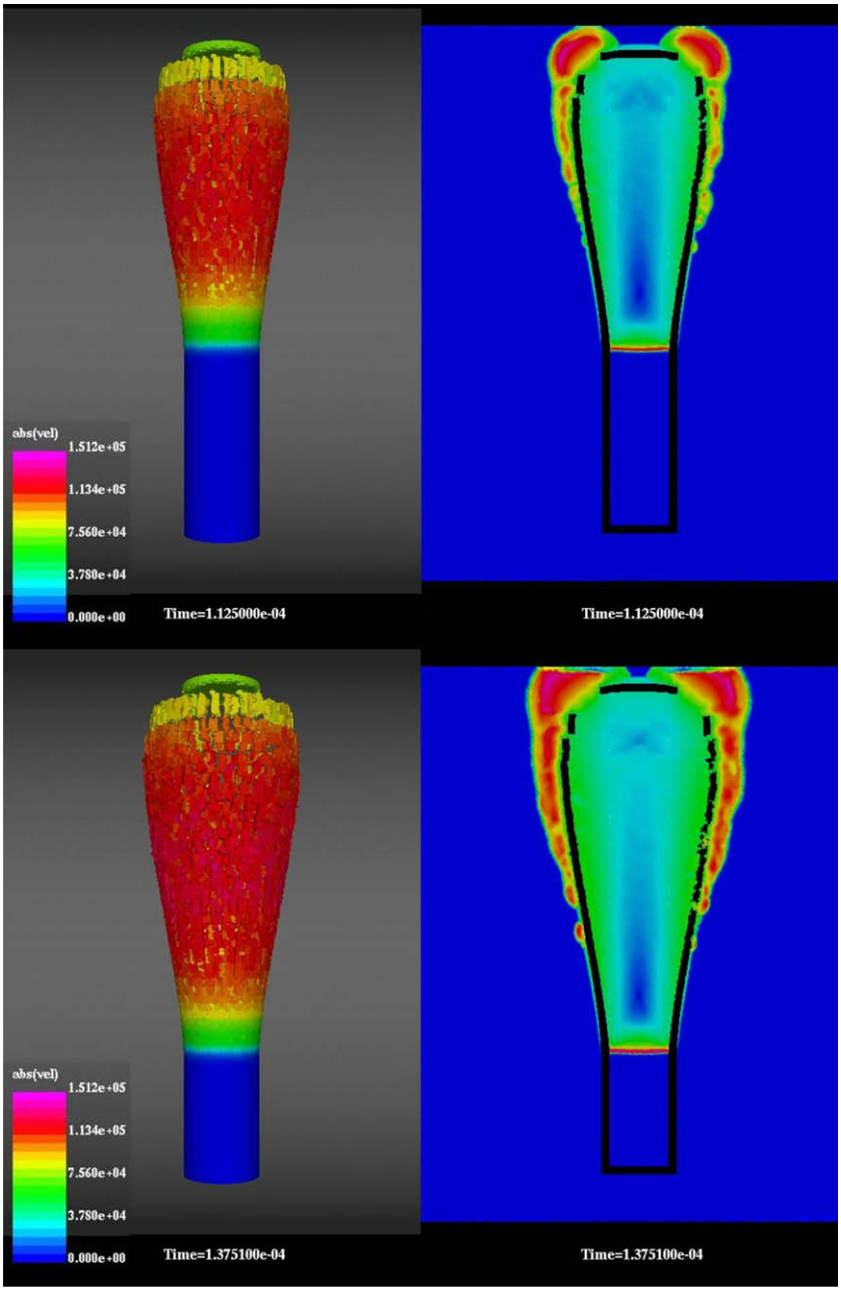

Fig. 6. From left to right and top to bottom: CSD velocities at 112.5 μs, vertical cut of fluid velocities at 112.5 μs, CSD velocities at 137.5 μs, vertical cut of fluid velocities at 137.5 μs (CFD velocity values are hidden for clearance reasons).

