COMPDYN 2011
III ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis, V. Plevris (eds.)
Corfu, Greece, 26–28 May 2011

STRONG AND WEAK FORMS OF A FULLY NON-CONFORMING FSI ALGORITHM IN FAST TRANSIENT DYNAMICS FOR BLAST LOADING OF STRUCTURES

F. Casadei[1], M. Larcher[2], and N. Leconte[3]

[1] Joint Research Center, European Laboratory for Structural Assessment
T.P. 480, 21027 Ispra, Italy
e-mail: folco.casadei@jrc.ec.europa.eu

[2] Universität der Bundeswehr München, Institut für Mechanik und Statik
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
martin.larcher@unibw.de

[3] Université de Valenciennes, LAMIH FRE CNRS-UVHC 3304
Le Mont-Houy, Bât. Jonas – 59313 Valenciennes Cedex 9, France
nicolas.leconte@univ-valenciennes.fr

Keywords: Fluid-Structure Interaction, Transient Dynamics, Blast Loading.

Abstract. *The simulation of blast loading effects on structures undergoing very large deformations, possibly up to complete failure and fragmentation, requires robust and specific Fluid-Structure Interaction (FSI) algorithms. The FLSR algorithm uses fully non-conforming fluid and structure meshes (the structure is "embedded" within the fluid), that are "strongly" coupled by suitable constraints on velocities near the F-S interface, enforced by Lagrange multipliers. This technique allows using a regular (or even a structured, typically Eulerian) fluid mesh, while the structure is Lagrangian and can undergo failure and fragmentation without the mesh entanglement difficulties that would arise with more traditional (conforming) FSI algorithms. However, the localization of coupled F-S nodes is more expensive than in the conforming case and must be continuously updated, by means of fast search algorithms, as the structure moves and deforms. The strong form of the algorithm (FLSR) is suitable for a Finite Element (FE) description of the fluid domain and has been used with success in large realistic simulations. Application to Node-Centered Finite Volumes (NCFV) requires an additional treatment of numerical fluxes, presented in the paper. Recently a "weak" variant of the algorithm (FLSW) has been developed for use with Cell-Centered Finite Volumes (CCFV) in the fluid domain. Suitable pressure (interaction) forces are used for the coupling, rather than strong conditions on velocities. Like for NCFV, appropriate treatment of numerical fluxes is required. The rationale and implications of this choice are discussed in the paper.*

1 INTRODUCTION

Fluid-structure interaction (FSI) phenomena play an important role in many areas, ranging from aeronautical and space applications, to civil and marine/offshore engineering and to the transport industry, to name just a few. The JRC has been involved for many years in the development of numerical methods for FSI modeling applied to safety studies—initially for the nuclear industry and more recently for conventional power plants (electrical machinery)—to civil engineering (vulnerability of buildings and other critical infrastructures to terrorist attacks) and to land mass transports (blast effects in railway stations, metro lines, rolling stock).

All these studies are characterized by the violent blast loading, resulting either from an accident or from an intentional attack, and by the very short time scale (fast transient dynamics). Strong pressure waves propagate in the fluid and load the surrounding structures, which typically undergo large deformations and in some cases reach complete failure and fragmentation. For this class of problems, an explicit time marching algorithm is usually adopted, where the fluid is modelled as compressible and inviscid (Euler equations). An Arbitrary Lagrangian Eulerian (ALE) formulation is adopted for the fluid sub-domain, while the structure is Lagrangian. These models are implemented in the EUROPLEXUS code [1], developed jointly by the JRC and by the French CEA.

Three different discretization approaches are available in the code for the fluid sub-domain, see Fig. 1: finite elements (FE), node-centred finite volumes (NCFV) and cell-centred finite volumes (CCFV). In the FE case, kinematic variables (such as the velocity $\mathbf{v}$) are discretized at the element nodes, while state variables (such as the fluid pressure p) are discretized at Gauss points, typically located at the element (shaded area in Fig. 1a) centroid. In the NCFV case, a virtual FV (dual) mesh centred on fluid nodes (shaded area in Fig. 1b) is automatically built up starting from the FE-like (primal) mesh provided in input, and all variables are discretized at the nodes. In the CCFV case, the FV mesh (shaded area in Fig. 1c) looks similar to the FE case, but all variables are discretized at the volume centres. Note that in this case the "nodes" carry no relevant information other than their position, used to compute the volume.

The coupling between the fluid (ALE) and the structure (Lagrangian) is realized by suitable FSI algorithms. Two broad classes of algorithms are considered here. The first class uses a "*strong*" approach, based on constraints imposed on the (velocity of) fluid and structure nodes at the F-S interface. The second class uses a "*weak*" approach, based on direct application of fluid pressure forces to the structure. This terminology (strong/weak) is tentatively adopted here in an attempt to characterize the different nature of the two approaches, but it should not be confused with other uses of the same terms in the literature, in particular with weak (i.e. integral) forms in continuum mechanics. Traditionally, strong FSI algorithms are mainly used in FE, while weak FSI algorithms are mainly used in FV.

Yet another classification of FSI algorithms concerns the degree of deformation/damage that the structure can undergo (and thus the type of application). One class of "*basic*" algorithms is suitable for large motion and large deformation of structures, but only provided these do not fail. Another class of algorithms can go up to complete *failure*, and *fragmentation*, of the loaded structures.

Finally, FSI algorithms can be classified in three types according to spatial discretization: (nodally) *conforming*, *non-conforming*, and *embedded* (or immersed). The first two types are mostly used in applications without structural failure (but there are exceptions, see Section 3.3), while embedded algorithms are the only ones capable of dealing with extreme loading cases where the structure fails and breaks up in pieces.

Examples of all such different algorithms are shortly presented, by highlighting their characteristics. The paper is organized as follows. Section 2 reviews some algorithms of the basic

type (no structure failure), either conforming or non-conforming, which use the strong approach in a FE or NCFV context. Section 3 presents algorithms of the weak type, in a CCFV context. Section 4 shows embedded algorithms with strong and weak formulations. Some numerical examples are presented in Section 5. Finally, Section 6 contains some conclusions and perspectives for future work.

2 BASIC FSI ALGORITHMS OF THE STRONG TYPE

The basic FSI algorithms—valid for largely deforming (but *not failing*) structures and using a strong approach—are briefly summarized, first in a FE and then in a NCFV context. Readers interested in full details should consult references [2-7]. A sample FSI problem is assumed as a reference, see Fig. 2a (continuous domain). A shell-like structure S is considered, in 2D for simplicity, but a junction (J) completely surrounded by fluid is included for generality. As indicated by colors, the nature of the fluid(s) may be different from one side to the other of the structure (F_1, F_2, F_3). Since by assumption the structure cannot fail, these fluids never come into contact and do not mix up with one another.

2.1 Conforming FE discretization

A spatial discretization must be chosen for the two sub-domains. The simplest case is that of a (nodally) conforming mesh, see Fig. 2b. In this case, to each node S of the structure there corresponds one node F of the fluid, and reciprocally (in case of more than one fluid sub-domain, this holds for each sub-domain). The nodes S and F are superposed, i.e. they have the same coordinates, but they are distinct (not merged). At a junction such as J there can be several fluid nodes (one for each fluid sub-domain) matching the same structure node.

The strong approach (*FSA algorithm* [2]) is illustrated in Fig. 3. Assuming an ALE formulation, there are *two* velocity vectors at each node, one for the particles ($\mathbf{v}$) and one for the mesh ($\mathbf{w}$). At each couple of matching F-S nodes the constraint on particle velocities reads:

$$\mathbf{v}_F \cdot \mathbf{n} = \mathbf{v}_S \cdot \mathbf{n}\,, \tag{1}$$

where subscript F indicates the fluid, subscript S indicates the structure and $\mathbf{n}$ is the unit normal to the *discrete* F-S interface at the concerned (fluid) node, see Fig. 3a. This condition on normal velocity components ensures that the fluid can neither traverse the structure wall, nor detach from it. The tangential components of the fluid and structure particle velocities are independent, so that the fluid (assumed inviscid) can freely slide along the structural walls. This is one of the arguments in favor of keeping the fluid and structure nodes distinct (though superposed) in the FSA algorithm: one single node would not provide enough degrees of freedom (dofs) for two distinct tangential velocity components. Computing the discrete normal(s) is not a trivial task in general, see [2-3] for a detailed discussion. In fact, the only (relatively) complex part of the algorithm is the automatic determination of the normal(s) in geometrically complex 3D cases, such as in the presence of junctions, free edges etc. especially when shell elements are used for the structure.

Relations of the form (1) are exactly enforced by Lagrange multipliers; see [2-3]. Thanks to implicit solution of the system of constraints, FSI conditions are automatically coupled with any other (compatible) essential conditions imposed by the user, e.g. blockages, symmetries, contact between solid bodies, etc. This is the only implicit part of the transient dynamic algorithm, which is otherwise completely explicit.

As concerns mesh velocity, which is arbitrary in ALE, the following compatibility condition is (trivially) imposed, see Fig. 3b:

$$\mathbf{w}_F = \mathbf{w}_S = \mathbf{v}_S \,, \tag{2}$$

ensuring that the fluid and the structure nodes at the F-S interface remain constantly superposed in the motion and deformation process. The second equality in (2) holds because a Lagrangian description is normally adopted for the structure. Then, the first equality provides the value of the fluid mesh velocity $\mathbf{w}_F$ along the F-S interface, necessary to complete the ALE description.

