Comput Mech (2017) 60:83–100
DOI 10.1007/s00466-017-1394-3

ORIGINAL PAPER

A new formulation for air-blast fluid–structure interaction using an immersed approach. Part I: basic methodology and FEM-based simulations

Y. Bazilevs[1] · K. Kamran[1] · G. Moutsanidis[1] · D. J. Benson[4] · E. Oñate[2,3]

Received: 3 December 2016 / Accepted: 20 February 2017 / Published online: 15 March 2017

Abstract In this two-part paper we begin the development of a new class of methods for modeling fluid–structure interaction (FSI) phenomena for air blast. We aim to develop accurate, robust, and practical computational methodology, which is capable of modeling the dynamics of air blast coupled with the structure response, where the latter involves large, inelastic deformations and disintegration into fragments. An immersed approach is adopted, which leads to an a-priori monolithic FSI formulation with intrinsic contact detection between solid objects, and without formal restrictions on the solid motions. In Part I of this paper, the core air-blast FSI methodology suitable for a variety of discretizations is presented and tested using standard finite elements. Part II of this paper focuses on a particular instantiation of the proposed framework, which couples isogeometric analysis (IGA) based on non-uniform rational B-splines and a reproducing-kernel particle method (RKPM), which is a Meshfree technique. The combination of IGA and RKPM is felt to be particularly attractive for the problem class of interest due to the higher-order accuracy and smoothness of both discretizations, and relative simplicity of RKPM in handling fragmentation scenarios. A collection of mostly 2D numerical examples is presented in each of the parts to illustrate the good performance of the proposed air-blast FSI framework.

✉ Y. Bazilevs
yuri@ucsd.edu

[1] Department of Structural Engineering, University of California, San Diego, La Jolla, CA 92093, USA

[2] Centre Internacional de Métodes Numérics en Enginyeria (CIMNE), 08034 Barcelona, Spain

[3] Universitat Politècnica de Catalunya, Barcelona, Spain

[4] Livermore Software Technology Corporation, Livermore, CA 94550, USA

Keywords Air blast · FSI · Immersed methods · FEM · IGA · RKPM

1 Introduction

In this two-part paper, motivated by applications stemming from extreme events occurring as a result of man-made or natural disasters, we begin the development of a new class of methods for modeling fluid–structure interaction (FSI) phenomena in the regime of explosions or blast. In this setting, we aim to develop accurate, robust, and practical computational methodology, which is capable of modeling the dynamics of air blast coupled with the structure response, where the latter involves not only large, inelastic deformations, but also disintegration into fragments, which typically occurs during explosions.

In the development of the aforementioned framework, the following modeling and computational challenges need to be addressed: (1) the air flow is in the regime of high Reynolds and Mach numbers, requiring the fluid mechanics numerical formulation to be simultaneously accurate and robust; (2) the solid objects undergo large inelastic deformations, come in and out of contact with one another, and often fragment into smaller pieces, which requires advanced modeling and discretization techniques for the solid itself, as well as the good management of the fluid mechanics domain and mesh; (3) at each time instant the solid and fluid need to have kinematics and tractions in equilibrium to ensure correct coupling between the two subsystems, requiring appropriate coupling at the space-discrete level, and an efficient solution strategy for the coupled system; (4) the coupled system undergoes rapid transients, which need to be accurately and efficiently captured.

To address the above challenges, a novel air-bast FSI formulation is developed based on the immersed methodology. Moving-mesh methods, such as the Arbitrary Lagrangian–Eulerian (ALE) [1–6] or Space–Time (ST) [4,7–27] approaches are not pursued in the present work largely due to the challenges arising in the item 2 above. While the advantage of moving-mesh methods over immersed-type techniques lies in their ability to deliver higher fluid-mechanics accuracy near the fluid–structure interfaces, we forego this advantage in favor of a relative ease with which immersed-type methods handle topological changes arising from structural contact or fragmentation. It should be noted, however, that recently developed ST–TC techniques [28–32], which are ST methods that are able to handle topology changes, may provide a pathway for the application of moving-mesh methods to air-blast FSI.

In the proposed method, the Navier–Stokes equations of compressible flows are coupled with a large-deformation inelastic solid. Balance equations for both media are written in the weak form on the current configuration, and the compatibility of kinematics and tractions is established at a fully continuous level. Two discretizations, background and foreground, are employed in the discrete formulation. Background discretization, which is assumed fixed, provides the basis functions used to approximate the unknowns of the coupled FSI problem. Foreground discretization, which moves with the solid material particles, is employed to track the solid current position, store the history-dependent variables needed in the solid formulation, and carry out numerical quadrature for the solid terms in the coupled FSI formulation. Pressure-primitive variables [33,34], which are compatible with the solid degrees-of-freedom (DOFs), are employed to discretize the compressible-flow equations. SUPG [35–39] and residual-based discontinuity capturing [40–46] are used to stabilize the compressible-flow formulation.

In this setting, since the background mesh does not track the fluid–solid interface, no formal restrictions on the solid motion are imposed in the coupled formulation. In addition, due to the fact that the fluid and solid share the same background-mesh DOFs, the resulting FSI formulation is a-priori monolithic, even in the case of explicit time integration, which is what we do here to efficiently handle the fast dynamics of air-blast FSI. This a-priori monolithic coupling is a major advantage over existing methods using explicit time stepping in combination with the so-called "loosely-coupled" FSI, in which, unlike in the present formulation, there is no guarantee that the fluid and solid tractions are in equilibrium during the time step.

The proposed methodology shares similarities with the existing immersed-type techniques, such as embedded domain [47–54], immersed boundary [55,56], immersed finite element [57,58], material point [59,60], finite cell [61], and immersogeometric [62–64] methods.

Part I of this paper presents the core air-blast FSI methodology, which is suitable for discretization using standard FEM as well as other techniques, such as, for example, isogeometric analysis (IGA) [65,66]. Part II of this paper provides a particular instantiation of the proposed framework, which couples IGA as the background discretization and the reproducing-kernel particle method (RKPM) [67] as the foreground discretization. This combination is particularly attractive for the problem class of interest due to the higher-order accuracy of both techniques and relative simplicity of RKPM in handling fragmentation scenarios.

Part I of this paper is outlined as follows. In Sect. 2, the governing equations of the fluid and solid mechanics, and their coupling, are presented at the continuous level. Space discretization of the coupled FSI system and time integration are presented in Sect. 3. In Sect. 4 several 2D numerical examples using low-order FEM are shown, which illustrate the good performance of the proposed air-blast FSI methodology.

2 Governing equations of fluid and solid mechanics, and their coupling

In this section we present the governing equations of compressible flow and inelastic solid, and their coupling. All the developments in this section take place at the continuum level, and a weak form of the coupled FSI problem is derived that is suitable for discretization using an immersed technique.

2.1 Compressible-flow formulation

The air blast phenomena are modeled using the Navier–Stokes equations of compressible flows, which may be expressed as

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} - \mathbf{S}^f = \mathbf{0}, \tag{1}$$

where

$$\mathbf{U} = \begin{pmatrix} \rho^f \\ \rho^f \mathbf{u} \\ \rho^f e_{tot} \end{pmatrix} \tag{2}$$

are the so-called conservation variables [33], ρ^f is the fluid density, $\mathbf{u}$ is the material-particle velocity, $e_{tot} = e + \frac{\|\mathbf{u}\|}{2}$ is the total energy density, and e is the internal energy density given by $e = c_v T$, where c_v is specific heat at constant volume and T is the temperature. In Eq. (1), $\mathbf{S}^f$ is the source term and $\mathbf{F}_i$ is the total flux that consists of advective, pressure, and diffusive contributions, namely,

$$\mathbf{F}_i = \mathbf{F}_i^a + \mathbf{F}_i^p - \mathbf{F}_i^d. \tag{3}$$

Here,

$$\mathbf{F}_i^a = u_i \mathbf{U}, \tag{4}$$

$$\mathbf{F}_i^p = \begin{pmatrix} 0 \\ p\mathbf{1}_i \\ \mathbf{u} \cdot p\mathbf{1}_i \end{pmatrix}, \tag{5}$$

and

$$\mathbf{F}_i^d = \begin{pmatrix} 0 \\ \boldsymbol{\tau}_i \\ \mathbf{u} \cdot \boldsymbol{\tau}_i - q_i, \end{pmatrix}, \tag{6}$$

where p is the pressure, $\mathbf{1}_i$ is the ith Cartesian basis vector in $\mathbb{R}^d$, d is the space dimension, $\boldsymbol{\tau}_i$ derives from the viscous stresses and is given by

$$\boldsymbol{\tau}_i = 2\mu \nabla^s \mathbf{u} \mathbf{1}_i + \lambda(\nabla \cdot \mathbf{u})\mathbf{1}_i, \tag{7}$$

where μ is the fluid viscosity, $\lambda = -2/3\mu$, and ∇^s is the symmetric spatial gradient, and q_i is the heat flux given by

$$q_i = -\kappa T_{,i}, \tag{8}$$

where κ is the fluid thermal conductivity. We assume that the fluid is an ideal gas with the equation of state given by

$$p = \rho^f R T, \tag{9}$$

where R is the specific gas constant. In the above formulas, i is the space dimension index, $(\cdot)_{,i}$ denotes a partial derivative with respect to spatial coordinates $\mathbf{x}$, and $(\cdot)_{,t}$ denotes a partial time derivative holding the spatial coordinates fixed. Einstein's summation convention is used throughout the manuscript.

2.2 Quasi-linear form of the compressible-flow equations

The Navier–Stokes equations of compressible flows may be written in a quasi-linear form using primitive variables based on pressure [33] as follows

$$\mathbf{A}_0^f \mathbf{Y}_{,t} + \mathbf{A}_i^a \mathbf{Y}_{,i} + \mathbf{F}_{i,i}^p - \mathbf{F}_{i,i}^d - \mathbf{S}^f = \mathbf{0}, \tag{10}$$

where $\mathbf{Y}$ denotes the set of pressure-primitive variables given by

$$\mathbf{Y} = \begin{pmatrix} p \\ \mathbf{u} \\ T \end{pmatrix}, \tag{11}$$

$\mathbf{A}_0^f$ is the Jacobian of the mapping $\mathbf{U}(\mathbf{Y})$ given by

$$\mathbf{A}_0^f = \frac{\partial \mathbf{U}}{\partial \mathbf{Y}}, \tag{12}$$

and $\mathbf{A}_i^a$ is the Jacobian of the mapping $\mathbf{F}_i^a(\mathbf{Y})$ given by

$$\mathbf{A}_i^a = \frac{\partial \mathbf{F}_i^a}{\partial \mathbf{Y}}. \tag{13}$$

We also define the Jacobian matrix that takes the pressure terms into account, namely,

$$\mathbf{A}_i^f = \mathbf{A}_i^a + \frac{\partial \mathbf{F}_i^p}{\partial \mathbf{Y}}. \tag{14}$$

The above Jacobian matrices may be analytically derived (see [33] for details). Here we provide an explicit expression for $\mathbf{A}_0^f$,

$$\mathbf{A}_0^f = \begin{pmatrix} \rho^f \beta_T & \mathbf{0}^T & -\rho^f \alpha_p \\ \rho^f \beta_T \mathbf{u} & \rho^f \mathbf{I} & -\rho^f \alpha_p \mathbf{u} \\ \rho^f \beta_T e_{tot} & \rho^f \mathbf{u}^T & \rho^f(-\alpha_p e_{tot} + c_v) \end{pmatrix}, \tag{15}$$

where $\mathbf{I}$ is a $d \times d$ identity matrix, $\beta_T = 1/p$ and $\alpha_p = 1/T$. Explicit expressions for $\mathbf{A}_i^a$'s and $\mathbf{A}_i^f$'s may be found in [33].