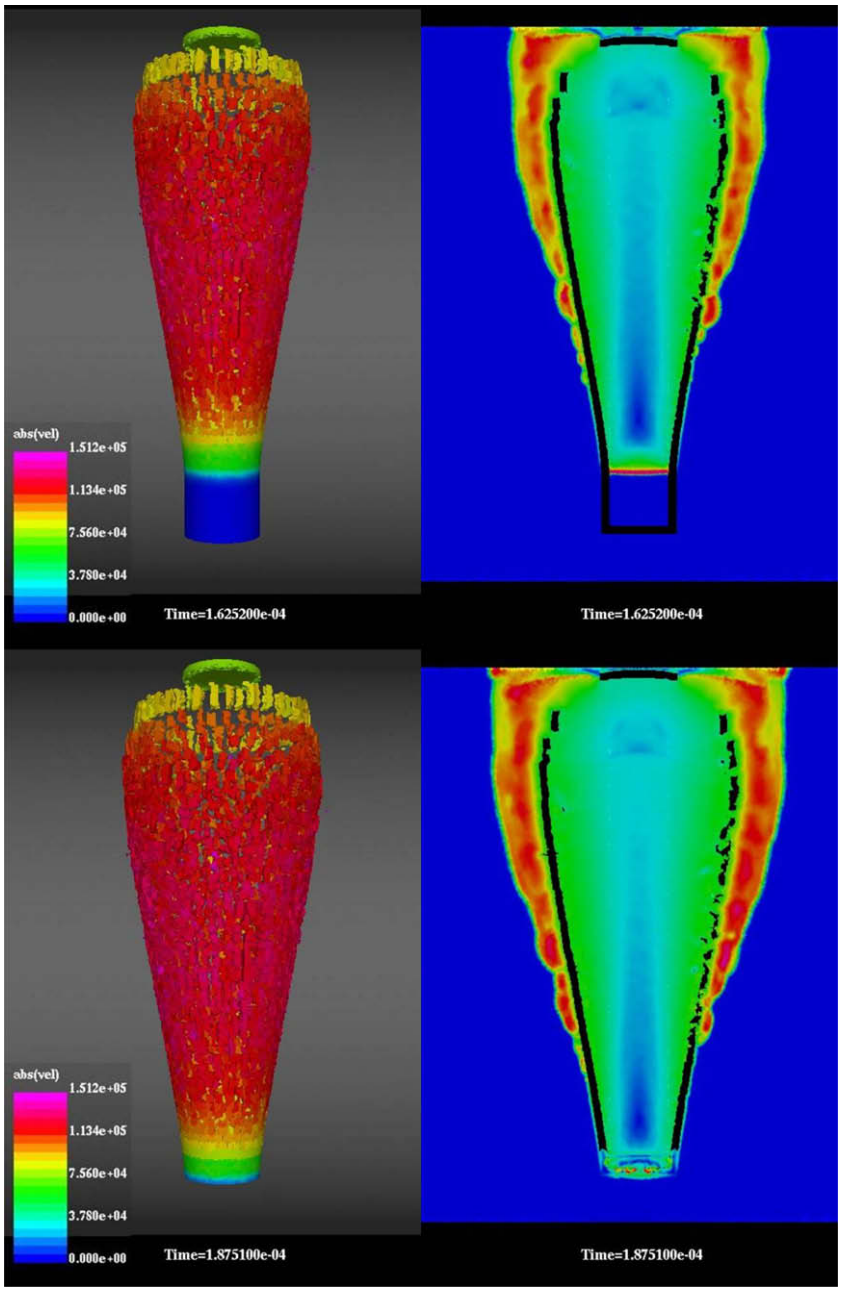

Fig. 7. From left to right and top to bottom: CSD velocities at 162.5 μs, vertical cut of fluid velocities at 162.5 μs, CSD velocities at 187.5 μs, vertical cut of fluid velocities at 187.5μs (CFD velocity values are hidden for clearance reasons).

Finally, in Fig. 10 the fragment mass distribution is shown. The agreement is good with experimental data for this type of weapons. Fig. 11 shows the evolution of the fragment energy. Again, it fits very well the experience for this type of weapons: most of the explosive energy is transformed in kinetic fragments energy.

As a final remark, it can be observed that all the fragments (including the top and bottom parts of the weapon) are effectively rigidized and they fly alone free of instabilities.