2.2 Non-conforming FE discretization

The FSA algorithm is easily generalized to the case of a nodally non-conforming F-S interface, see [4] and Fig. 4. This generalization is important in practical applications, because typically a finer mesh is needed in the fluid than in the structure—especially if the latter is meshed by shells—in order to obtain an accurate pressure field. The so-called *hierarchic* approach is usually convenient, see Fig. 4b, whereby to each structure node along the F-S interface there always corresponds a fluid node (*matching* nodes), but the inverse is not true in general (*non-matching* nodes) since F-S fluid nodes are more numerous than structure nodes.

For each couple of matching nodes, expressions (1) and (2) continue to hold. At a non-matching fluid node F (see Fig. 5) similar expressions can be written, involving the fluid node itself and the corresponding *point* S^* of the structure (note that this is *not* a node). For the particle velocity, in analogy with (1):

$$\mathbf{v}_F \cdot \mathbf{n}_F = \mathbf{v}_{S^*} \cdot \mathbf{n}_F \,, \tag{3}$$

where the local normal is now $\mathbf{n}_F$, perpendicular to the fluid domain. This normal is easily computed because the fluid domain is flat at non-conforming nodes. The constraint is then expressed in terms of nodal dofs (of the structure) by means of suitable shape functions N_I. For example, in the case of Fig. 5a the constraint (3) becomes:

$$\mathbf{v}_F \cdot \mathbf{n}_F = (N_1 \mathbf{v}_{S_1} + N_2 \mathbf{v}_{S_2}) \cdot \mathbf{n}_F \,. \tag{4}$$

Similarly, for the mesh velocity at non-matching nodes, see Fig. 5b, one has instead of (2):

$$\mathbf{w}_F = \mathbf{w}_{S^*} = \mathbf{v}_{S^*} = (N_1 \mathbf{w}_{S_1} + N_2 \mathbf{w}_{S_2}) \,. \tag{5}$$

2.3 Application to NCFV

The strong FSI algorithms presented so far fit particularly well in a FE discretization of the fluid sub-domain, because the corresponding constraints are imposed at fluid nodes, which coincide (spatially) with structure nodes (in the conforming case) or with other points of the structure (in the non-conforming case). The constraint (1) is quite "natural" in this case, because it expresses the physically intuitive condition that the fluid cannot traverse the solid wall, but is allowed to slide along it.

Exactly the same algorithms can be applied in the NCFV context [5-7], because also in that case fluid velocities are discretized at the "nodes" of the (primal) fluid mesh, which coincide with the FV centroids of the dual mesh, see Fig. 1b. Like in the FE case, enforcement of constraints is obtained via Lagrange multipliers, see [5-7] for details. Note, however, that in this case the constraint (1) is "stronger" than in the FE case, because $\mathbf{v}_F$ represents the velocity of the whole NCFV to which the fluid node belongs, while in FE it is the local value at the node. In fact, in FE the velocity is linearly interpolated over the element, while in NCFV (and also in CCFV) the velocity is uniform (just one value) over the entire finite volume.

3 FSI ALGORITHMS OF THE WEAK TYPE

The weak FSI algorithms considered here do not use constraints, but consist in applying directly to the structure the pressure forces generated by the fluid. These forces are interpreted as "external" forces in computing the equilibrium (in a dynamic sense) of the structure, and are assembled together with any other (user-imposed) external forces and with the internal forces resulting from the stress state in the structure.

The (ALE) fluid nodes at the F-S interface are tied to continuously follow the motion of the structure, and this provides a (weak) feed-back mechanism, influencing the physical state of the fluid. Like in the strong case, eqs. (2) or (5) are used to compute the mesh velocity at conforming or non-conforming fluid nodes at the F-S interface, respectively.

3.1 Conforming CCFV discretization

The weak FSI formulation seems particularly well suited for use with CCFV [8]. In the conforming case, illustrated in Fig. 6, the simplest technique is to merge together each couple of fluid and structure nodes along the F-S interface, rather than keeping them distinct. In this way, the pressure forces computed in the FV act directly on the structure thanks to the standard force assembly procedure. If p is the fluid pressure in the FV, L the length (area in 3D) of its face on the F-S interface, and $\mathbf{n}_S$ the unit normal to the structure (coincident with the normal to the fluid face), see Fig. 6a, the pressure force contribution for this volume is:

$$\mathbf{f}_p = pL\mathbf{n}_S \,. \tag{6}$$

This force is equally distributed on the nodes of the corresponding face, which belong both to the fluid and to the structure, and therefore contributes to the motion of the structure.

3.2 Non-conforming CCFV discretization

The weak FSI algorithm for a non-conforming (hierarchic) CCFV mesh is shown in Fig. 7. In this case, some fluid nodes on the F-S interface match a corresponding structure node, some don't. Each couple of matching nodes can be merged together, like in the conforming case, but this is not possible for the non-matching fluid nodes. Therefore, the algorithm is set up in such a way that the merging or not of the (matching) nodes has no effect on the solution. In Fig. 7a the matching nodes are kept distinct, for example.

Apart from this, the only difference with respect to the conforming case is the distribution of fluid pressure forces to the structure nodes, see Fig. 7b. Let p_i be the fluid pressure in the i-th FV, L_i the length of its face (area in 3D) on the F-S interface, and $\mathbf{n}_S$ the unit normal to the structure (coincident with the normal to the fluid face), see Fig. 7a. Then, the pressure force contribution for this volume is again given by (6), which becomes in this case:

$$\mathbf{f}_{p,i} = p_i L_i \mathbf{n}_S \,. \tag{7}$$

This force is supposed to act at the point C_i of the structure coinciding with the centroid of the fluid face (Fig. 7b). Therefore, in the 2D case, if $L_S = L_{Ai} + L_{Bi}$ is the length of the structure face, the force contributions at the two nodes A and B of the structure are:

$$\mathbf{f}_{p,Ai} = (L_{Bi}/L_S)\mathbf{f}_{p,i} \qquad ; \qquad \mathbf{f}_{p,Bi} = (L_{Ai}/L_S)\mathbf{f}_{p,i} \,. \tag{8}$$

3.3 Structural failure with CCFV

The CCFV formulation, in conjunction with the present weak FSI algorithm, lends itself to the treatment of structural failure even in the basic cases of conforming or non-conforming meshes (which is *not* the case for FE or CCFV, see Section 4).

Consider a structure separating two fluid sub-domains occupied by the *same type of fluid*, see Fig. 6b for the conforming and Fig. 7a for the non-conforming case. The key observation is that, since in CCFV the velocities are discretized at volume centers (not at nodes), the fluid meshes on either part of the structure can be set up as a *continuous mesh*. That is, assuming that fluid nodes occupy the same positions on either side of the structure, these nodes can be merged together two by two—and this independently from whether they are also merged with structure nodes and from whether the structure mesh is conforming or not with the fluid mesh—as shown in the Figures.

As long as the structure resists, each fluid FV transmits its pressure force to the structure and the corresponding volume face is considered as a boundary face (solid wall). Therefore, no mass or energy numerical fluxes (typical of FV technology) are computed across such faces, and the result is that the fluid cannot traverse the wall.

When the structure fails, the structural element is eroded and no longer takes part in the computation. To let the fluid freely flow across the (failed) wall, it is then sufficient to change the status of the corresponding FV face from solid wall (i.e. no neighbor FV) to internal fluid face. That is, the two FV on either part of the structure suddenly become neighbors.

This technique has been applied with success in some realistic simulations. It does extend the range of application of basic algorithms to the case of structural failure but, if the structure undergoes too large motions (in particular rotation) before failing, it suffers from the same limitations (fluid mesh entanglement) as the basic algorithms.

The condition that the same type of fluid be present in both sub-domains is necessary to let the two fluids mix up after structure failure, assuming that a single-material model is used in the formulation, i.e. a volume (or element) can only contain one type of material. This is often the case in real applications, think e.g. of explosions in buildings where the explosive gas products can often be modeled by the same constitutive equation as the gas (air) which fills and surrounds the building. When this is not the case, e.g. in under-water explosions, one can resort to using a multi-material model (but usually with an associated loss of precision).

3.4 Weak boundary condition algorithm for NCFV

A weak boundary condition algorithm has been implemented for the NCFV fluid model; see [5-7]. It consists of the following evaluation of numerical fluxes. Assume that a NCFV i, currently at pressure p_i, is adjacent to a "solid wall" (either fixed or moving at velocity $\mathbf{v}_S$) along a facet (portion of the FV boundary) of area F and of unit normal $\mathbf{n}$. Note that in such a case the node i representing the FV is located on the wall, see Fig. 1. Then the mass flux across the facet is set to zero, the momentum flux is given by $p_i F\mathbf{n}$ and the energy flux is given by $p_i F(\mathbf{v}_i \cdot \mathbf{n})$, where $\mathbf{v}_i$ is the fluid velocity in volume i.

This technique has been used with success in purely Eulerian calculations, where the (solid) wall is fixed ($\mathbf{v}_S = 0$) by definition and no real FSI takes place (the structure is not even modeled). Note that even in such a case using the above expression for the momentum flux does *not* guarantee, in general, that the normal component of the fluid velocity is exactly zero (i.e. that $\mathbf{v}_i \cdot \mathbf{n} = 0$). This is characteristic of a weak formulation and is in line with the fact that $\mathbf{v}_i$ represents the (average) velocity over the entire volume i, not the value at point i (i.e. at the wall).