Remark The choice of primitive variables based on pressure to discretize the Navier–Stokes equations of compressible flows in this work is based on two factors: a. This variable set, or a subset thereof, is typically employed for the discretization of the equations of solid and structural mechanics, and thus presents a convenient variable choice for the discretization of the coupled FSI problem; b. The Jacobian matrices have a well defined incompressible limit, which presents a pathway to a unified formulation for both compressible and incompressible flows [68].

2.3 Updated Lagrangian formulation of an inelastic solid

We consider an inelastic solid, and state the point-wise balance of mass and momentum written in the updated Lagrangian form [69] as

$$\rho_0^s - \rho^s J = 0, \tag{16}$$

$$\rho^s \dot{\mathbf{u}} - \nabla \cdot \boldsymbol{\sigma} - \mathbf{s} = \mathbf{0}. \tag{17}$$

In Eqs. (16)–(17), $\mathbf{u}$, as before, is a material particle velocity, ρ^s and ρ_0^s are the solid densities in the current and reference configurations, respectively, $\boldsymbol{\sigma}$ is the Cauchy stress, $\mathbf{s}$ is the source term, $\dot{(\cdot)}$ is used to denote the material time derivative, that is, time derivative holding material coordinates $\mathbf{X}$ fixed, and $J = \det \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$.

Standard J_2 flow theory with isotropic hardening [70,71], which is suitable for metals, is considered in this work. The solid constitutive equations, written in the rate form at a material point, are summarized in what follows:

- *Additive decomposition of the rate-of-deformation tensor*

$$\mathbf{D} = \nabla^s \mathbf{u} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \mathbf{D}^e + \mathbf{D}^p, \tag{18}$$

  where $\mathbf{D}^e$ and $\mathbf{D}^p$ are its elastic and plastic components, respectively.
- *Stress-rate constitutive relation*

$$\boldsymbol{\sigma}^{\nabla J} = \mathbb{C} : \mathbf{D}^e = \mathbb{C} : (\mathbf{D} - \mathbf{D}^p), \tag{19}$$

  where $\mathbb{C}$ is the constitutive material tensor, and $\boldsymbol{\sigma}^{\nabla J}$ is the objective Jaumann rate of the Cauchy stress [69] given by

$$\boldsymbol{\sigma}^{\nabla J} = \dot{\boldsymbol{\sigma}} - \boldsymbol{\sigma}\boldsymbol{\omega}^T - \boldsymbol{\omega}\boldsymbol{\sigma}, \tag{20}$$

  where $\dot{\boldsymbol{\sigma}}$ is the material time derivative of the Cauchy stress, and

$$\boldsymbol{\omega} = \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^T) \tag{21}$$

  is the spin tensor.
- *Yield surface*

$$f(\boldsymbol{\sigma}, \bar{\epsilon}^p) = \bar{\sigma}(\boldsymbol{\sigma}) - \sigma_Y(\bar{\epsilon}_p) = 0, \tag{22}$$

  where $\bar{\sigma}$ is the equivalent or von Mises stress given by

$$\bar{\sigma}(\boldsymbol{\sigma}) = \sqrt{\frac{3}{2}\boldsymbol{\sigma}' : \boldsymbol{\sigma}'}, \tag{23}$$

$$\boldsymbol{\sigma}' = \boldsymbol{\sigma} - \frac{1}{3}(\mathrm{tr}\,\boldsymbol{\sigma})\,\mathbf{I}, \tag{24}$$

  and σ_Y is the yield stress assumed dependent on the equivalent plastic strain $\bar{\epsilon}_p$.
- *Flow rule*

$$\mathbf{D}^p = \dot{\bar{\epsilon}}_p \frac{\partial f}{\partial \boldsymbol{\sigma}}, \tag{25}$$

  which corresponds to associative plasticity [70,71] and states that the plastic flow occurs in the direction orthogonal to the yield surface. For the von Mises yield criterion the partial derivative in the associative flow rule equation may be computed explicitly and becomes

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\frac{\boldsymbol{\sigma}'}{\|\boldsymbol{\sigma}'\|}. \tag{26}$$

- *Consistency condition*

$$\dot{f} = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} - \frac{\partial f}{\partial \bar{\epsilon}_p}\dot{\bar{\epsilon}}_p = 0, \tag{27}$$

  which, in combination with the associative flow rule, leads to the following expression for the equivalent plastic-strain rate:

$$\dot{\bar{\epsilon}}_p = \frac{\frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{D}}{H + \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \frac{\partial f}{\partial \boldsymbol{\sigma}}}. \tag{28}$$

Remark Note that the key quantity appearing in the above constitutive model is $\nabla \mathbf{u}$, the spatial velocity gradient. It is the time history of this quantity that "drives" the evolution of the solid Cauchy stress.

Remark In the present work we consider the solid as isothermal. However, introducing thermal coupling by making the Cauchy stress temperature-dependent and adding the energy-balance equation to the system given by Eqs. (16)–(17) does not present a conceptual difficulty and will be pursued in the future work.

2.4 Quasi-linear form of the inelastic solid equations

Solid linear-momentum balance given by Eq. (17) may be written in a quasi-linear form consistent with that of the Navier–Stokes equations of compressible flows as

$$\mathbf{A}_0^s \dot{\mathbf{Y}} - \mathbf{F}_{i,i}^{\sigma} - \mathbf{S}^s = \mathbf{0}, \tag{29}$$

where $\mathbf{Y}$ is the set of primitive variables from Eq. (11),

$$\mathbf{A}_0^s = \begin{pmatrix} 0 & \mathbf{0}^T & 0 \\ \mathbf{0} & \rho^s \mathbf{I} & \mathbf{0} \\ 0 & \mathbf{0}^T & 0 \end{pmatrix}, \tag{30}$$

$$\mathbf{F}_i^{\sigma} = \begin{pmatrix} 0 \\ \boldsymbol{\sigma}\mathbf{1}_i \\ 0 \end{pmatrix}, \tag{31}$$

and

$$\mathbf{S}^s = \begin{pmatrix} 0 \\ \mathbf{s} \\ 0 \end{pmatrix}. \tag{32}$$

It is convenient to express the material time derivative in Eq. (29) using its spatial counterpart and a convection term as

$$\dot{\mathbf{Y}} = \mathbf{Y}_{,t} + u_i \mathbf{Y}_{,i}, \tag{33}$$

in which case the quasi-linear form of Eq. (17) may be written as

$$\mathbf{A}_0^s \mathbf{Y}_{,t} + \mathbf{A}_i^s \mathbf{Y}_{,i} - \mathbf{F}_{i,i}^{\sigma} - \mathbf{S}^s = \mathbf{0}, \tag{34}$$

where

$$\mathbf{A}_i^s = \begin{pmatrix} 0 & \mathbf{0}^T & 0 \\ \mathbf{0} & \rho^s u_i \mathbf{I} & \mathbf{0} \\ 0 & \mathbf{0}^T & 0 \end{pmatrix}. \tag{35}$$

Note that, in this formulation, because the time derivatives in the fluid and solid mechanics governing equations take on the same meaning, one can naturally define a time integration scheme that consistently accounts for the fluid and solid parts of the the coupled FSI problem.

2.5 Weak form of the coupled FSI problem

Let Ω denote the combined fluid and solid domain, and let Ω^f and Ω^s denote the individual, time-dependent fluid and solid subdomains in the spatial configuration, such that $\Omega^f \bigcup \Omega^s = \Omega$ and $\Omega^f \bigcap \Omega^s = \emptyset$. Let Γ^{fs} denote their interface, which also evolves in time. We define the following semilinear forms and linear functionals corresponding to the weak forms of the fluid and solid subproblems:

$$M_{\omega}^f(\mathbf{W}, \mathbf{Y}) = \int_{\omega} \mathbf{W} \cdot \mathbf{A}_0^f \mathbf{Y}_{,t} \, d\omega, \tag{36}$$

$$B_{\omega}^f(\mathbf{W}, \mathbf{Y}) = \int_{\omega} \mathbf{W} \cdot \mathbf{A}_i^a \mathbf{Y}_{,i} \, d\omega - \int_{\omega} \mathbf{W}_{,i} \cdot \left(\mathbf{F}_i^p - \mathbf{F}_i^d \right) \, d\omega, \tag{37}$$

$$F_{\omega}^f(\mathbf{W}) = \int_{\omega} \mathbf{W} \cdot \mathbf{S}^f \, d\omega + \int_{\Gamma_H^f} \mathbf{W} \cdot \mathbf{H}^f \, d\Gamma, \tag{38}$$

$$M_{\omega}^s(\mathbf{W}, \mathbf{Y}) = \int_{\omega} \mathbf{W} \cdot \mathbf{A}_0^s \mathbf{Y}_{,t} \, d\omega, \tag{39}$$

$$B_{\omega}^s(\mathbf{W}, \mathbf{Y}) = \int_{\omega} \mathbf{W} \cdot \mathbf{A}_i^s \mathbf{Y}_{,i} \, d\omega + \int_{\omega} \mathbf{W}_{,i} \cdot \mathbf{F}_i^{\sigma} \, d\omega, \tag{40}$$

$$F_{\omega}^s(\mathbf{W}) = \int_{\omega} \mathbf{W} \cdot \mathbf{S}^s \, d\omega + \int_{\Gamma_H^s} \mathbf{W} \cdot \mathbf{H}^s \, d\Gamma, \tag{41}$$

where $\mathbf{Y}$ and $\mathbf{W}$, the vector-valued trial and test functions, respectively, are the members of $\mathcal{S}$ and $\mathcal{V}$, the corresponding trial and test function spaces, respectively, *defined on all of* Ω, Γ_H^f and Γ_H^s are the subsets of the fluid- and solid-domain boundaries where natural boundary conditions are imposed, and $\mathbf{H}^f$ and $\mathbf{H}^s$ contain the prescribed values of the natural boundary conditions. Note that in Eqs. (36)–(41) the subscript ω on the semilinear forms and linear functionals denotes the domain of integration.