6. Conclusions and final commentaries

An efficient CSD/CFD scheme for coupled blast simulations was presented. It is based on a fully integrated solid hexahedral element with mixed interpolation (Q1/P0: trilinear in velocities and constant piecewise-discontinuous

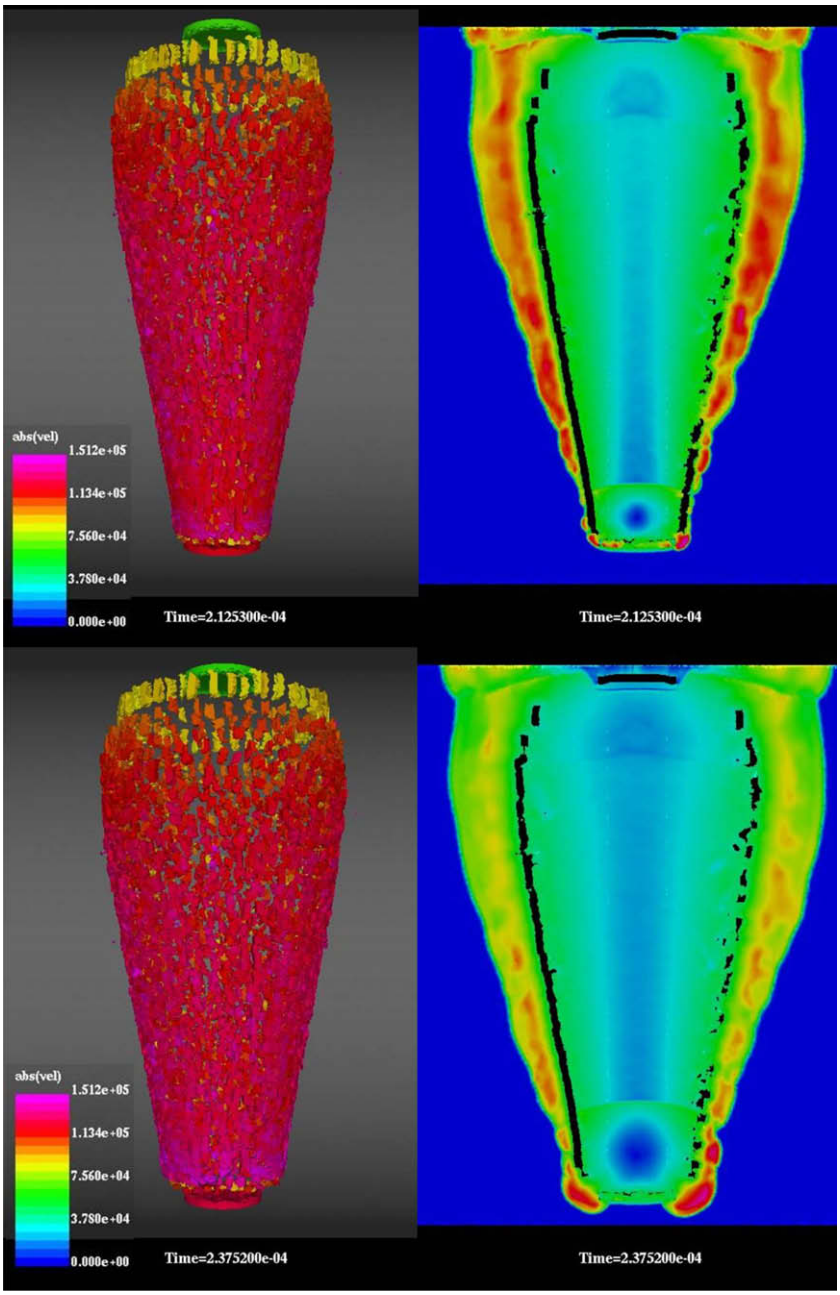

Fig. 8. From left to right and top to bottom: CSD velocities at 212.5 μs, vertical cut of fluid velocities at 212.5 μs, CSD velocities at 237.5 μs, vertical cut of fluid velocities at 237.5 μs (CFD velocity values are hidden for clearance reasons).

pressures), a well-founded large deformation plasticity constitutive model, and in some fast procedures to treat the material's fracture. Moreover, all the algorithms are fully parallelized: the CPU speed up obtained in the numerical examples was quasi-optimum. One coupled numerical example was presented to demonstrate the overall scheme, which shows very realistic results.