3.5 Comparison of strong and weak FSI formulations

As shown in the previous Sections, the strong FSI approach is the most natural with a FE discretization of the fluid, because velocities are discretized at fluid nodes and they represent the local value. Exact enforcement of strong constraints by Lagrange multipliers guarantees high precision and consistency of numerical results. In the coupling process, the two sub-domains are treated at the same level (there is no "master" and no "slave") and the constraints ensure that both the internal forces and the masses of each sub-domain are taken into account in the (coupled) equilibrium equation.

The weak FSI approach is the most natural with a CCFV discretization of the fluid. In this case velocities are discretized at volume centers (and represent the average value for the entire volume), so that imposing strong constraints would be less appropriate than in the FE case. The coupling is qualified as weak because the fluid transmits a pressure force to the structure, but the feedback (structure action on the fluid) only occurs via the motion of the structure, which drives the numerical flux in the fluid at the F-S interface. In a certain sense, the structure acts as a "master", and the fluid as a "slave". Furthermore, the fluid mass is typically not taken into account in computing the structure motion (at least not in the non-conforming case). The effect can be negligible with low-density fluids (gases), but not so much with liquids.

The case of NCFV fluid discretization lies somewhat in between. Fluid velocities are discretized at nodes, like in FE, but represent the average value for the entire volume, like in CCFV. Both strong and weak FSI formulations can appear as natural in this case (and have been implemented), each one with its advantages and drawbacks.

4 EMBEDDED FSI ALGORITHMS

The basic FSI algorithms presented so far are built on the ALE formulation for the fluid sub-domain, which allows the fluid mesh to *constantly follow* the structure mesh at the F-S interface. Fluid nodes in the interior of the fluid sub-domain are typically moved (arbitrarily) by a suitable automatic mesh rezoning algorithm, whose primary goal is to prevent the fluid mesh from entangling. The technique is quite elegant and, as long as it can be applied, it provides highly accurate results with a comparably low computational effort (CPU time). However, it suffers from two limitations:

- If the structure motion, especially *rotation*, is too large, then the mesh rezoning algorithm is unable to keep the fluid mesh reasonably regular, and the calculation must stop.
- If the structure *fails* and is either simply eroded or replaced by a set of flying debris (e.g. a cloud of particles), then it is difficult to "sew" the two fluid sub-domains which were previously separated by the structure in order to let the fluid flow through.

The second limitation stems from the fact that in the basic algorithms the fluid nodes at the interface are kept distinct from structure nodes (because fluid velocities are discretized at nodes), and thus the two fluid sub-domains on either part of the structure are not merged together. A notable exception is the weak algorithm for CCFV, see Section 3.3, for which this second limitation does not hold. However, note that the first limitation applies also in the case of Section 3.3, and this considerably reduces the range of applicability of that algorithm.

To overcome these limitations (at the expense of lower accuracy, however), a basically different approach to FSI is taken, see Fig. 8: the structure sub-domain and the fluid sub-domain are discretized in a completely independent manner, and then the structure mesh is *embedded* (or immersed) within the fluid mesh, i.e. the two meshes are simply superposed. With this technique, it is no longer necessary to use an unstructured (conforming or non-conforming) fluid mesh. The fluid mesh can be structured, and even regular, see Fig. 8b. Furthermore, it is

no longer necessary to use an ALE formulation in the fluid sub-domain. This sub-domain can be simply Eulerian (fixed mesh), so that the (background) fluid mesh will never entangle.

In order to prescribe suitable FSI conditions (either in a strong or in a weak manner) between the two sub-domains, it is necessary to find the nodes (or points) of the fluid mesh to be coupled with the structure, i.e. which are sufficiently "close" to the structure. Note that this search is not trivial and must be repeated during the transient calculation as the structure moves and deforms. This must be done with an efficient fast search algorithm, otherwise the computational cost of the search becomes preponderant in realistic applications. For example, a bucket sort algorithm is used in the current implementation.

The *coupled fluid nodes* are defined as all fluid nodes located within the *influence domain* of the structure, the shaded area in Fig. 9a. This influence domain is built up at each time step in the following manner: first, a sphere (circle in 2D) is generated, centered on each node of the structure. The radius of these spheres can either be prescribed by the user, or be computed automatically based on the local size of the fluid mesh. Spheres should be big enough so that at least some fluid nodes fall within them, but not too big. Then, the spheres are connected by quadrangles in 2D (or by cones, prisms and hexahedra in 3D) in the way shown in Fig. 9a. The union of nodal spheres and of all other mentioned geometric entities is the influence domain, used to find the coupled fluid nodes (marked by circles in Fig. 9a). Note that, on one hand, too thin a domain (too small sphere radius) produces some fluid leakage (spurious fluid passage) across the structure. But, on the other hand, too thick a domain (too big sphere radius) ties too much fluid to the motion of the structure (too large "added mass"). Therefore, a good compromise must be found for the sphere radii.

4.1 Embedded FSI algorithm of the strong type

The strong form of the embedded FSI algorithm, called the *FLSR algorithm*, is suitable for use either with FE [10-11] or with NCFV [9]. In both cases the fluid velocities are discretized at nodes. Let F be a coupled fluid node, see Fig. 9b. Then, the closest point of the structure S^* (not a node, in general) is found and then a suitable constraint is imposed on particle velocities. Two alternatives have been tested. The first one is similar to eq. (4) and reads in this case:

$$\mathbf{v}_F \cdot \mathbf{n}_S = \mathbf{v}_{S^*} \cdot \mathbf{n}_S = (N_1 \mathbf{v}_{S_1} + N_2 \mathbf{v}_{S_2}) \cdot \mathbf{n}_S \,. \tag{9}$$

Note that in the present case the normal to the fluid domain $\mathbf{n}_F$ cannot be defined, so we use the normal to the structure $\mathbf{n}_S$ instead (assuming that it can be computed). This form of the constraint aims at leaving the fluid free to slide along the structure, like in the basic algorithms. However, numerical examples have shown that sometimes this may produce fluid leakage, i.e. spurious (non-physical) passage of fluid across the structure.

The second alternative form of the constraint reads simply:

$$\mathbf{v}_F = \mathbf{v}_{S^*} = N_1 \mathbf{v}_{S_1} + N_2 \mathbf{v}_{S_2} \,, \tag{10}$$

i.e. the fluid is tied to the structure in all directions (both along the normal and along the tangent). This condition is "stronger" than (9) and may lead to some non-physical loading of the structure along the tangent direction, but it has the advantage that it avoids fluid leakage.

4.2 Application to NCFV

The FLSR algorithm of Section 4.1 works very well with FE fluids [10-11], but when used as such with NCFV it tends to produce spurious fluid leakage, and this irrespective of the width of the chosen structure influence domain (radius of the spheres).

This behavior is due to the way in which transport terms, in particular the transport of density (mass) and of energy, are computed in the different fluid formulations. In FE, transport terms in the governing Euler equations (for conservation of mass and energy) depend (are proportional to) the relative velocity $\mathbf{u} = \mathbf{w} - \mathbf{v}$ between the grid (mesh) and the fluid particles at the fluid nodes (i.e. at the interface between two neighboring elements). Consider for a moment a rigidly blocked structure wall embedded in an Eulerian fluid domain. The grid velocity $\mathbf{w}$ is zero because of the Eulerian description, and the structure velocity $\mathbf{v}_S$ is also zero because the structure is blocked. The FLSR algorithm enforces condition (10) strongly, so that $\mathbf{v}_F$ is zero as well. The result is that the relative velocity $\mathbf{u}$ vanishes at the fluid node, thus no mass and energy are transported across the node (and across the blocked structure).

In NCFV (and more generally in FV) technology, transport terms are represented by so-called *numerical fluxes*. The calculation of such fluxes is more complex in FV than in FE formulations, since it depends on the space order of the chosen algorithm (first or second order) and on the chosen Riemann solver (e.g. Roe's solver or others). The result is that a vanishing relative velocity at a node (i.e. in the corresponding NCFV) does *not* ensure zero local mass and energy fluxes. This is due, *in primis*, to the fact that fluxes are computed not at nodes, but at inter-volume faces, using an interpolated value $\mathbf{u}_{ij}$ of the velocity. Thus, if two volumes i and j are neighbors and $\mathbf{u}_i$ is zero but $\mathbf{u}_j$ is not, then $\mathbf{u}_{ij} \neq 0$ and some mass and energy are exchanged between the two volumes. It would then seem that blocking *two* (or more) consecutive fluid nodes rather than just one (i.e. using a larger sphere radius in the FLSR algorithm) should avoid fluid leakage, because in this case the relative velocity $\mathbf{u}_{ij}$ is strictly zero. However, numerical tests show that unfortunately this is *not* the case (at least when Roe's solver is used): spurious fluid leakage is reduced but not eliminated.

The solution to the above problem consists in explicitly setting to zero the mass and energy numerical fluxes at locations "close" to the structure. Two alternatives have been tested. The first one consists in blocking fluxes between two NC volumes when *both* volumes (i.e. both fluid nodes F_1, F_2) are coupled with the structure, see Fig. 10a. The second one consists in blocking fluxes between two NC volumes when *at least one* of them is coupled with the structure, see Fig. 10b. The second condition is "stronger" than the first one (it blocks more fluxes).

4.3 Embedded FSI algorithm of the weak type

The weak form of the embedded FSI algorithm, called the *FLSW algorithm*, is best suited for use with CCFV. The structure influence domain is built up exactly like for the strong algorithm (FLSR). However, instead of searching for fluid nodes, it is convenient to search directly for the element-to-element interfaces—indicated simply as *faces* in following for brevity—located within the influence domain (*coupled faces*), see Fig. 11a. Recall in fact that in the CCFV case the fluid velocities are discretized at the volume centers, not at the nodes. The numerical fluxes, already introduced above for the NCFV case, are evaluated in this case at face centers, marked by small squares in Figs. 11 and 12.