With the above definitions, the coupled FSI problem maybe stated as: Find $\mathbf{Y} \in \mathcal{S}$, such that $\forall \mathbf{W} \in \mathcal{V}$,

$$\begin{aligned} &M_{\Omega^f}^f(\mathbf{W}, \mathbf{Y}) + B_{\Omega^f}^f(\mathbf{W}, \mathbf{Y}) - F_{\Omega^f}^f(\mathbf{W}) \\ &\qquad + \\ &M_{\Omega^s}^s(\mathbf{W}, \mathbf{Y}) + B_{\Omega^s}^s(\mathbf{W}, \mathbf{Y}) - F_{\Omega^s}^s(\mathbf{W}) \\ &\qquad = \\ &\qquad 0, \end{aligned} \tag{42}$$

where we assume that functions in $\mathcal{S}$ and $\mathcal{V}$ have sufficient regularity for the coupled FSI problem given by Eq. (42) to be well-posed, and are continuous across Γ^{fs}.

Examination of the Euler–Lagrange conditions for the above coupled problem reveal that the fluid and solid governing equations hold on the interior of their respective domains, namely,

$$\mathbf{R}^f(\mathbf{Y}) = \mathbf{A}_0^f \mathbf{Y}_{,t} + \mathbf{A}_i^a \mathbf{Y}_{,i} + \mathbf{F}_{i,i}^p - \mathbf{F}_{i,i}^d = \mathbf{0} \quad \text{in } \Omega^f, \tag{43}$$

and

$$\mathbf{R}^s(\mathbf{Y}) = \mathbf{A}_0^s \mathbf{Y}_{,t} + \mathbf{A}_i^s \mathbf{Y}_{,i} - \mathbf{F}_{i,i}^{\sigma} - \mathbf{S}^s = \mathbf{0} \quad \text{in } \Omega^s, \tag{44}$$

and the natural boundary conditions hold on their respective boundaries, namely,

$$-\mathbf{F}_i^p n_i^f + \mathbf{F}_i^d n_i^f - \mathbf{H}^f = \mathbf{0} \quad \text{on } \Gamma_H^f, \tag{45}$$

and

$$\mathbf{F}_i^{\sigma} n_i^s - \mathbf{H}^s = \mathbf{0} \quad \text{on } \Gamma_H^s, \tag{46}$$

where n_i^f and n_i^s are the cartesian components of the outward unit normal vector to the fluid and solid domains, respectively, in the current configuration. In addition, at the fluid–solid interface, the following compatibility condition holds,

$$-\mathbf{F}_i^p n_i^f + \mathbf{F}_i^d n_i^f + \mathbf{F}_i^{\sigma} n_i^s = \mathbf{0} \quad \text{on } \Gamma^{fs}, \tag{47}$$

which is a consequence of the test-function continuity at the fluid–solid interface. Equation (47) implies that the fluid and solid tractions are in equilibrium at the fluid–solid interface, namely,

$$-p n_i^f + \tau_{ij} n_j^f + \sigma_{ij} n_j^s = 0 \quad \text{on } \Gamma^{fs}. \tag{48}$$

In addition, the energy-equation component of Eq. (47) yields the following condition at the fluid–solid interface

$$-p u_i n_i^f + u_i \tau_{ij} n_j^f - q_i n_i^f = 0 \quad \text{on } \Gamma^{fs}, \tag{49}$$

which is a consequence of not considering thermal coupling in the solid mechanics formulation. In case thermal coupling is added to the solid mechanics formulation, Eq. (47) would naturally lead to the equilibrium of heat fluxes at the fluid–solid interface.

Remark Using the additive property of integrals, the coupled FSI formulation given by Eq. (42) may be re-written as: Find $\mathbf{Y} \in \mathcal{S}$, such that $\forall \mathbf{W} \in \mathcal{V}$,

$$\begin{aligned} & M^f_\Omega(\mathbf{W},\mathbf{Y}) + B^f_\Omega(\mathbf{W},\mathbf{Y}) - F^f_\Omega(\mathbf{W}) \\ & \quad + \\ & M^s_{\Omega^s}(\mathbf{W},\mathbf{Y}) + B^s_{\Omega^s}(\mathbf{W},\mathbf{Y}) - F^s_{\Omega^s}(\mathbf{W}) \\ & \quad - \\ & M^f_{\Omega^s}(\mathbf{W},\mathbf{Y}) + B^f_{\Omega^s}(\mathbf{W},\mathbf{Y}) - F^f_{\Omega^s}(\mathbf{W}) \\ & \quad = \\ & \quad 0, \end{aligned} \tag{50}$$

where the integration over the fluid mechanics domain is replaced by integration over the combined domain *minus* that over the solid domain. This form of the coupled problem, which at the continuous level is equivalent to the original formulation given by Eq. (42), is convenient for the application of an immersed approach to the discretization of the coupled FSI equations (see, e.g., [72]).

3 Discrete formulation and algorithmic aspects

Here we present the discretization of the continuous FSI formulation developed in the previous section. We briefly cover a stabilized formulation with discontinuity capturing employed for the compressible-flow equations, state the semi-discrete immersed FSI formulation, and present time discretization of the coupled FSI problem, including the stress update algorithm.

3.1 Stabilization of the compressible-flow equations

To discretize the compressible-flow equations we make use of the SUPG formulation [35–39] augmented with a discontinuity-capturing operator [40–46]. The latter is important to ensure stability for high-Mach-number flows, which is the case in the present work. We briefly summarize the SUPG and discontinuity-capturing operators in what follows. The reader is referred to [73] for the details of the compressible-flow formulation employed in the present work.

The SUPG stabilization operator for compressible flows may be expressed by means of the following semilinear form:

$$B^{st}_\omega(\mathbf{W},\mathbf{Y}) = \int_{\tilde{\omega}} \left(\mathbf{A}^f_i\right)^T \mathbf{W}_{,i} \cdot \boldsymbol{\tau}\mathbf{R}^f(\mathbf{Y})\, d\tilde{\omega}, \tag{51}$$

where it is assumed that ω is discretized into elements, $\tilde{\omega}$ is a collection of element interiors, the integral $\int_{\tilde{\omega}}$ is taken element-wise, and $\boldsymbol{\tau}$ is a $(d+2) \times (d+2)$ stabilization matrix. We make use of the following definition of $\boldsymbol{\tau}$,

$$\boldsymbol{\tau} = \left(\mathbf{A}^f_0\right)^{-1} \hat{\boldsymbol{\tau}}, \tag{52}$$

where $\hat{\boldsymbol{\tau}}$ is a stabilization matrix defined for conservation variables (see, e.g., [33,34]). Premultiplication of $\hat{\boldsymbol{\tau}}$ by $(\mathbf{A}^f_0)^{-1}$ gives an appropriate transformation of the stabilization matrix between the two variable sets.

The discontinuity-capturing operator is also designed for conservation variables with a transformation to the pressure-primitive variables, leading to the following definition,

$$B^{dc}_\omega(\mathbf{W},\mathbf{Y}) = \int_{\tilde{\omega}} \mathbf{W}_{,i} \cdot \hat{\boldsymbol{\nu}}^{dc} \mathbf{A}^f_0 \mathbf{Y}_{,i}\, d\tilde{\omega}, \tag{53}$$

where $\hat{\boldsymbol{\nu}}^{dc}$ is a diagonal $(d+2) \times (d+2)$ matrix of shock-capturing parameters defined for conservation variables (see e.g., [42–46]) and $\mathbf{A}^f_0$ gives the transformation to the primitive-variable formulation. The design of $\hat{\boldsymbol{\nu}}^{dc}$ makes use of the compressible-flow-equation residuals, which renders the discontinuity-capturing operator consistent.

Remark In compressible-flow computations in the blast regime one often makes use of the classical von Neumann–Richtmyer artificial viscosities (see, e.g., [74,75].) No such viscosities are employed in the present formulation, and the numerical results presented later in the article indicate that shock-capturing given by Eq. (53) is sufficient to stabilize the formulation in the high-Mach-number regime of blast waves.

3.2 Coupled FSI formulation at the semi-discrete level

Taking the coupled FSI formulation at the continuous level given by Eq. (50) as a starting point, and using the SUPG and discontinuity-capturing operators to stabilize the compressible-flow equations, the semi-discrete immersed FSI formulation may be stated as: Find $\mathbf{Y}^h \in \mathcal{S}^h$, such that $\forall \mathbf{W}^h \in \mathcal{V}^h$,

$$\begin{aligned} & M^f_\Omega(\mathbf{W}^h,\mathbf{Y}^h) + B^f_\Omega(\mathbf{W}^h,\mathbf{Y}^h) - F^f_\Omega(\mathbf{W}^h) \\ & \qquad + B^{st}_\Omega(\mathbf{W}^h,\mathbf{Y}^h) + B^{dc}_\Omega(\mathbf{W}^h,\mathbf{Y}^h) \\ & \qquad + \\ & M^s_{\Omega^s}(\mathbf{W}^h,\mathbf{Y}^h) + B^s_{\Omega^s}(\mathbf{W}^h,\mathbf{Y}^h) - F^s_{\Omega^s}(\mathbf{W}^h) \\ & \qquad - \\ & M^f_{\Omega^s}(\mathbf{W}^h,\mathbf{Y}^h) + B^f_{\Omega^s}(\mathbf{W}^h,\mathbf{Y}^h) - F^f_{\Omega^s}(\mathbf{W}^h) \\ & \qquad + B^{st}_{\Omega^s}(\mathbf{W}^h,\mathbf{Y}^h) + B^{dc}_{\Omega^s}(\mathbf{W}^h,\mathbf{Y}^h) \\ & \qquad + \end{aligned} \tag{54}$$

$$\int_{\tilde{\Gamma}^f} \beta h \, [\![\mathbf{w}^h_{,i} n_i]\!] \cdot [\![\mathbf{u}^h_{,j} n_j]\!] \, d\tilde{\Gamma} = 0.$$

Here, $\mathbf{Y}^h$ and $\mathbf{W}^h$, the discrete trial and test functions, respectively, and $\mathcal{S}^h$ and $\mathcal{V}^h$, the corresponding discrete function spaces, are defined *on the background domain* Ω. As a result, the unknown degrees of freedom (DOFs) are defined completely on the background mesh. Equal-order discretization is employed for all the unknowns in the variable set $\mathbf{Y}^h$.

In Eq. (54) the integrals are computed using numerical quadrature. The terms on the first and fourth lines on the left-hand-side of Eq. (54) are computed using quadrature rules defined on the background domain, while the terms on the second and third lines are computed using foreground-domain quadrature rules. To carry out integration on the foreground mesh, the background mesh quantities need to be evaluated at the locations corresponding to the quadrature points of the foreground mesh. These locations are found through a simple inverse mapping.