Furthermore, it was found during the simulations that the use of hypoelastic schemes to treat the plasticity in metals, combined with widely used objective stress rates (Jaumann–Zaremba, Green–McInnis–Naghdi, etc.) and unstructured meshes, violates in a unacceptable manner the incompressible character of the plastic flow. This produces negative volume elements during the simulation, edge crossing and, therefore, very poor results. Hence, it is highly recommended the utilization of hyperelastic models and convective formulation to treat the plastic material behavior in unstructured FE meshes.

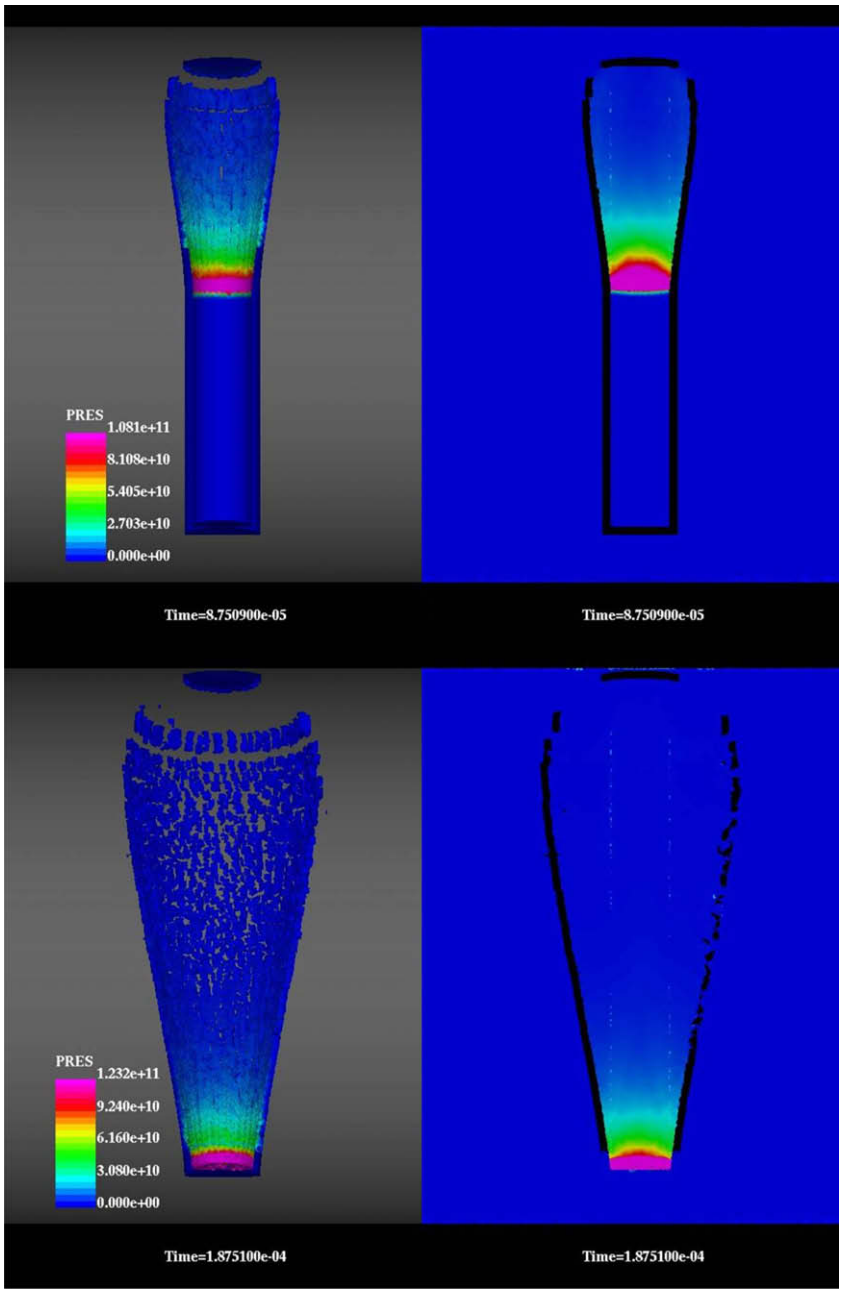

Fig. 9. From left to right and top to bottom: Interpolated pressure over CSD faces at 87.5 μs (vertical clipping to show the interior solid faces of the weapon), vertical cut of flow pressures at 87.5 μs, interpolated pressure over CSD faces at 187.5 μs, vertical cut of flow pressures at 187.5 μs.