According to the weak approach, the forces generated by fluid pressure have to be computed and transmitted to the structure. With reference to Fig. 11b, for each coupled face f located between two volumes V_1, V_2 currently at pressures p_1, p_2, the pressure drop force $\mathbf{f}_{\Delta p}$ is evaluated according to:

$$\mathbf{f}_{\Delta p} = (p_1 - p_2) L \mathbf{n}_f \,, \tag{11}$$

where L is the length (area, in 3D) of the face and $\mathbf{n}_f$ is the unit normal to the face. Then, as shown in Fig. 12a, the point S^* of the structure closest to the face center is computed. The force (11) is supposed to act at this point S^* and is distributed on the structure nodes A, B according to an expression (in 2D) similar to (8), where $L_S = L_A + L_B$:

$$\mathbf{f}_{\Delta p,A} = (L_B / L_S)\mathbf{f}_{\Delta p} \qquad ; \qquad \mathbf{f}_{\Delta p,B} = (L_A / L_S)\mathbf{f}_{\Delta p} \,. \tag{12}$$

Then, exactly as in the NCFV case discussed in Section 4.2, in order to avoid spurious fluid leakage across the structure it is necessary to inhibit the mass and energy numerical fluxes at each coupled face, as illustrated in Fig. 12b (thick shaded line).

Note that expression (11) is the weak equivalent of (10), i.e. the fluid and the structure are coupled in all directions. Alternatively, one may project the pressure drop force along the normal to the structure $\mathbf{n}_S$ (assuming that this can be computed), obtaining the weak equivalent of (9). However, like in the strong case, such an expression risks to produce fluid leakage.

5 NUMERICAL EXAMPLES

Three numerical examples are presented. The first one is an academic problem comparing all the FSI algorithms presented in a relatively simple case: an internal explosion in a 2D rectangular box, with relatively small structural deformations and without structural failure. The second and third examples are realistic applications: blast effects on a railway station with a glass roof (which completely fails), and on a metallic container placed in a long tunnel.

5.1 Explosion in a metallic box

An elasto-plastic deformable box in 2D (plane strain) geometry, see Fig. 13a, contains a 0.4 m long deformable internal wall. The box measures 2 m width by 1 m height, and is filled by air at atmospheric pressure (1 bar). The left compartment contains an explosive bubble of 0.4 m by 0.4 m, represented by air at a higher initial pressure (10 bar). Externally to the box acts a constant atmospheric pressure (1 bar). The structure is uniformly meshed by 64 thin shell elements each 0.1 m long and with a thickness of 0.01 m. The fluid mesh within the box is a uniform grid of 3,200 (i.e. 80×40) four-node quadrilaterals of size 0.025 m, i.e. $1/4$ of the shell element length, so that non-conforming (hierarchic) FSI is used.

This test problem is taken from [9], where FE, NCFV and CCFV solutions were presented, obtained with the basic (non-conforming) strong and weak FSI algorithms, respectively. Here three solutions are added and compared, obtained with FE, NCFV and CCFV using the strong and weak embedded FSI algorithms, respectively. In the embedded case the fluid mesh extends all around the box for an additional 0.2 m. The total number of fluid elements is therefore 5,376 (96×56). A layer of absorbing elements is placed along the external boundary of the fluid mesh in order to simulate an infinite atmosphere and to prevent spurious pressure wave reflections.

All FV solutions shown are second-order in space and time. Calculations are performed up to 5 ms. This is sufficient to see propagation of the pressure waves to the top and to the right part of the box and several reflections. The structural displacements computed at points P9S, P10S and P12S of Fig. 13a are compared in Figs 13b, 13c and 13d, showing very good agreement of all six solutions.

Then, Figs. 14, 15, 16 and 17 compare the fluid velocity field (not shown in the explosive bubble region) and the fluid pressure field, at an intermediate time and at the final time. The agreement is very good for the first three solutions, which use the basic non-conforming FSI

algorithms. The other three solutions, which use the (new) embedded FSI algorithms, show a larger variability, but this has to be expected, given the lower accuracy of such algorithms (for the same fluid mesh). The shown embedded techniques are just a first implementation and can certainly be ameliorated, but already now one can say that results are consistent with the basic algorithms (see the comparison of box displacements in Fig. 13), so that the embedded algorithms can be used to predict structural behavior in applications with failing structures.

5.2 Blast effects in a railway station

Blast effects in railway stations, metro stations and rolling stock have been recently studied at JRC within the RailProtect project; see e.g. [12-14]. In Fig. 18a an historical railway station with a glass roof supported by metallic trusses is shown. A big explosion at the center of the main hall is simulated. Most roof glass panels completely fail and are replaced by clouds of flying debris particles in the simulation.

The numerical model uses about 4.5 million elements, of which more than 90% are fluid FE. The simulation up to 250 ms takes about 2 weeks on a PC (single-processor code version). The FLSR algorithm is used for FSI because of structure failure and fragmentation and of very large rotations of the (macroscopic) flying debris. Figs. 18b and 18e show the model before blast, while Figs 18c and 18f show blast effects at 100 ms. Fig. 18d shows the map of death risk [12-13] for the occupants of the station, automatically computed from blast overpressure and impulse time histories in the fluid sub-domain.

5.3 Blast effects on an ISO container

Standard 20 ft ISO containers are sometimes used as first shelters in peacekeeping, rescue and reconstruction operations. The resistance of such containers to blast loading is studied at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway [15-16]. Blast tests were carried out in the Large Blast Simulator (LBS) at the Bundeswehr Technical Center for Protective and Special Technologies (WTD 52) in Oberjettenberg, Germany.

Fig. 19a shows one of the experiments in the LBS. The container is placed at one end of a 100 m long tunnel. A blast wave is generated at the other end, propagates along the tunnel and hits the container. Large motions and plastic deformation occur in the container, but failure is much localized and no fragmentation occurs. Fig. 19b shows the FE model of the container (~50 thousand FE), and Fig 19c shows the container within the tunnel. The air in the tunnel is meshed by about 0.6 million fluid FE or FV.

Simulation of the container alone by applying the pressure recorded in the test (no fluid in the model) reproduces well the experiment, see Fig. 19d. However, in order to make predictive simulations the fluid must be modeled as well. Figs. 19e and 19f show two such simulations, for different load cases. The first one uses a FE model of the fluid, and the FLSR algorithm. The second one uses a CCFV model of the fluid—more accurate and less diffusive than FE in modeling the very long propagation of the blast wave in the tunnel—and the FLSW algorithm. These (purely demonstrative) simulations take from 3 to 9 weeks on a PC (single-processor code version) up to 280 or 700 ms of physical time, respectively.

6 CONCLUSIONS AND PERSPECTIVES

The strong and weak approaches to fluid-structure interaction in fast transient dynamics have been shortly reviewed. Examples of basic (either conforming or non-conforming) and of embedded FSI algorithms have been presented and their application with FE, NCFV or CCFV fluid discretization has been discussed.

The embedded FSI algorithms are the latest ones developed and have not yet reached full maturity and validation, at least not at the same level as the basic algorithms. However, numerical results are encouraging and, in any case, these are the only algorithms applicable in extreme loading cases with complete structure failure and fragmentation, typical of blast effect studies.

These algorithms are more general than the basic ones but, as a counterpart, they are less accurate for the same size of the fluid cells (elements or volumes). Therefore, along with improvement of the embedded algorithms by thorough numerical testing, future work will concentrate on implementing mesh adaptivity in the fluid sub-domain, i.e. automatic mesh refinement in fluid zones near an embedded structure wall.

REFERENCES

[1] The EUROPLEXUS manual: http://europlexus.jrc.ec.europa.eu/.

[2] F. Casadei, J. P. Halleux, An algorithm for permanent fluid-structure interaction in explicit transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, **128**/3-4, 231-289, 1995.

[3] F. Casadei, J.P. Halleux, A. Sala, F. Chillè, Transient fluid-structure interaction algorithms for large industrial applications. *Computer Methods in Applied Mechanics and Engineering*, **190**/24-25, 3081-3110, 2001.

[4] F. Casadei, S. Potapov, Permanent fluid-structure interaction with non-conforming interfaces in fast transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, **193**/39-41, 4157-4194, 2004.

[5] A. Soria, F. Casadei, Arbitrary Lagrangian-Eulerian multicomponent compressible flow with fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, **25**, 1263-1284, 1997.

[6] A. Sala, F. Casadei, A. Soria, A 3D finite volume numerical model of compressible multicomponent flow for fluid-structure interaction applications. *4th Congreso de Métodos Numéricos en Ingenieria*, Sevilla, Spain, June 7–10, 1999.

[7] F. Casadei, A. Sala, Finite element and finite volume simulation of industrial fast transient fluid-structure interactions. *European Conference on Computational Mechanics – Solids, Structures and Coupled Problems in Engineering (ECCM '99)*, Munich, Germany, August 31 – September 3, 1999.

[8] P. Galon, V. Faucher, F. Casadei, S. Potapov, Modeling complex fluid-structure interaction problems with EUROPLEXUS. *8th World Congress on Computational Mechanics* (*WCCM8)*, Venice, Italy, June 30 – July 5, 2008.