Remark It is well known that solid near incompressibility may lead to volumetric locking, which is especially pronounced for lower-order elements. To alleviate this phenomenon, we adapt a *B-bar* methodology [76–78] to the solid mechanics part of the present immersed FSI formulation. In the evaluation of the velocity and test-function gradients, the strain displacement matrix *from the background discretization* is replaced with its *B-bar* counterpart in a standard fashion, wherein the dilatational part of the motion is projected to a lower-order space (see, e.g., [76,79]). In the present work, the *B-bar* methodology is only employed for C^0-continuous linear FEM background discretizations.

The last term on the left-hand-side of Eq. (54) is the so-called "ghost velocity" stabilization developed for immersed FEM in [80]. In this term $\mathbf{w}$ is used to denote the linear-momentum components of the test function $\mathbf{W}$, $[\![\cdot]\!]$ is the "jump" operator, $\tilde{\Gamma}^f$ is the set of all edges in 2D and faces in 3D near the interface, h is the size of the local edge or face, and β is the mesh-independent penalty parameter. Ghost velocity stabilization penalizes the jump in the velocity gradient near the fluid–solid interface and significantly improves the stability of the velocity solution.

Remark The edge or face over which the ghost velocity stabilization is applied is such that both elements sharing that edge or face are either covered by the structure or cut by the interface. This condition excludes edges or faces that are on the exterior of the cut-element set.

Remark For a thermally coupled solid, an analogous stabilization technique may be employed also for the temperature variable.

Remark Note that the ghost velocity stabilization terms are active only in the case when a C^0-continuous background discretization is employed. When smooth splines are used, this term vanishes due to the basis-function derivative continuity across element boundaries. Although stability of spline discretizations may be further improved by penalizing higher-order velocity and temperature derivatives at cut-element edges or faces, this is not done in the present work.

3.3 Time discretization

An *explicit* version of the Generalized-α method [81–83] is employed for the time discretization of Eq. (54). In this case the discrete residual $\boldsymbol{\mathcal{R}}$ is thought of as a function of the background nodal or control-point unknown vector $\boldsymbol{\mathcal{Y}}$ and its time derivative $\dot{\boldsymbol{\mathcal{Y}}}$, namely,

$$\boldsymbol{\mathcal{R}}(\dot{\boldsymbol{\mathcal{Y}}}, \boldsymbol{\mathcal{Y}}) = \boldsymbol{\mathcal{M}}\dot{\boldsymbol{\mathcal{Y}}} + \boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{Y}}). \tag{55}$$

In Eq. (55), to set the stage for an explicit time integration algorithm, the discrete residual is decomposed into the parts containing $\boldsymbol{\mathcal{Y}}$ and $\dot{\boldsymbol{\mathcal{Y}}}$. Also in Eq. (55),

$$[\boldsymbol{\mathcal{M}}]^{ab}_{AB} = \int_{\Omega} N_A \left[\mathbf{A}^f_0\right]_{ab} N_B \, d\Omega + \int_{\Omega^s} N_A \left[\mathbf{A}^s_0\right]_{ab} N_B \, d\Omega - \int_{\Omega^s} N_A \left[\mathbf{A}^f_0\right]_{ab} N_B \, d\Omega, \tag{56}$$

are the components of the coupled mass matrix, A, B are the nodal or control-point indices, a, b are the local DOF indices, and N's are the basis function coming from the background discretization. The residual vector $\boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{Y}})$ is comprised of the remaining terms in the coupled FSI formulation given by Eq. (54), and its components may be expressed as

$$\begin{aligned}[\boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{Y}})]^a_A =& B^f_{\Omega}\left(N_A\mathbf{E}_a, \mathbf{Y}^h\right) - F^f_{\Omega}(N_A\mathbf{E}_a) \\ &+ B^{st}_{\Omega}\left(N_A\mathbf{E}_a, \mathbf{Y}^h\right) + B^{dc}_{\Omega}(N_A\mathbf{E}_a, \mathbf{Y}^h) \\ &+ B^s_{\Omega^s}\left(N_A\mathbf{E}_a, \mathbf{Y}^h\right) - F^s_{\Omega^s}(N_A\mathbf{E}_a) \\ &- B^f_{\Omega^s}\left(N_A\mathbf{E}_a, \mathbf{Y}^h\right) - F^f_{\Omega^s}(N_A\mathbf{E}_a)\end{aligned} \tag{57}$$

$$+ B^{st}_{\Omega^s}\left(N_A\mathbf{E}_a, \mathbf{Y}^h\right) + B^{dc}_{\Omega^s}\left(N_A\mathbf{E}_a, \mathbf{Y}^h\right) + \int_{\tilde{\Gamma}^f} \beta h \left[\!\left[N_{A,i} n_i \mathbf{1}_{b(a)}\right]\!\right] \cdot \left[\!\left[\mathbf{u}^h_{,j} n_j\right]\!\right] d\tilde{\Gamma},$$

where $\mathbf{E}_a$ is the ath Cartesian basis vector in $\mathbb{R}^{d+2}$, and $\mathbf{1}_b$, as before, is the bth Cartesian basis vector in $\mathbb{R}^d$, with $b(a) = a - 1$ and $\mathbf{1}_0 = \mathbf{1}_{d+1} = \mathbf{0}$.

The generalized-α technique applied to Eq. (55) amounts to collocating the discrete residual at the intermediate locations within a time step as follows: Given $\mathcal{Y}_n$ and $\dot{\mathcal{Y}}_n$, find $\mathcal{Y}_{n+1}$ and $\dot{\mathcal{Y}}_{n+1}$, such that,

$$\mathcal{R}\left(\dot{\mathcal{Y}}_{n+\alpha_m}, \mathcal{Y}_{n+\alpha_f}\right) = \mathbf{0}, \tag{58}$$

where the intermediate solution time levels are defined as

$$(\cdot)_{n+\alpha} = (\cdot)_n + \alpha((\cdot)_{n+1} - (\cdot)_n), \tag{59}$$

and the relationship between the solution and its time derivative in the time-discrete setting is given by the Newmark formula

$$\mathcal{Y}_{n+1} = \mathcal{Y}_n + \Delta t \left((1-\gamma)\dot{\mathcal{Y}}_n + \gamma \dot{\mathcal{Y}}_{n+1}\right). \tag{60}$$

Here, α_m, α_f, and γ are real-valued parameters chosen based on the second-order accuracy and unconditional stability requirements of the Generalized-α method. See [81–83] for more details.

3.3.1 Predictor–multicorrector algorithm

To solve the nonlinear system given by Eqs. (58)–(60) we adopt an *explicit* version of a two-stage predictor-multicorrector algorithm presented in what follows.

Predictor stage Given the solution at time level t_n, initialize the time-level t_{n+1} solution as

$$\begin{aligned}\dot{\mathcal{Y}}^0_{n+1} &= \frac{\gamma - 1}{\gamma}\dot{\mathcal{Y}}_n,\\ \mathcal{Y}^0_{n+1} &= \mathcal{Y}_n.\end{aligned} \tag{61}$$

In addition, the foreground solid domain position at time level t_{n+1} is initialized as follows:

$$\begin{aligned}\mathbf{a}^0_{n+1} &= \frac{\gamma - 1}{\gamma}\mathbf{a}_n,\\ \mathbf{u}^0_{n+1} &= \mathbf{u}_n,\\ \mathbf{d}^0_{n+1} &= \mathbf{d}_n + \Delta t \mathbf{u}_n + \frac{\Delta t^2}{2}\left((1-2\beta)\mathbf{a}_n + 2\beta \mathbf{a}^0_{n+1}\right),\end{aligned} \tag{62}$$

where $\mathbf{a}$, $\mathbf{u}$, and $\mathbf{d}$ denote the solid nodal or control-point values of the acceleration, velocity and displacement, and β is the additional Newmark parameter of the Generalized-α scheme. The displacement variable $\mathbf{d}$ is used to place the foreground solid mesh in the appropriate configuration for the purposes of numerical integration.

Multicorrector stage Compute the solution at time level t_{n+1} by repeating the following steps:

1. Evaluate the iterates at intermediate time levels on the background mesh,

$$\begin{aligned}\dot{\mathcal{Y}}^l_{n+\alpha_m} &= \dot{\mathcal{Y}}_n + \alpha_m\left(\dot{\mathcal{Y}}^l_{n+1} - \dot{\mathcal{Y}}_n\right),\\ \mathcal{Y}^l_{n+\alpha_f} &= \mathcal{Y}_n + \alpha_f\left(\mathcal{Y}^l_{n+1} - \mathcal{Y}_n\right),\end{aligned} \tag{63}$$

   and on the foreground mesh,

$$\mathbf{d}^l_{n+\alpha_f} = \mathbf{d}_n + \alpha_f(\mathbf{d}^l_{n+1} - \mathbf{d}_n), \tag{64}$$

   where l is the multicorrector iteration counter.
2. Use the intermediate solution values to assemble the discrete residual of the coupled FSI problem and solve for the increment of the solution time derivative,

$$\Delta\dot{\mathcal{Y}}^l_{n+1} = -(\alpha_m \mathcal{M})^{-1}\,\mathcal{R}\left(\dot{\mathcal{Y}}^l_{n+\alpha_m}, \mathcal{Y}^l_{n+\alpha_f}\right). \tag{65}$$

   In the above equation, in the interest of efficiency, $\mathcal{M}$ may approximated by its lumped counterpart given by

$$\begin{aligned}\mathcal{M} &\approx \mathcal{M}_L,\\ [\mathcal{M}_L]^{ab}_{AB} &= \left(\sum_C [\mathcal{M}]^{ab}_{BC}\right)\delta_{AB},\end{aligned} \tag{66}$$

   where δ_{AB} is the Kronecker delta.
3. Update the solution on the background mesh,

$$\begin{aligned}\dot{\mathcal{Y}}^{l+1}_{n+1} &= \dot{\mathcal{Y}}^l_{n+1} + \Delta\dot{\mathcal{Y}}^l_{n+1},\\ \mathcal{Y}^{l+1}_{n+1} &= \mathcal{Y}^l_{n+1} + \gamma\Delta t\Delta\dot{\mathcal{Y}}^l_{n+1},\end{aligned} \tag{67}$$

   and on the foreground mesh,

$$\begin{aligned}\mathbf{u}^{l+1}_{n+1} &= \mathcal{I}\mathcal{Y}^{l+1}_{n+1},\\ \mathbf{a}^{l+1}_{n+1} &= \frac{\mathbf{u}^{l+1}_{n+1} - \mathbf{u}_n}{\gamma\Delta t} - \frac{1-\gamma}{\gamma}\mathbf{a}_n,\\ \mathbf{d}^{l+1}_{n+1} &= \mathbf{d}_n + \Delta t\mathbf{u}_n + \frac{\Delta t^2}{2}\left((1-2\beta)\mathbf{a}_n + 2\beta\mathbf{a}^{l+1}_{n+1}\right),\end{aligned} \tag{68}$$

   and increase the Newton-iteration counter l by one. Note that, just like in the predictor stage, the update of the