Finally, it is important to remark that the described CSD scheme is being used to solve real coupled blast cases. The obtained results for the large scale applications show to be realistic, and the used CPU time is also very encouraging. To give an idea of the CSD code speed, a simulation using a mesh of 1×10^5 3D Q1/P0 CSD elements, approximately spent 0.5 s of CPU on 16 ALTIX SGI processors.

Acknowledgement

The authors are thankful for the financial support of the US Defense Thread Reduction Agency (DTRA).

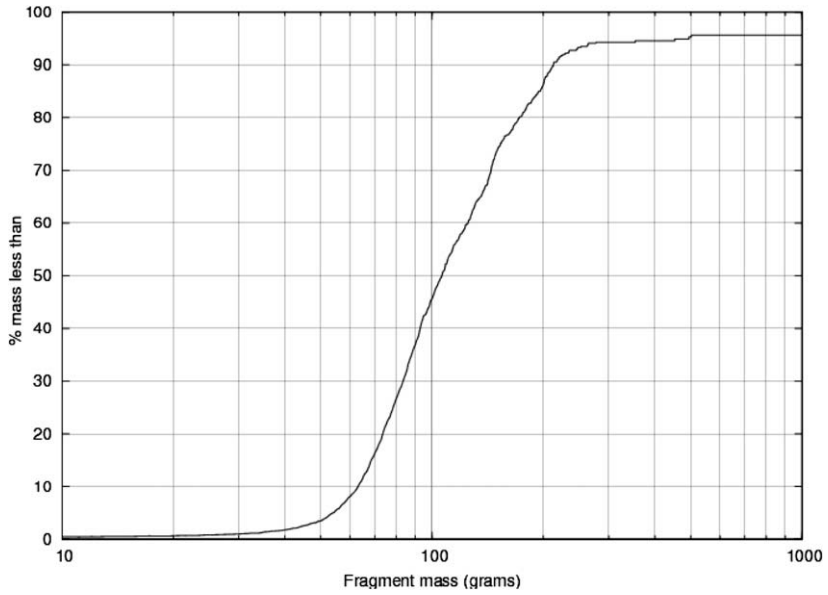

Fig. 10. Fragment mass distribution.

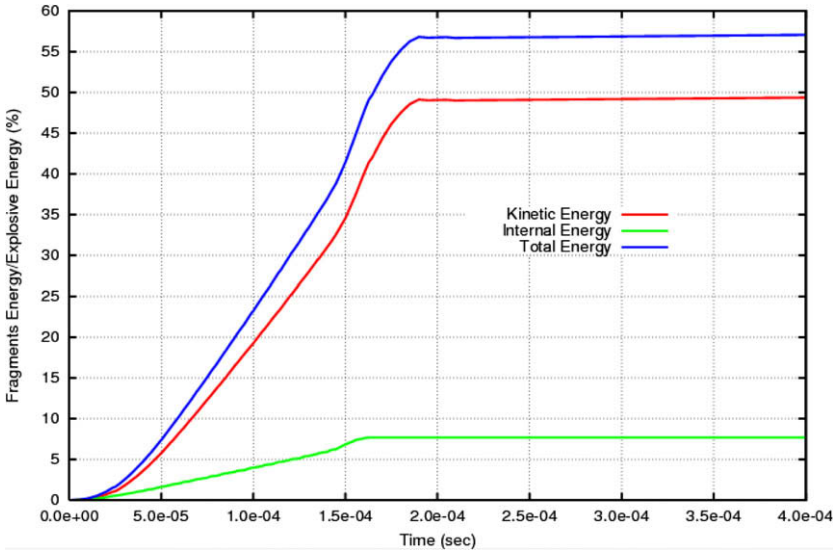

Fig. 11. Fragment energy as a fraction of the total flow (CFD) energy.