[9] F. Casadei, N. Leconte, Coupling finite elements and finite volumes by Lagrange multipliers for explicit dynamic fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, DOI: 10.1002/nme.3042 (in publication).

[10] F. Casadei, Fast transient fluid-structure interaction with failure and fragmentation. *8th World Congress on Computational Mechanics* (*WCCM8)*, Venice, Italy, June 30 – July 5, 2008.

[11] F. Casadei, M. Larcher, On some computational methods for the simulation of structures subjected to blast loading and fragmentation. *4th European Conference on Computational Mechanics (ECCM2010)*, Paris, France, May 16–21, 2010.

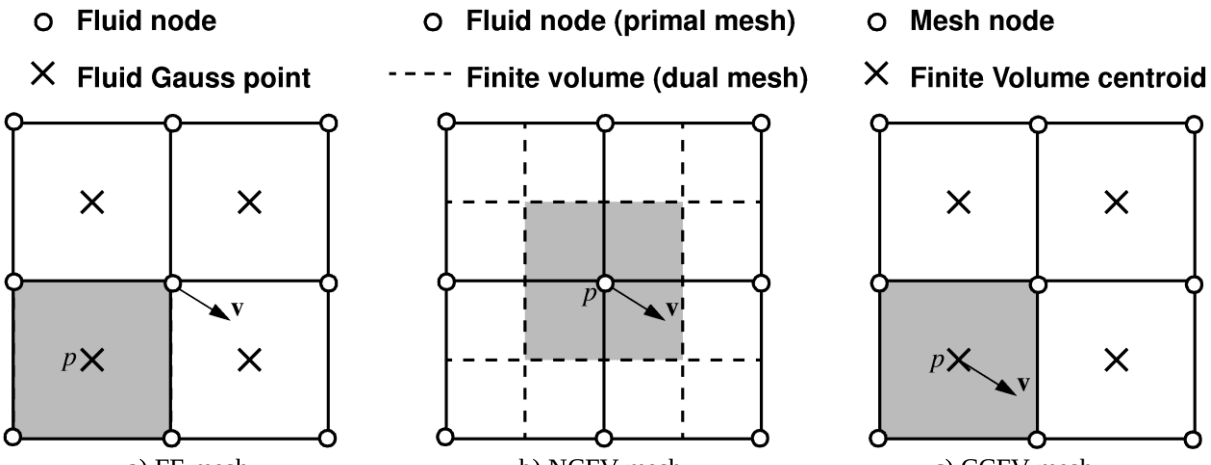
[12] G. Giannopoulos, M. Larcher, F. Casadei and G. Solomos, Risk analysis of the fatality in land mass transport infrastructure by fast transient dynamic analysis. *Journal of Hazardous Materials*, **173**, 401-408, 2010.

[13] M. Larcher, F. Casadei and G. Solomos, Risk analysis of explosions in trains by fluid-structure calculations. *Journal of Transportation Security*, **3**, 57-71, 2010.

[14] M. Larcher, F. Casadei and G. Solomos, Influence of venting areas on the air blast pressure inside tubular structures like railway carriages. *Journal of Hazardous Materials*, **183**, 839-846, 2010.

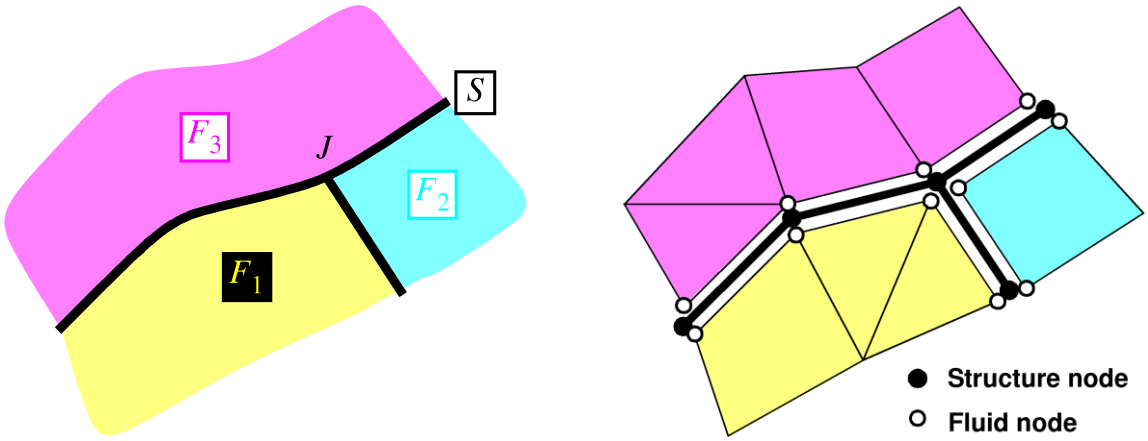
[15] T. Børvik, A. Burbach, H. Langberg, M. Langseth, On the ballistic and blast load response of a 20 ft. ISO container protected with aluminum panels filled with local mass – Phase II: Validation of protective system. *Engineering Structures*, **30**/6, 1621-1631, 2008.

[16] T. Børvik, A.G. Hanssen, M. Langseth, L. Olovsson, Response of structures to planar blast loads – A finite element engineering approach. *Computers and Structures*, **87**, 507-520, 2009.



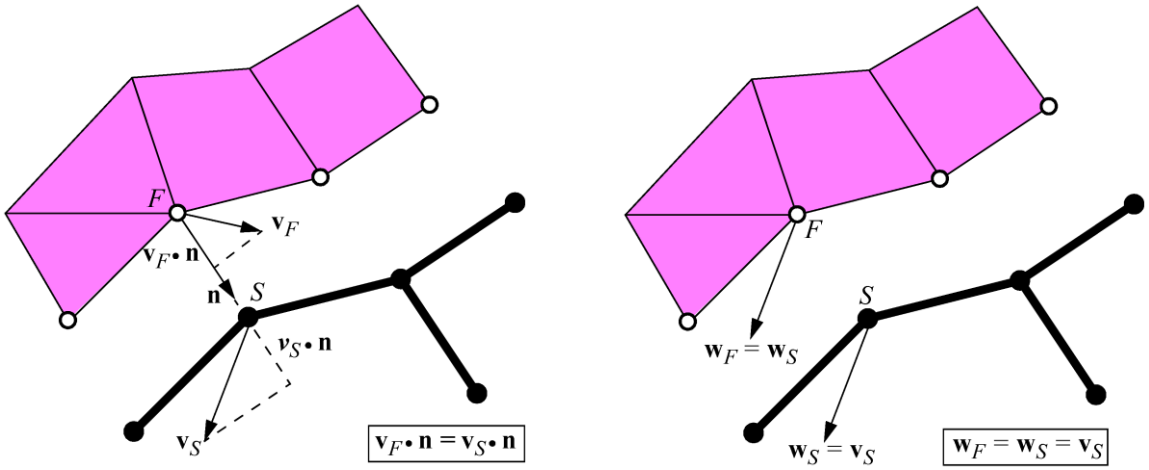
a) FE mesh. b) NCFV mesh. c) CCFV mesh.

Fig. 1 – Finite-element, node-centred finite volume and cell-centred finite volume meshes.



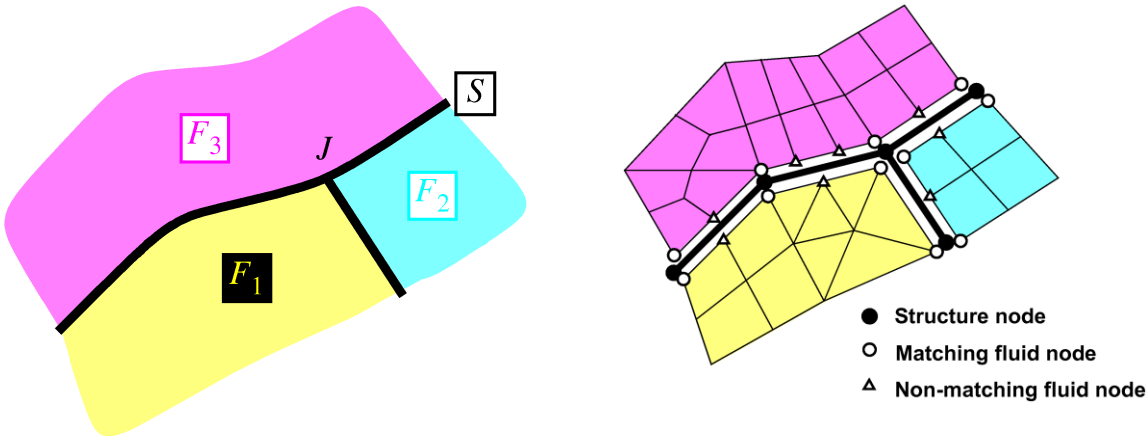
a) Continuous F-S domain. b) Nodally conforming discretization.

Fig. 2 – Sample FSI problem and its conforming discretization.