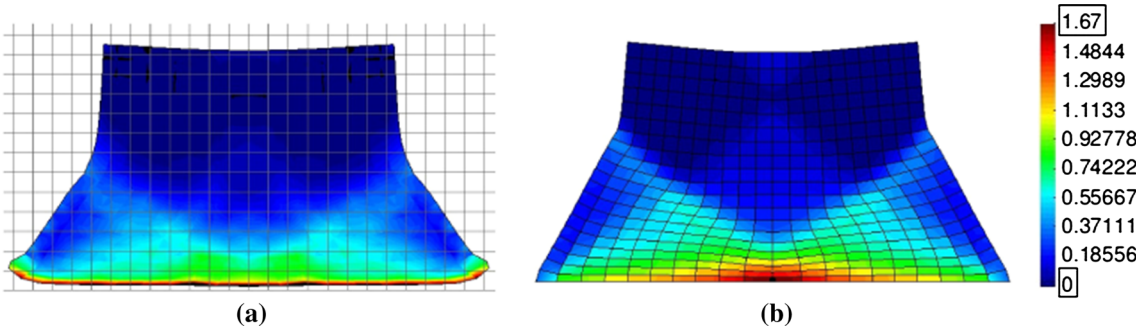


Fig. 1 Square bar impacting a rigid wall. Equivalent plastic strain for the immersed (**a**) and Lagrangian (**b**) simulations

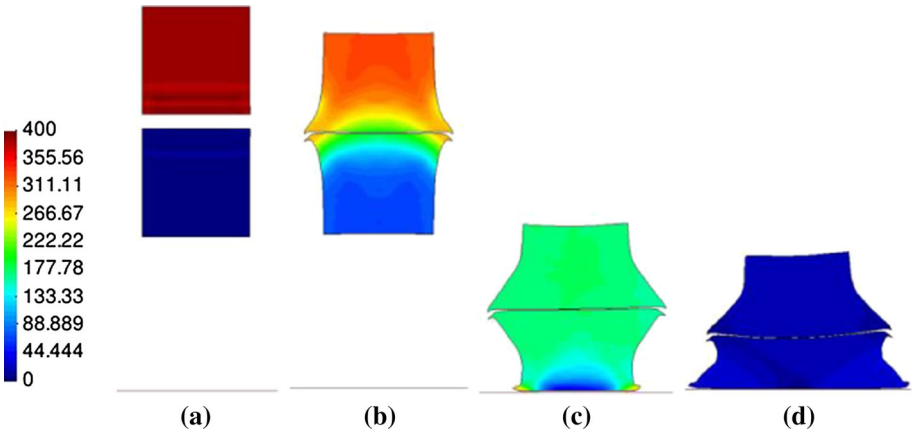


Fig. 2 Moving bar contacting stationary bar and impacting a rigid wall. A temporal sequence showing the bar current configuration and velocity magnitude. **a** 0.0 ms, **b** 0.2 ms, **c** 1.8 ms and **d** 2.4 ms

foreground mesh kinematics entails the velocity projection from the background mesh, denoted by the symbol $\mathcal{I}$ above, followed by the reconstruction of the remaining kinematic quantities via the Newmark formulas.

Remark Computation of the solution-time-derivative increment in Eq. (65) using a lumped mass matrix requires computing the inverse of a $(d+2)\times(d+2)$ matrix at each background-mesh node or control point. It is possible to develop a vector representation of the lumped-mass matrix of the form

$$[\mathcal{M}_L]_A^{ab} = [\overline{\mathbf{A}}_0]^{ab} \int_\Omega N_A \, d\Omega, \tag{69}$$

where $\overline{\mathbf{A}}_0$ is the equivalent Jacobian matrix associated with node or control point A. The matrix $\overline{\mathbf{A}}_0$ has an inverse, which may be derived analytically (see [84]) and programmed directly, leading to modest improvements in the computational efficiency of the time-integration algorithm.

Remark In the last step of the multicorrector algorithm, once the solution is obtained on the background mesh, kinematic quantities are transferred to the foreground mesh and the solid is moved to a new position. This procedure is also employed in the particle finite element method (PFEM) [85,86].

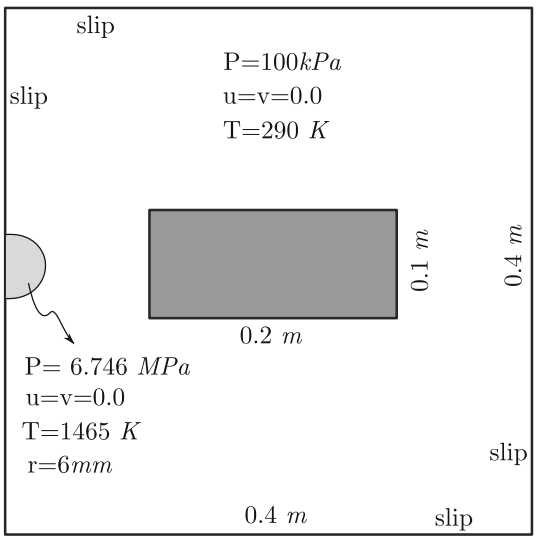


Fig. 3 Chamber detonation. Problem geometry and setup

3.3.2 Stress update

Computation of the discrete residual In Eq. (65) requires evaluation of the solid Cauchy stress at the quadrature points of the foreground mesh, which act as material points in the current formulation. The Cauchy stress may be advanced within a time step as follows,

$$\begin{aligned}\boldsymbol{\sigma}_{n+1} &= \boldsymbol{\sigma}_n + \Delta t\, \dot{\boldsymbol{\sigma}}_{n+\alpha_f} \\ &= \boldsymbol{\sigma}_n + \Delta t\, (\boldsymbol{\sigma}^{\nabla J} + \boldsymbol{\omega}\boldsymbol{\sigma} + \boldsymbol{\sigma}\boldsymbol{\omega}^T)_{n+\alpha_f},\end{aligned} \tag{70}$$

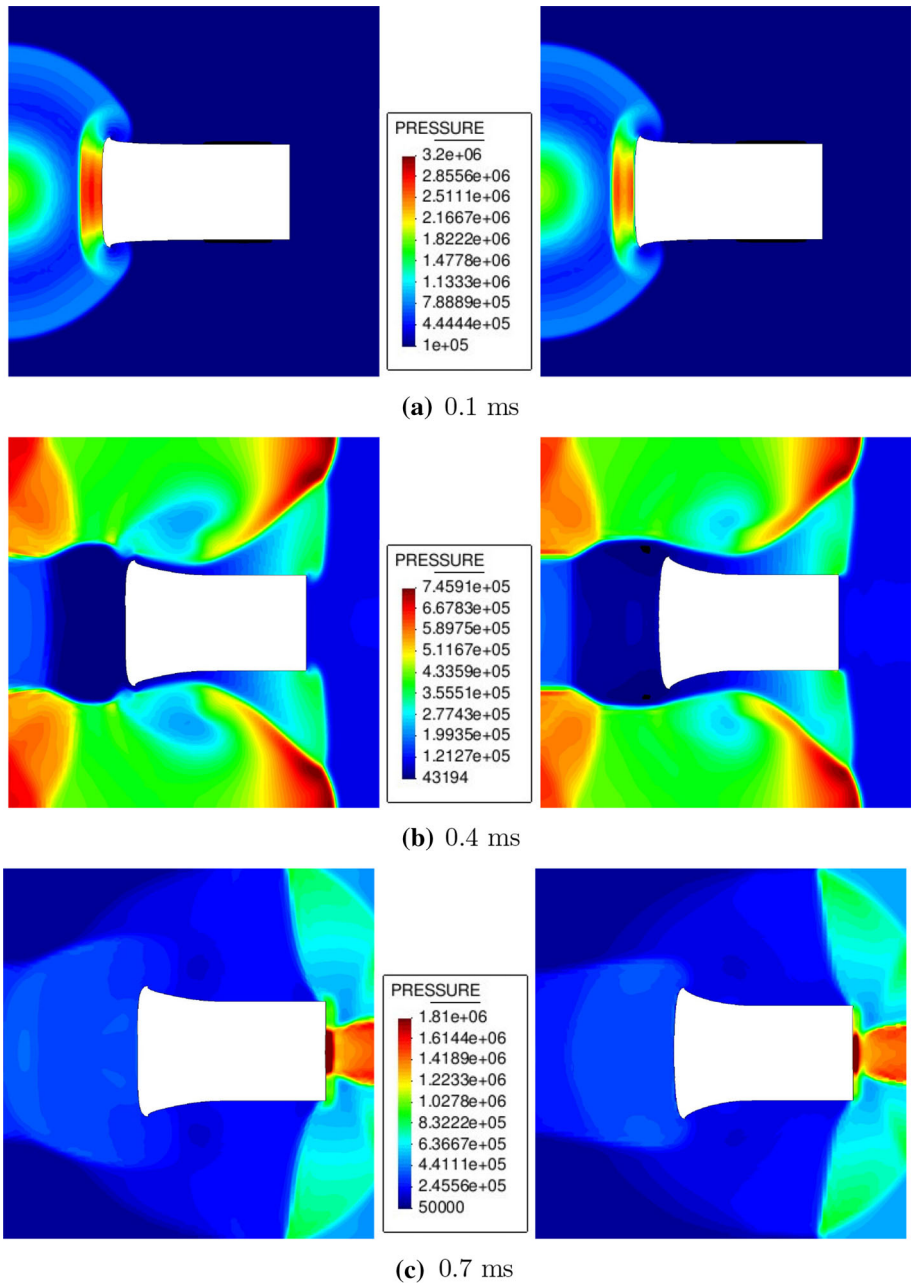
where the objective Jaumann stress rate is employed, and, for consistency with the Generalized-α time integration algorithm, the material time derivative of the stress is taken at the time level $t_{n+\alpha_f}$. To carry out the above stress update, at each multicorrector iteration of the Generalized-α algorithm, we repeat following steps:

1. Rotate the Cauchy stress to the time level $t_{n+\alpha_f}$ as

$$\tilde{\boldsymbol{\sigma}}^l_{n+\alpha_f} = \boldsymbol{\sigma}_n + \alpha_f \Delta t \left(\boldsymbol{\omega}^l_{n+\alpha_f} \boldsymbol{\sigma}_n + \boldsymbol{\sigma}_n \left(\boldsymbol{\omega}^l_{n+\alpha_f} \right)^T \right), \tag{71}$$

where l is the iteration counter of the multicorrector stage, and the spin tensor at time level $t_{n+\alpha_f}$ is computed from the background discretization as

$$\boldsymbol{\omega}^l_{n+\alpha_f} = \frac{1}{2} \left(\nabla \mathbf{u}^l_{n+\alpha_f} - \left(\nabla \mathbf{u}^l_{n+\alpha_f} \right)^T \right). \tag{72}$$