References

[1] Whirley RG, Hallquist JO. DYNA3D: a nonlinear explicit three-dimensional finite element code for solid and structural mechanics, user manual. Report UCRL-MA-107254, Lawrence Livermore National Laboratory, Livermore, CA; 1991.
[2] Hallquist JO. LS-DYNA theoretical manual. Livermore, CA: Livermore Software Technology Corporation; 1998.
[3] Codina R. On stabilized finite element methods for linear systems of convection–diffusion–reaction equations. Comput Methods Appl Mech Engng 2000;188:61–82.
[4] Tessler A, Hughes TJR. A three-node Mindlin plate element with improved transverse shear. Comput Methods Appl Mech Engng 1985;50:71–101.
[5] Garnet H, Pascal C, Pifko AB. Aspects of a simple triangular plate bending finite element. Comput Struct 1980;12:783–9.
[6] MacNeal RH. A simple quadrilateral shell element. Comput Struct 1978;8:175–83.
[7] Dvorkin EN, Bathe KJ. A continuum mechanics based four-node shell element for general nonlinear analysis. Engng Computat 1984;1:77–8.
[8] Lee P, Bathe KJ. Development of MITC isotropic triangular shell finite elements. Comput Struct 2004;82:945–62.
[9] Oñate E, Zárate F. Rotation-free plate and shell triangles. Int J Numer Methods Engng 2000;47:557–603.
[10] Belytschko T, Leviathan I. Projection schemes for one-point quadrature shell elements. Comput Methods Appl Mech Engng 1994;115:277–86.
[11] Belytschko T, Leviathan I. Physical stabilization of the 4-node shell element with one point quadrature. Comput Methods Appl Mech Engng 1994;113:321–50.

[12] Hughes TJR, Liu WK. Nonlinear finite element analysis of shells: part I. Three dimensional shells. Comput Methods Appl Mech Engng 1981;26:331–62.
[13] Belytschko T, Liu WK, Moran B. Nonlinear finite element for continua and structures. Wiley; 2000.
[14] Hughes TJR. The finite element method: linear static and dynamic finite element analysis. Prentice-Hall; 2000.
[15] Zienkiewicz OC, Taylor RL. The finite element method, vol. I. McGraw Hill; 1989 [vol. II, 1991].
[16] Zienkiewicz OC, Taylor RL, Papadopoulos RL, Oñate E. Plate bending elements with discrete constraints. Comput Struct 1990;35:505–22.
[17] Simo JC, Hughes TJR. Computational inelasticity. New York: Springer-Verlag Inc.; 1998.
[18] A. Tzamtzis. Finite element modeling of cracks and joints in discontinuous structural systems. In:16th ASCE engineering mechanics conference, American society of civil engineers, University of Seattle, Washington, July 16–18; 2003.
[19] Oliver J, Huespe AE. Continuum approach to material failure in strong discontinuity settings. Comput Methods Appl Mech Engng 2004;193:3195–220.
[20] Moës N, Dolbow J, Belytshcko T. A finite element method for crack growth without remeshing. Int J Numer Methods Engng 1999;46:131–50.
[21] Mosler J, Meschke G. Embedded crack vs. smeared crack models: a comparison of elementwise discontinuous crack path approaches with emphasis on mesh bias. Comput Methods Appl Mech Engng 2004;193:3351–75.
[22] Schöllmann M, Richard HA, Kullmer G, Fulland M. Development of a new software for adaptive crack growth simulations in 3D structures. Engng Fract Mech 2003;70:249–68.
[23] Löhner R. Applied CFD techniques. John Wiley & Sons; 2001.
[24] Soto O. CSD code development. SAIC internal report. May 19; 2004.
[25] Löhner R, Baum JD, Mestreau E, Sharov D, Charman C, Pelessone D. Adaptive embedded unstructured grid methods. Int J Numer Methods Engng 2004;60:641–60.
[26] Löhner R, Baum JD, Mestreau EL. Advances in adaptive embedded unstructured grid methods. AIAA-04-0083; 2004.
[27] Gold VM, Baker EL. A model for fracture of explosively driven metal shells. Engng Fract Mech 2008;75:275–89.
[28] Mott NF, Linfoot EH. A theory of fragmentation. Ministry of supply, AC 3348; January 1943.
[29] Johnson GR, Cook WH. A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. In: Proceedings of the 7th international symposium on ballistics. The Netherlands: The Hague; 1983. p. 541–7.
[30] Johnson GR, Cook WH. Fracture characteristics of three metals subjected to variuous strains, strain rates, temperatures and pressures. Engng Fract Mech 1985;21:31–48.