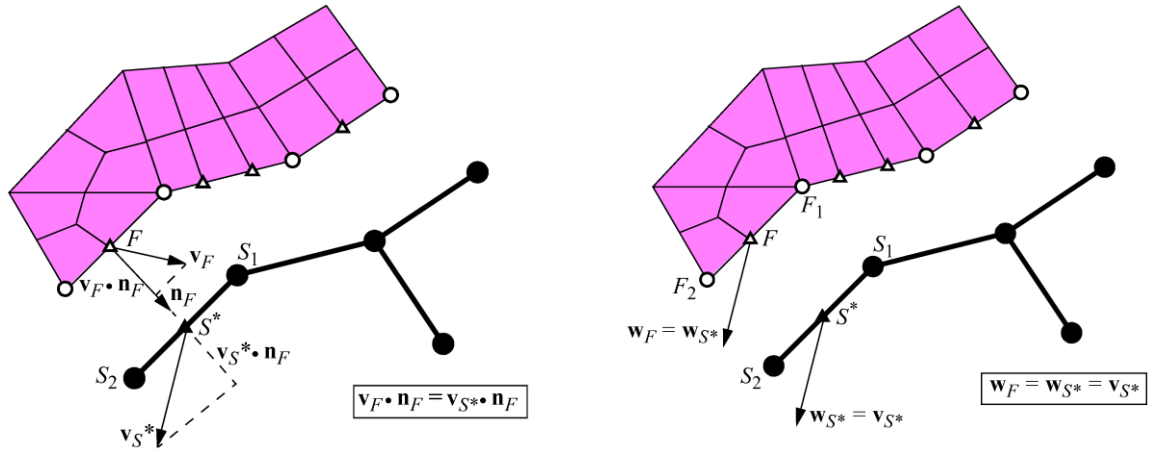
a) Constraints on particle velocities. b) Compatibility of mesh velocities.

Fig. 3 – FSA algorithm (strong approach) for a conforming F-S mesh.



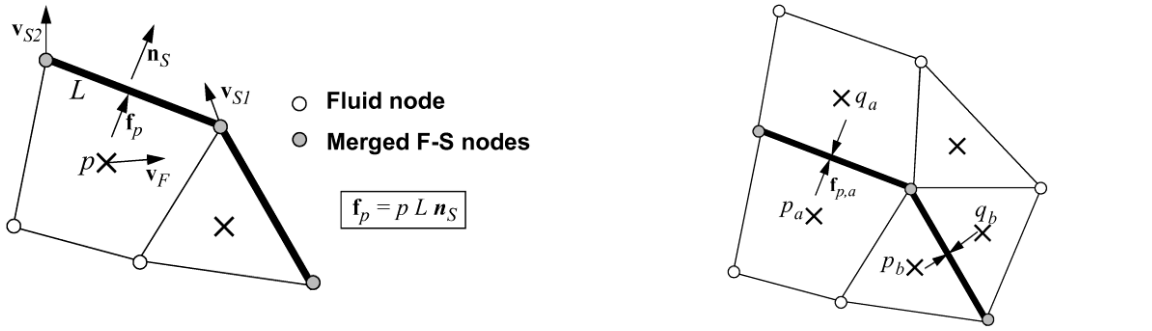
a) Continuous F-S domain. b) Non-conforming (hierarchic) discretization.

Fig. 4 – Sample FSI problem and its non-conforming discretization.



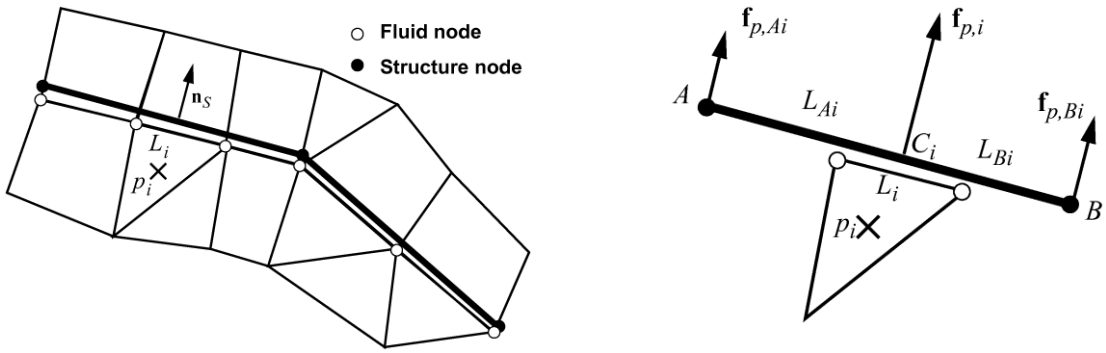
a) Constraints on particle velocities. b) Compatibility of mesh velocities.

Fig. 5 – FSA algorithm (strong approach) for a non-conforming F-S mesh.



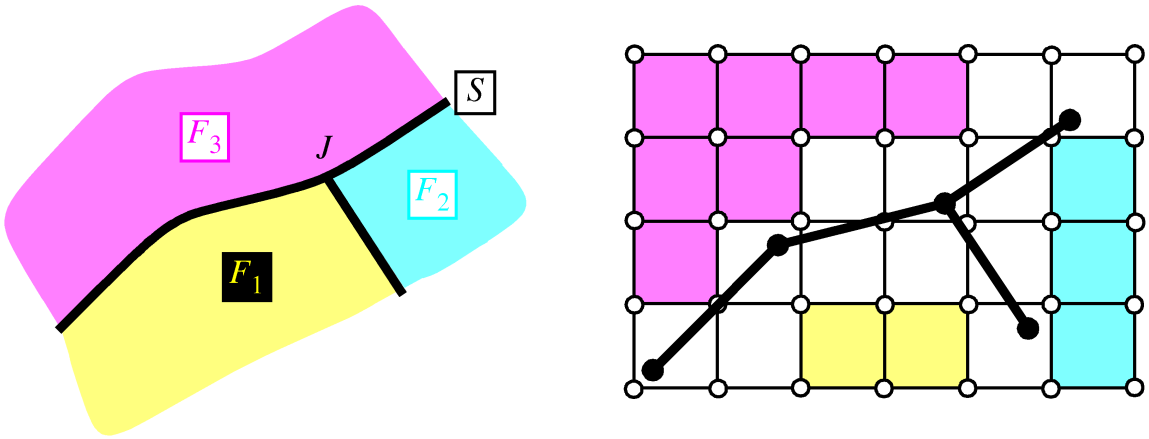
a) One fluid. b) Two fluids.

Fig. 6 – Weak algorithm for a conforming F-S mesh (with merged nodes).



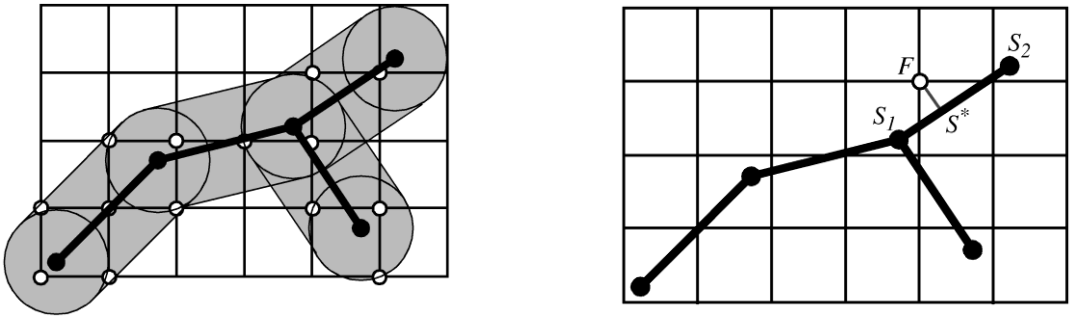
a) Hierarchic mesh, two fluids (continuous mesh). b) Distribution of pressure force.

Fig. 7 – Weak algorithm for a non-conforming F-S mesh.



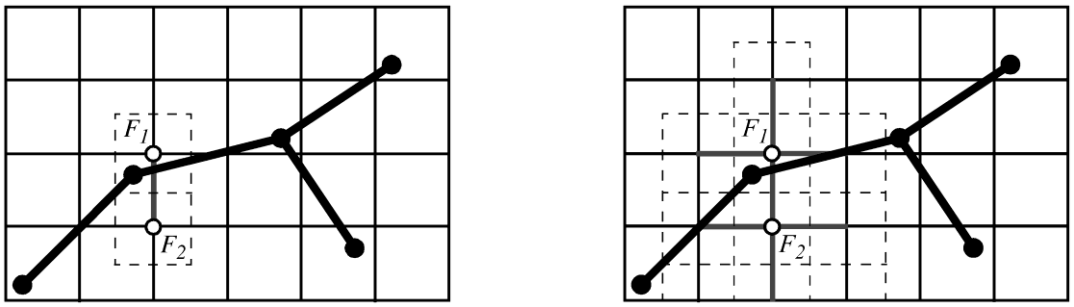
a) Continuous F-S domain. b) Structure mesh embedded in the fluid.

Fig. 8 – Sample FSI problem and its embedded discretization.



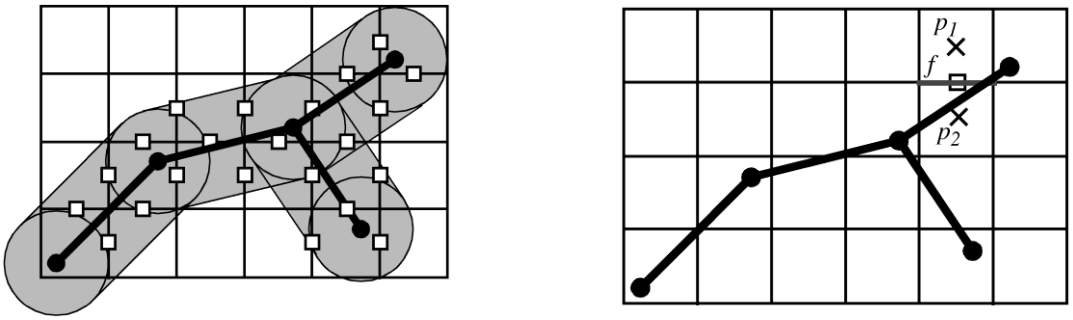
a) Influence domain (shaded) of the structure. b) F-S coupling.