(a) 0.1 ms

(b) 0.4 ms

(c) 0.7 ms

Fig. 4 Chamber detonation. Pressure at different time instants. *Left* ALE simulation; *Right* immersed simulation. **a** 0.1 ms, **b** 0.4 ms and **c** 0.7 ms

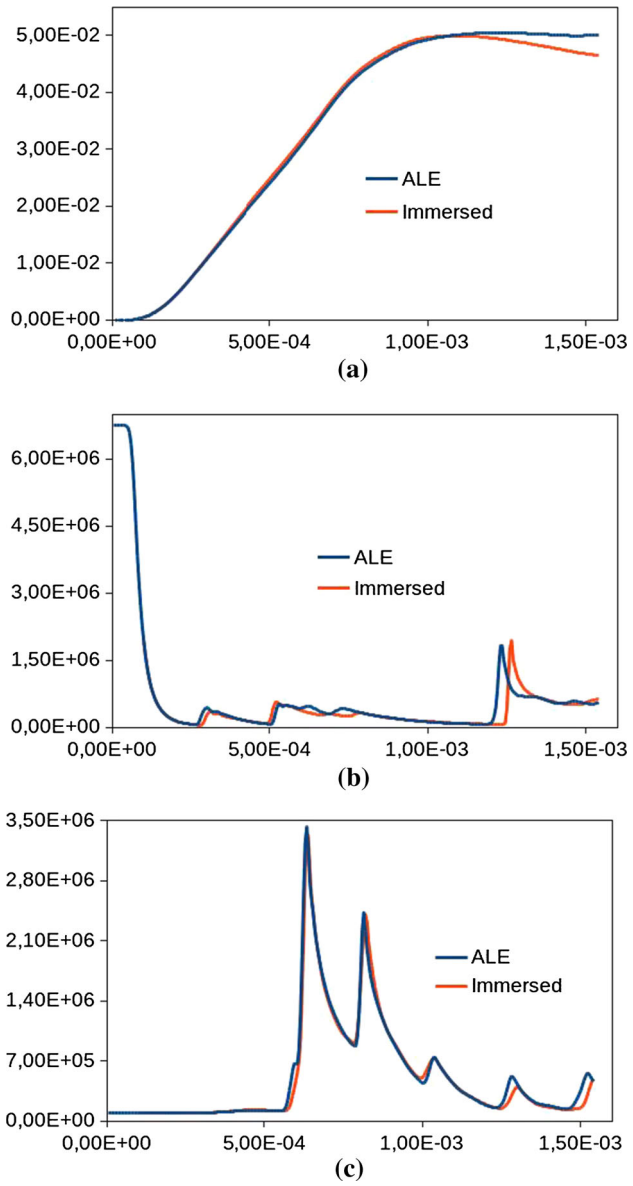


Fig. 5 Chamber detonation. Comparison of ALE and immersed simulation results. **a** Horizontal displacement (in m) of the bar center of mass; **b** pressure (in Pa) at the center of detonation; **c** pressure (in Pa) at the center of the right wall. All quantities are plotted versus time (in s)

2. Update the Cauchy stress at time level $t_{n+\alpha_f}$ as

$$\boldsymbol{\sigma}^l_{n+\alpha_f} = \tilde{\boldsymbol{\sigma}}^l_{n+\alpha_f} + \Delta\boldsymbol{\sigma}^l_{n+\alpha_f}, \tag{73}$$

where $\Delta\boldsymbol{\sigma}_{n+\alpha_f} = \Delta t\, \boldsymbol{\sigma}^{\nabla_J}_{n+\alpha_f}$ is the stress increment coming form the radial return mapping scheme [69].

3. Rotate the Cauchy stress to the time level t_{n+1} as

$$\boldsymbol{\sigma}^l_{n+1} = \boldsymbol{\sigma}^l_{n+\alpha_f} + (1-\alpha_f)\, \Delta t \left(\boldsymbol{\omega}^l_{n+\alpha_f}\boldsymbol{\sigma}^l_{n+\alpha_f} + \boldsymbol{\sigma}^l_{n+\alpha_f}\left(\boldsymbol{\omega}^l_{n+\alpha_f}\right)^T\right). \tag{74}$$

Remark The above stress update is a modification of the well-known half-step rotation technique [87,88]. The resulting Cauchy stress in Step 2 is used in the computation of the solid contribution to the discrete residual in Eq. (65), while the stress in Step 3 is saved for the purposes of performing the stress update in the next time step.

4 Numerical examples

We present several numerical examples in 2D using piecewise-linear quadrilateral FEM discretization for both background and foreground domains to illustrate that the proposed methodology works well when discretized with standard FEM. Computational results are compared with the conforming-mesh in-house Lagrangian and ALE simulations as well as with experimental data, where applicable. In all FSI computations the fluid is assumed to have properties of air with constant viscosity $\mu = 1.81 \times 10^{-5}$ kg/(m s), Prandtl number 0.72, and adiabatic index $\gamma = 1.4$.

4.1 Standalone solid computation: contact and impact modeling

Before presenting coupled FSI results, we first show the ability of the proposed formulation to handle standalone solid mechanics simulations, including contact and impact phenomena.

Impact of a metallic square bar on a rigid wall is simulated first. The bar has dimensions of 1 m × 1 m, and is made of steel with Young's modulus $E = 200$ GPa, density $\rho^s_0 = 7870\,\text{kg/m}^3$, Poisson ratio $\nu = 0.29$, yield stress $\sigma_y = 4.0 \times 10^8$ Pa, and plastic modulus $H = 1.0 \times 10^8$ Pa. The bar is placed a short distance away from the rigid wall and given the initial velocity of 200 m/s. No-penetration and free-slip boundary conditions are applied at the wall. The background domain has dimensions 4 m × 4 m and is discretized using 60 × 60 elements. The foreground domain is discretized using 20 × 20 elements.

As the bar approaches the wall, contact is intrinsically detected as a result of the velocity-field continuity and no-penetration boundary conditions, and the bar deforms plastically until the steady-state shape is reached. Figure 1 compares the final configuration of the bar and the equivalent plastic strain for the Lagrangian and immersed calculations. Very good agreement is observed between the two simulations. The small differences in the shape arise mainly due to the boundary conditions employed. In the Lagrangian computation no contact model is implemented, but rather no-slip and no-penetration conditions are imposed directly on the contact surface. As a result, the nodes in the Lagrangian computation stay on the contact surface. In the immersed

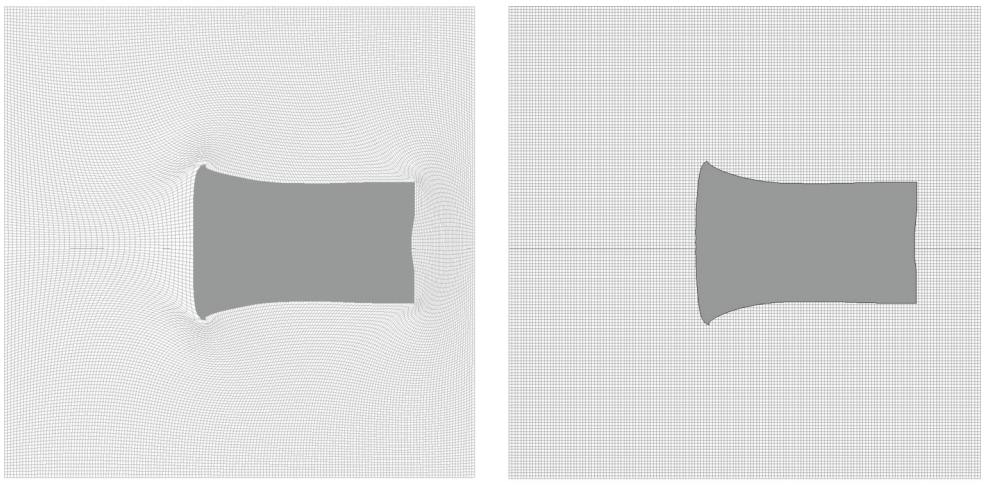
computation, the actual contact is modeled intrinsically, which results in slight "mushrooming" of the final shape. This mushrooming phenomenon is well known for Taylor-bar impact problems, and is reported, for example, in [53].

This example is developed further by considering contact between two square bars followed by impact on the rigid wall. The dimensions, material parameters, and meshes employed are the same as in the single-bar problem. A bar moving at 400 m/s impacts another bar, which is at rest, as shown in Fig. 2. After contact, the objects deform plastically and travel together at a lower speed, and eventually impact the wall, deform further, and come to rest. This sequence is illustrated in Fig. 2 where the deformed bar configuration is colored with the particle velocity, showing that the velocity field is continuous at the interface between the two objects. The proposed formulation is able to handle this contact-impact situation with a topology change without any difficulty.

4.2 Chamber detonation

In this fully coupled FSI example a bar is subjected to a detonation blast load. Figure 3 gives the problem description. A bar with dimensions 0.2 m $\times$ 0.1 m is placed at the center of a closed chamber with dimensions 0.4 m $\times$ 0.4 m. The bar thickness is set to 3.5 mm. The material properties of steel as in the previous section are assumed for the bar. The air in the chamber is initially at rest with $T = 270$ K and $p = 100{,}000$ Pa. The detonation is initiated by setting higher values of pressure ($p = 6{,}746{,}268.65$ Pa) and temperature ($T = 1,465$ K) in a semi-circular region centered on the left wall and with radius of 6.1 mm. Free-slip and no-penetration boundary conditions are assumed at the chamber walls. The background domain is discretized using 160×160 elements. The foreground domain is discretized using 80×40 elements.

Figure 4 shows snapshots of air pressure at different instants for both the ALE and immersed simulations,



(a) **(b)**

Fig. 6 Chamber detonation. Final configuration of the solid and surrounding mesh. **a** ALE; **b** immersed

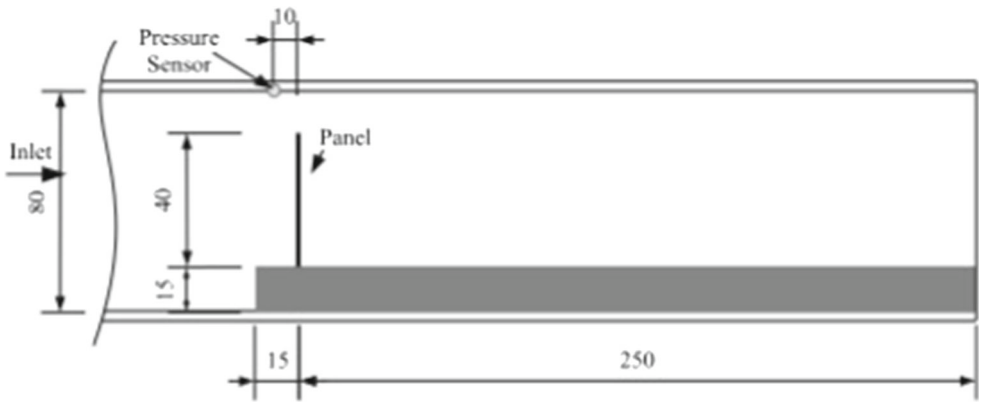


Fig. 7 Flexible panel subjected to a shock load. Problem geometry and setup

with good agreement achieved between the two solutions. Figure 5 depicts the time history of horizontal displacement of the center of the bar, air pressure at the center of detonation, and air pressure at the center of the right wall. These quantities are in very good agreement between the ALE and immersed computations.

Figure 6 shows the final shape of the bar as well as the deformed fluid mesh for the ALE computation. While very good agreement is achieved between the two solutions, the ALE simulation produces a mesh that is distorted in the vicinity of the bar's left-edge top and bottom corners.

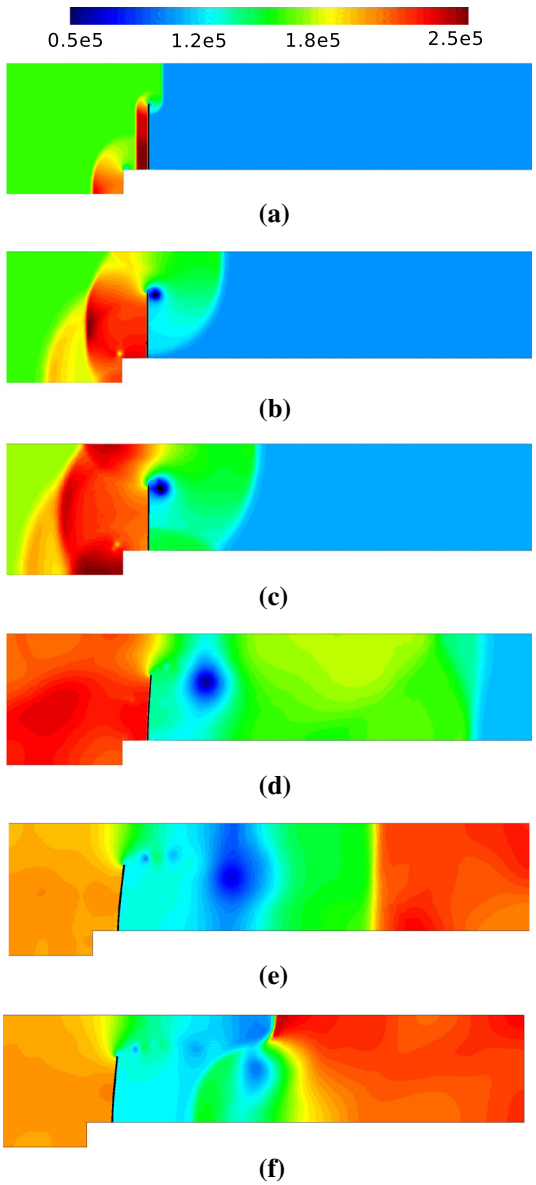


Fig. 8 Flexible panel subjected to a shock load. Pressure field and panel deflection at different time instants. **a** 70.0 μs, **b** 150 μs, **c** 200 μs, **d** 570 μs, **e** 1000 μs and **f** 1245 μs

4.3 Flexible panel subjected to a shock load

We compute one of the test cases presented in [89], wherein a panel of length 40 mm and thickness 1 mm is exposed to a shock load. The problem setup and dimensions are shown in Fig. 7. Reference [89] also provides the shock-tube experimental data for this setup, which we use to validate our computational methodology. The panel is assumed to be elastic with Young's modulus $E = 220$ GPa, density $\rho_0^s = 7600$ kg/m^3 and Poisson's ratio $\nu = 0.33$. Initially, the shock is placed 5 mm ahead of the step. The conditions behind the shock are assumed to be $\rho^f = 1.6458$ kg/m^3, $\|\mathbf{u}\| = 112.61$ m/s, and $p = 156{,}180.0$ Pa, while ahead of the shock air is assumed to be at rest with $\rho^f = 1.2$ kg/m^3 and $p = 100{,}000$ Pa. Slip boundary conditions are applied on the lateral and back walls of the shock tube. At the inlet the boundary conditions are set consistently with the initial conditions. The fluid mesh resolution around the panel is 0.3 mm, and the panel is discretized with uniform elements

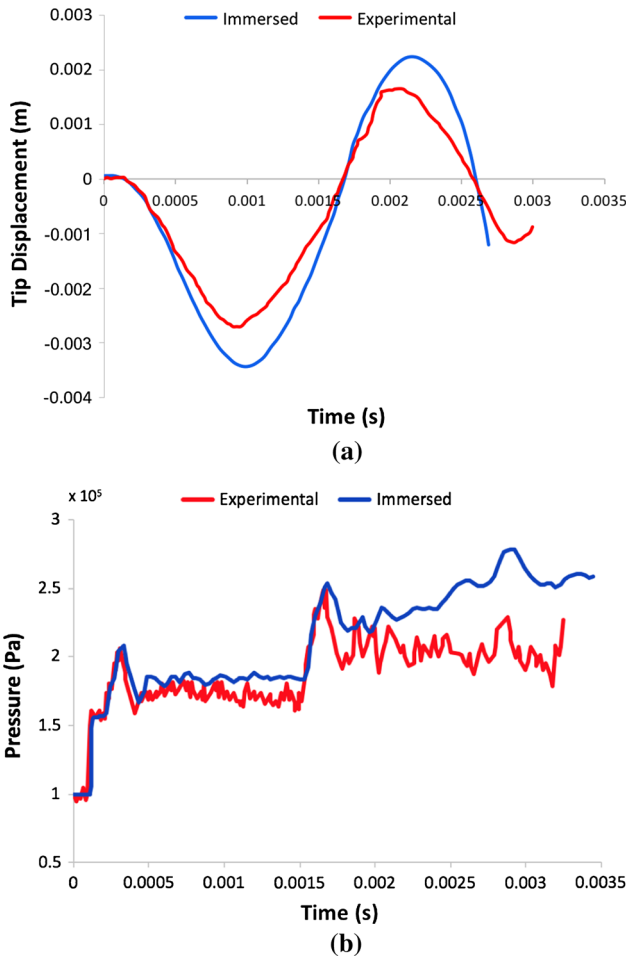


Fig. 9 Flexible panel subjected to a shock load. **a** History of the panel tip horizontal displacement; **b** history of the pressure at the pressure-sensor location. Experimental results of [89] are plotted for comparison

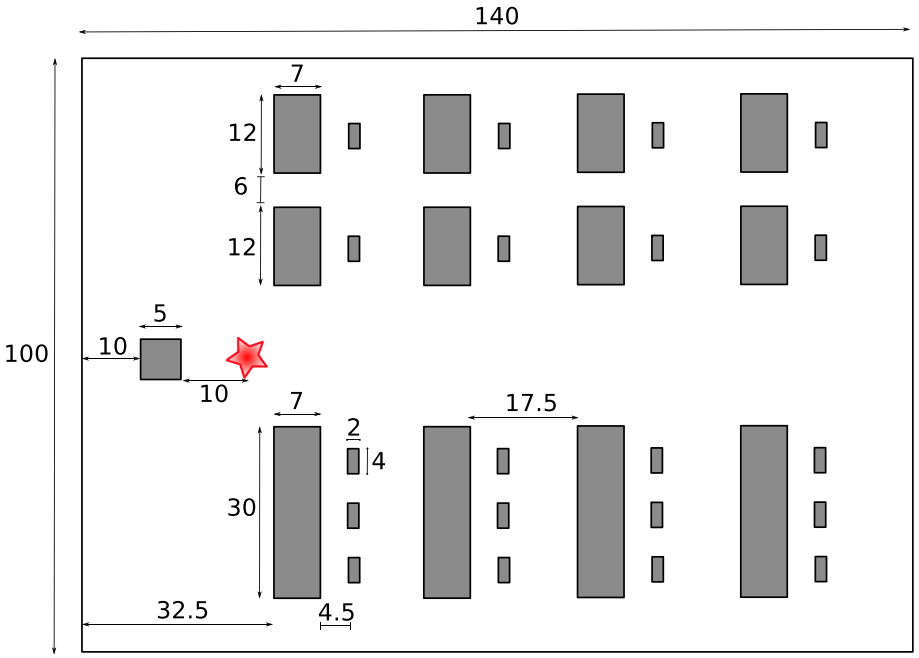


Fig. 10 Detonation with multiple objects. Problem geometry and setup. All dimensions are in cm

of size 0.05 mm. Note that, in order to minimize reflections from the left wall, the computational domain is extended behind the shock.

Figure 8 shows the pressure field along the tube at different instants during the simulation. The shock wave impacts the panel, which, in turn, begins to vibrate. The shock wave continues to travel to the right, while several other shock waves are initiated due to reflections from the panel and step.

Figure 9 compares the simulation results with experimental data for the panel tip horizontal displacement and pressure at a pressure-sensor location (shown in Fig. 7). Good agreement between the computational and experimental results is obtained. Panel vibration amplitude and period are captured well in the simulation. Simulation results for the pressure time history are also in good agreement with the experimental data in the earlier stages of the computation. However, deviation from the experimental curve after time $t = 2.2$ ms is observed, and is due to a spurious reflection wave that arrives from the left wall in the simulation. We note that such a discrepancy is also reported in the numerical simulations presented in [89].

4.4 Detonation with multiple objects

We conclude this section with a qualitative example showing the ability of the proposed methodology to naturally handle scenarios of detonation in the presence of multiple objects. Several rectangular objects are placed in a rectangular chamber with dimensions 1 m $\times$ 4 m, and are subjected to a detonation load. See Fig. 10 for the problem setup. To the right of the detonation location the larger objects have dimensions 0.3 m $\times$ 0.07 m, 0.12 m $\times$ 0.07 m, and 0.02 m $\times$ 0.04 m. The object to the left of the detonation location has dimensions 0.05 m $\times$ 0.05 m. The detonation is initiated by assuming air at rest with $T = 270$ K and $p = 100{,}000$ Pa, and elevating the pressure to $p = 6.7$ MPa and temperature to $T = 1465$ K in the zone of radius 0.055 m. All objects are assumed to be made of steel, as in previous examples, and are modeled as elastic. Slip boundary conditions are applied at the chamber walls. Uniform mesh with dimension 0.008 m for the air and 0.002 m for the solid objects is employed for the problem discretization.