Fig. 9 – Embedded FSI algorithm (strong approach).



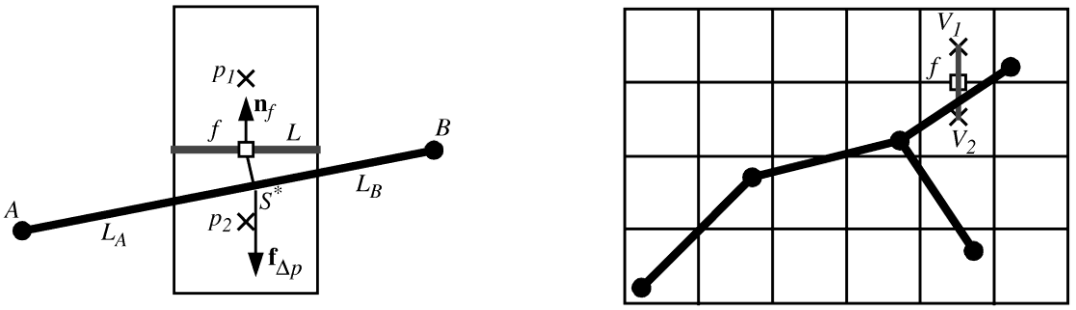
a) NCFV flux blockage (thick shaded line). b) More restrictive flux blockage.

Fig. 10 – Embedded FSI algorithm (strong approach) with NCFV.



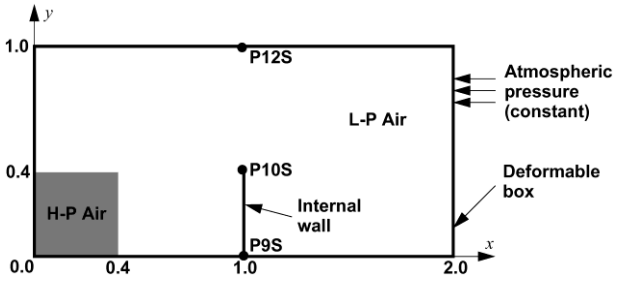
a) CCFV faces in the influence domain. b) Pressure drop force calculation.

Fig. 11 – Embedded FSI algorithm (weak approach) with CCFV.

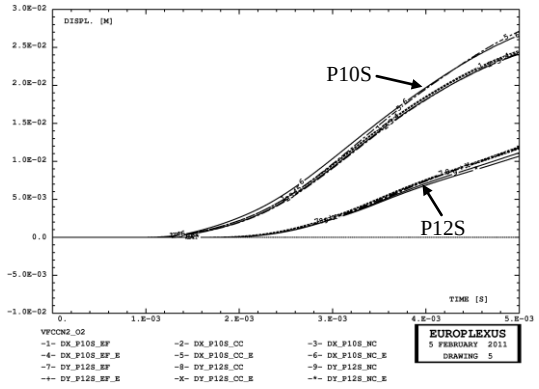


a) F-S coupling force. b) CCFV flux blockage (thick shaded line).

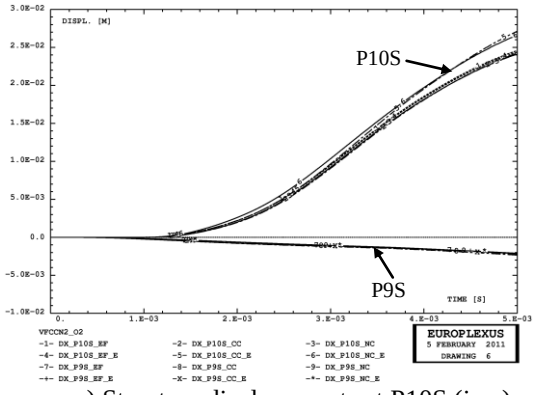
Fig. 12 – Embedded FSI algorithm (weak approach) with CCFV.



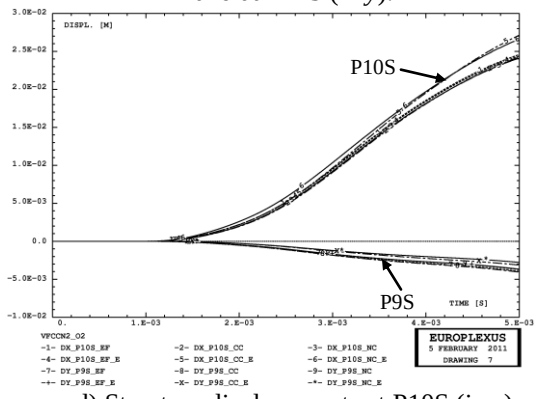
a) Geometry.



b) Structure displacements at P10S (in *x*) and at P12S (in *y*).



c) Structure displacements at P10S (in *x*) and at P9S (in *x*).



d) Structure displacements at P10S (in *x*) and at P9S (in *y*).

Fig. 13 – Explosion in a metallic box.

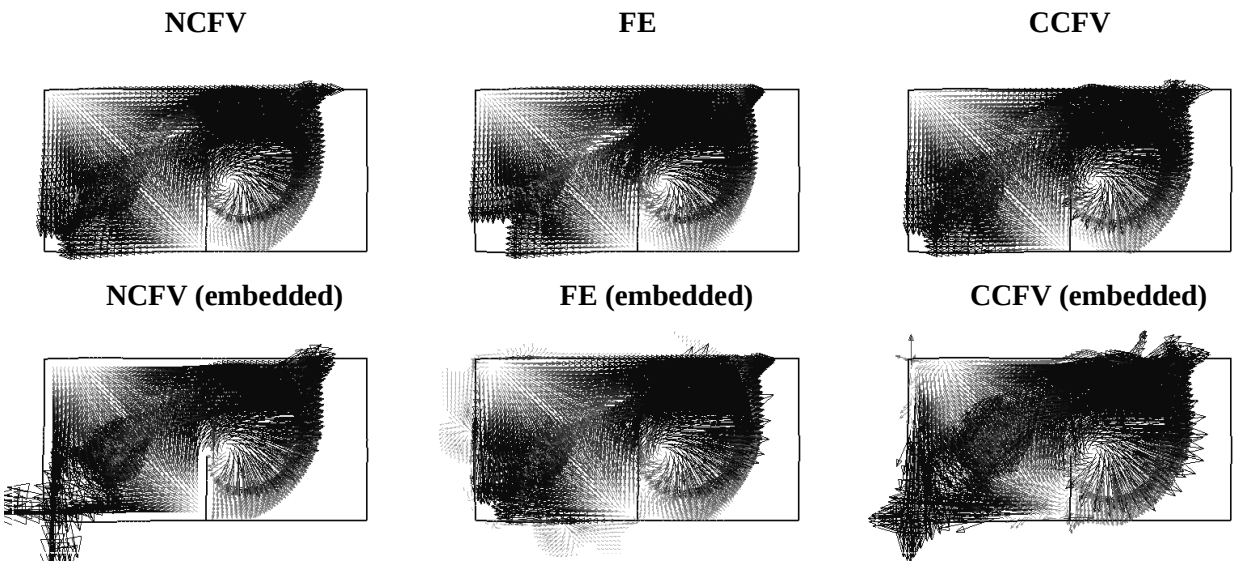


Fig. 14 – Explosion in a metallic box: fluid velocity at 2.5 ms.

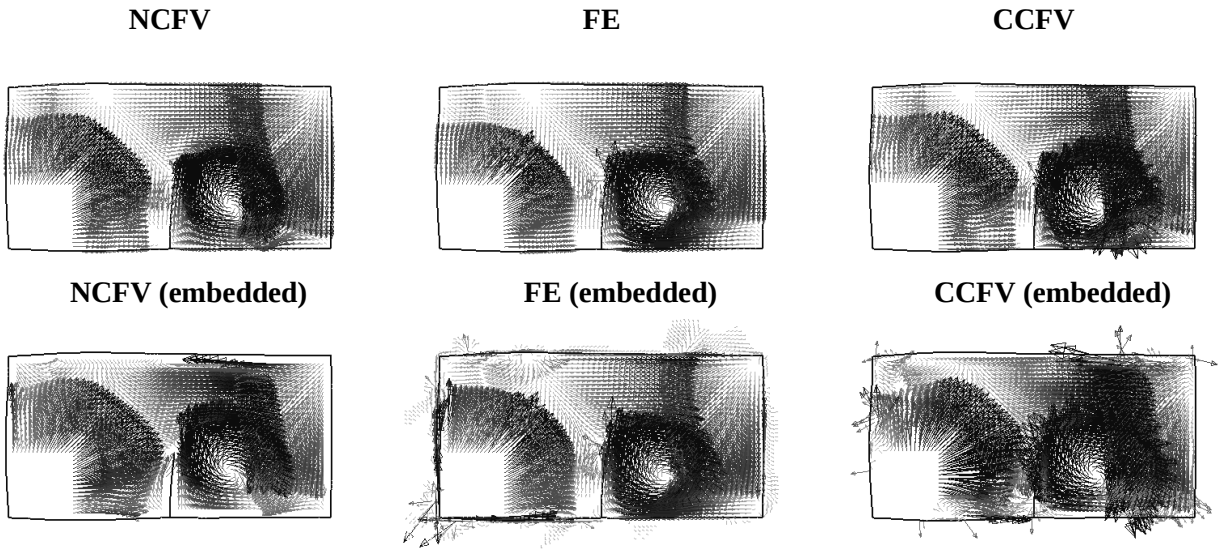


Fig. 15 – Explosion in a metallic box: fluid velocity at 5.0 ms.

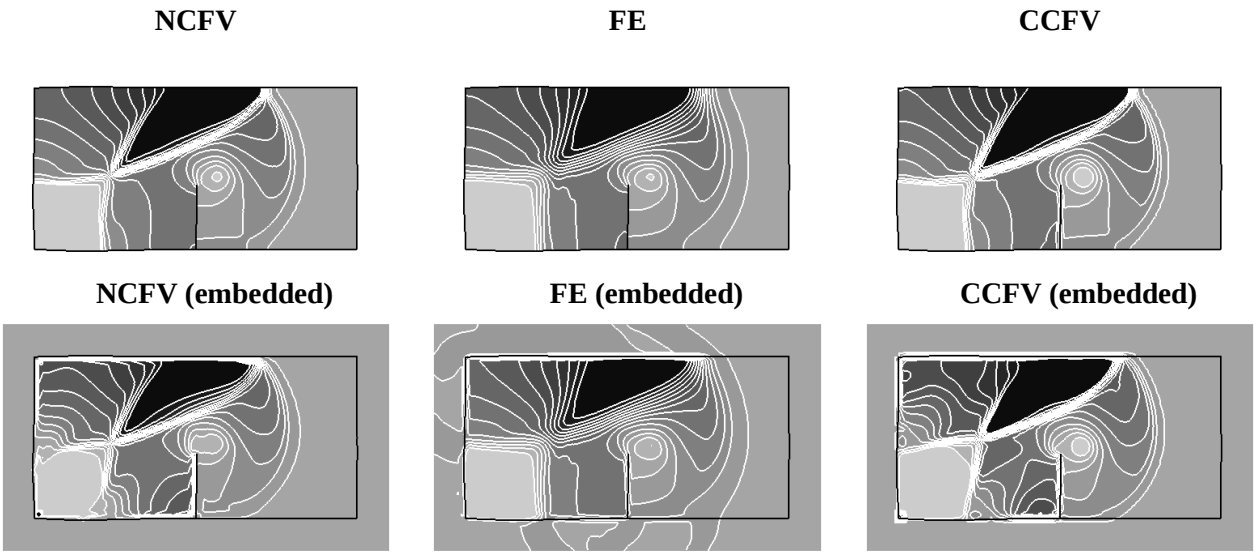


Fig. 16 – Explosion in a metallic box: fluid pressure at 2.5 ms.

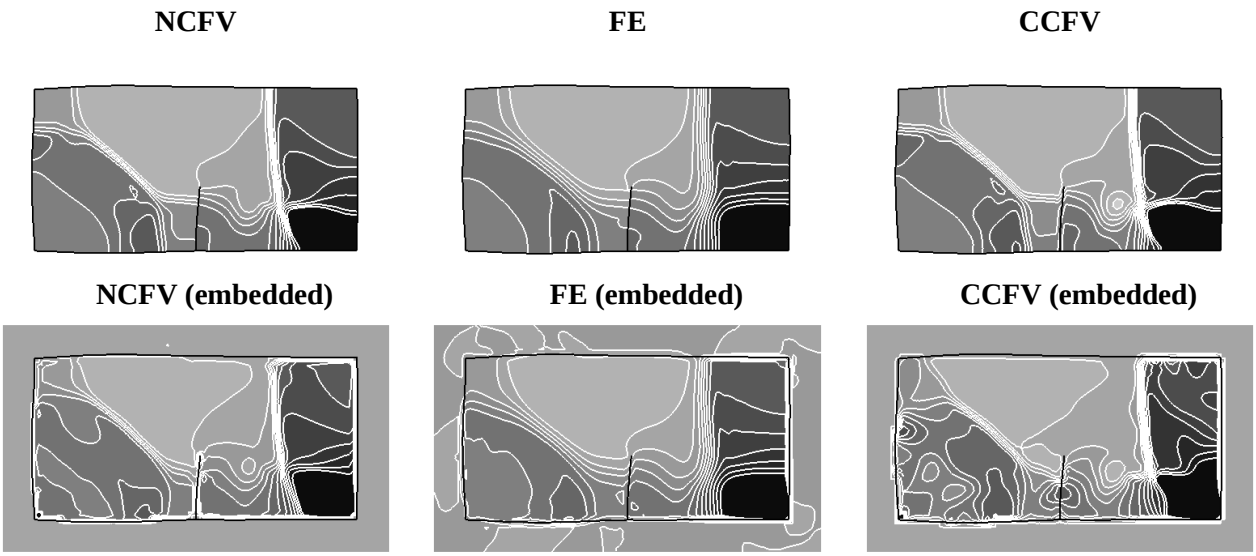
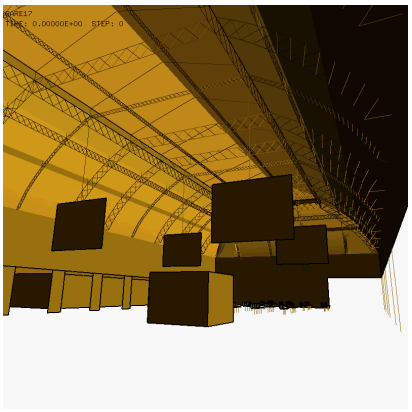
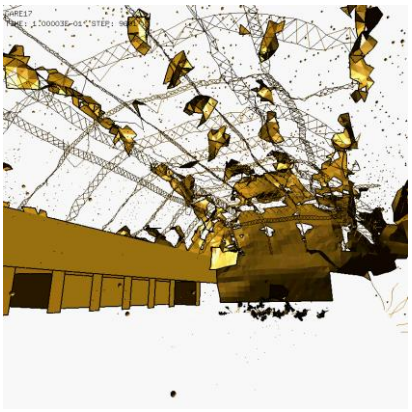


Fig. 17 – Explosion in a metallic box: fluid pressure at 5.0 ms.

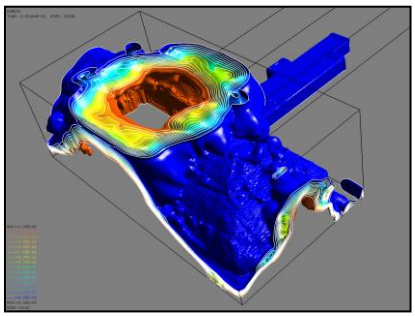
a) Glass roof.



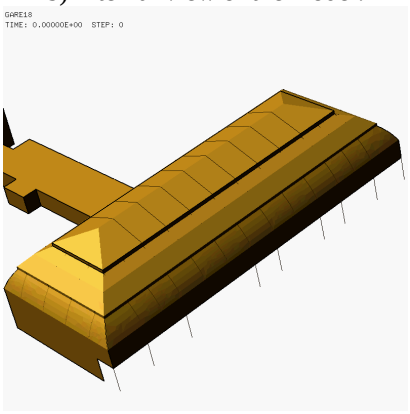
b) Internal view of the model.



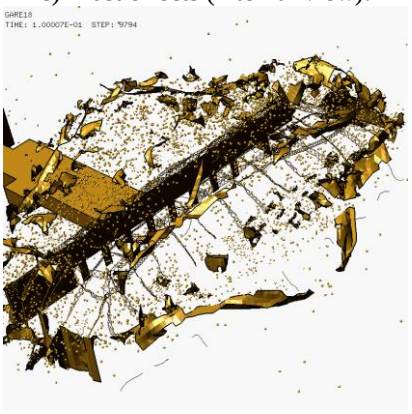
c) Blast effects (internal view).



d) Risk map.



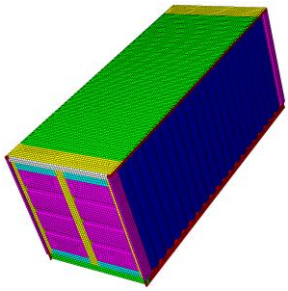
e) External view of the model.



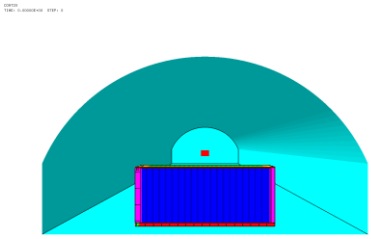
f) Blast effects (external view).

Fig. 18 – Blast effects in a railway station.

a) Experiment (case 1).

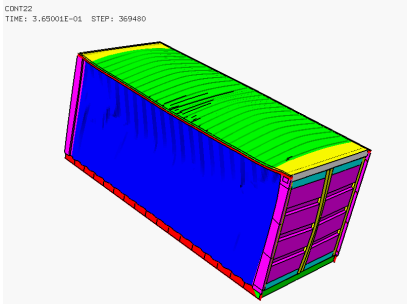


b) Container model.

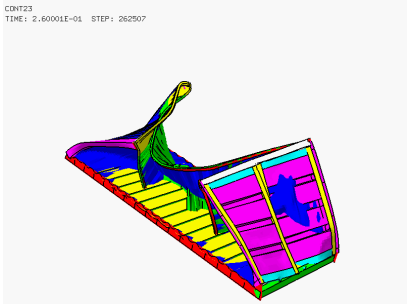


c) Container in the tunnel.

d) Blast effects (applied pressure).



e) Blast effects (case 1).



f) Blast effects (case 2).

Fig. 19 – Blast effects on an ISO container.