Figure 11 shows the air speed and the solid deformed configuration at different instances after the detonation is initiated. Shock waves travel in the chamber and between the objects, forming complex patterns, while objects contact each other and impact the chamber walls without any restriction on their motion. This example clearly illustrates the general applicability of the proposed methodology.

5 Conclusions

In Part I of this paper, a computational framework for air-blast FSI based on an immersed approach is proposed. The framework couples compressible flow in the high-Mach-number regime with inelastic structures. The discrete formulation employes the background and foreground discretization as follows. Background discretization is fixed and provides the discrete trial and test function spaces for the coupled

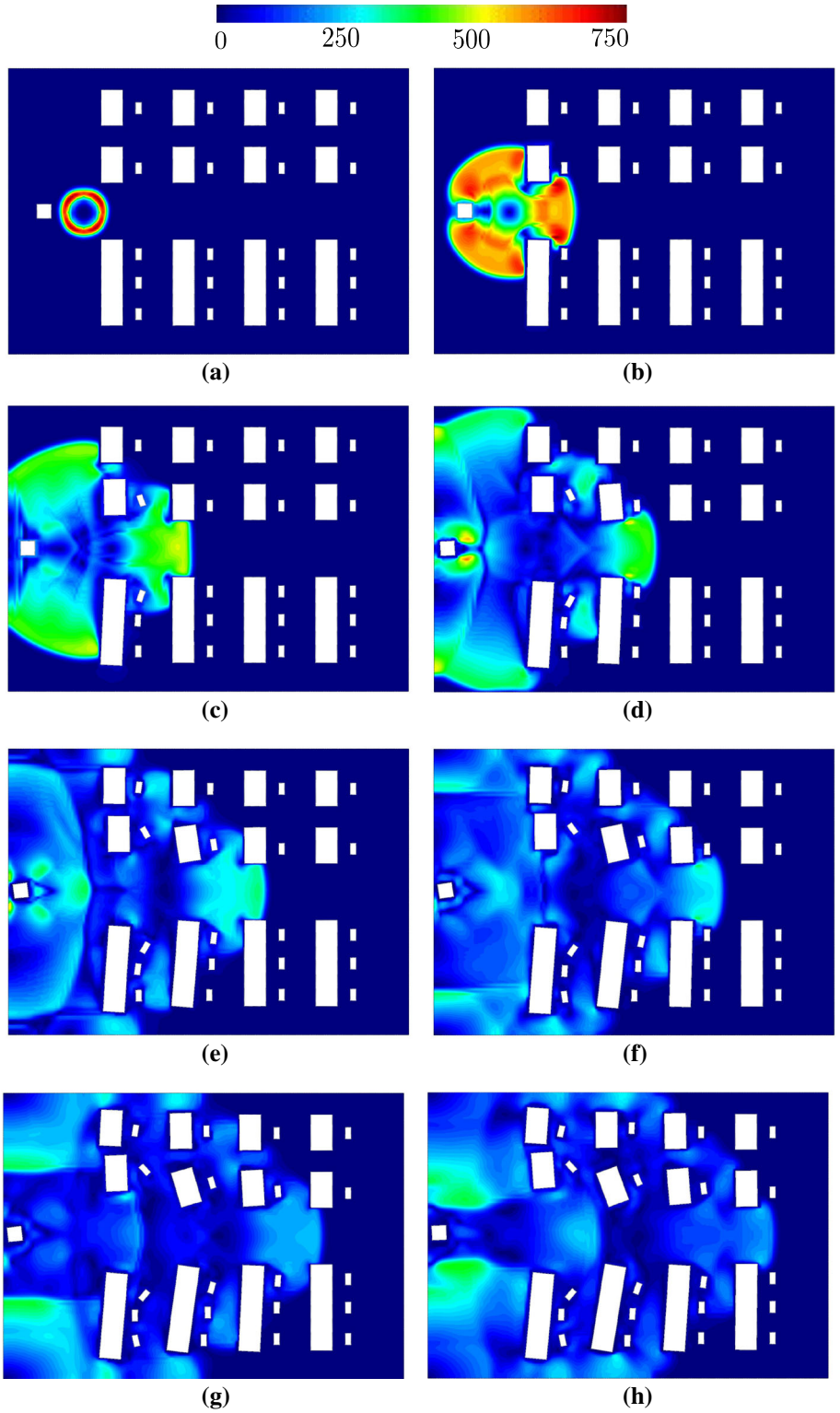


Fig. 11 Detonation with multiple objects. Air speed and solid in the current configuration at different instants during the simulation. **a** 0.03 ms, **b** 0.2 ms, **c** 0.4 ms, **d** 0.6 ms, **e** 0.8 ms, **f** 1.0 ms, **g** 1.2 ms and **h** 1.4 ms

FSI problem. Foreground discretization is moving with the solid material particles and is employed to track its current position, store history-dependent variables, and perform numerical quadrature.

The compressible-flow equations are discretized using pressure-primitive variables for compatibility with the solid DOFs and stabilized using residual-based SUPG and discontinuity-capturing techniques. These appear to be suffi-

cient to produce a robust computational methodology for air blast without resorting to classical shock viscosities, which are not consistent and often tend to produce overly diffusive results.

The proposed immersed approach has the advantage over existing embedded domain methods in that a monolithic FSI formulation is naturally obtained, even if the governing equations are advanced in time using an explicit algorithm. In addition, no restrictions on the solid motion is imposed in the framework, which enables handling of the domain topological changes with relative ease. Although, due to the utilization of an immersed approach, fluid mechanics accuracy near the solid surfaces is not as high as in a moving-mesh technique, preliminary comparisons with an ALE-based approach suggest that in the regime of air blast the overall accuracy of the coupled simulations does not suffer significantly. This is likely due to the fact that because of the fast dynamics of air blast, fluid boundary layers, which require higher mesh resolution near solid walls for good accuracy, do not have a chance to develop fully.

The framework was tested using standard low-order FEM discretization for the background and foreground domains. Ghost velocity stabilization was introduced in the coupled FSI formulation, and was found essential in stabilizing the velocity field near the fluid–solid interface. Preliminary 2D numerical examples presented indicate the good accuracy and robustness of the proposed computational methodology. Future work will involve 3D computations as well as the assessment of the framework's ability to represent solid disintegration and fragmentation during blast.

Acknowledgements KK was supported by the Secretary for Universities and Research of the Ministry of Economy and Knowledge of the Government of Catalonia and the Cofund program of the Marie Curie Actions of the 7th R&D Framework Program of the European Union under the Beatriu de Pinós grant. YB was partially supported by the ARO W911NF-14-1-0296 award. This support is gratefully acknowledged. The numerical examples were computed in the open-source parallel multiphysics software platform of KRATOS [90] developed at CIMNE. The pre- and post-processing of data is provided by the GID [91] software.

References

1. Hughes TJR, Liu WK, Zimmermann TK (1981) Lagrangian–Eulerian finite element formulation for incompressible viscous flows. Comput Methods Appl Mech Eng 29:329–349
2. Bazilevs Y, Hsu M-C, Takizawa K, Tezduyar TE (2012) ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction. Math Models Methods Appl Sci 22(supp02):1230002
3. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Kostov N, McIntyre S (2014) Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. Arch Comput Methods Eng 21:359–398
4. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid–structure interaction: methods and applications. Wiley, New York
5. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Challenges and directions in computational fluid–structure interaction. Math Models Methods Appl Sci 23:215–221
6. Bazilevs Y, Takizawa K, Tezduyar TE (2015) New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods. Math Models Methods Appl Sci 25:2217–2226
7. Tezduyar TE, Behr M, Liou J (1992) A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. Comput Methods Appl Mech Eng 94(3):339–351
8. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 28:1–44
9. Tezduyar TE, Behr M, Mittal S, Liou J (1992) A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. Comput Methods Appl Mech Eng 94(3):353–371
10. Tezduyar TE (2001) Finite element methods for flow problems with moving boundaries and interfaces. Arch Comput Methods Eng 8:83–130
11. Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters. Int J Numer Methods Fluids 43:555–575
12. Takizawa K, Bazilevs Y, Tezduyar TE (2012) Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. Arch Comput Methods Eng 19:171–225
13. Tezduyar TE, Sathe S (2007) Modeling of fluid–structure interactions with the space–time finite elements: solution techniques. Int J Numer Methods Fluids 54:855–900
14. Takizawa K, Tezduyar TE (2011) Multiscale space–time fluid–structure interaction techniques. Comput Mech 48:247–267
15. Takizawa K, Tezduyar TE (2012) Space–time fluid–structure interaction methods. Math Models Methods Appl Sci 22(supp02):1230001
16. Takizawa K, Montes D, McIntyre S, Tezduyar TE (2013) Space–time VMS methods for modeling of incompressible flows at high Reynolds numbers. Math Models Methods Appl Sci 23:223–248
17. Takizawa K, Bazilevs Y, Tezduyar TE, Long CC, Marsden AL, Schjodt K (2014) ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling. Math Models Methods Appl Sci 24:2437–2486
18. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Øiseth O, Mathisen KM, Kostov N, McIntyre S (2014) Engineering analysis and design with ALE-VMS and space–time methods. Arch Comput Methods Eng 21:481–508
19. Takizawa K (2014) Computational engineering analysis with the new-generation space–time methods. Comput Mech 54:193–211
20. Takizawa K, Tezduyar TE, Boswell C, Tsutsui Y, Montel K (2015) Special methods for aerodynamic-moment calculations from parachute FSI modeling. Comput Mech 55:1059–1069
21. Takizawa K, Tezduyar TE, Kolesar R (2015) FSI modeling of the Orion spacecraft drogue parachutes. Comput Mech 55:1167–1179
22. Takizawa K, Tezduyar TE, Kuraishi T (2015) Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. Math Models Methods Appl Sci 25:2227–2255
23. Takizawa K, Tezduyar TE, Mochizuki H, Hattori H, Mei S, Pan L, Montel K (2015) Space–time VMS method for flow computations with slip interfaces (ST-SI). Math Models Methods Appl Sci 25:2377–2406

24. Takizawa K, Tezduyar TE, Kuraishi T, Tabata S, Takagi H (2016) Computational thermo-fluid analysis of a disk brake. Comput Mech 57:965–977
25. Takizawa K, Tezduyar TE, Hattori H (2017) Computational analysis of flow-driven string dynamics in turbomachinery. Comput Fluids 142:109–117
26. Takizawa K, Tezduyar TE, Otoguro Y, Terahara T, Kuraishi T, Hattori H (2017) Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA). Comput Fluids 142:15–20
27. Takizawa K, Tezduyar TE, Terahara T (2016) Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA). Comput Fluids 141:191–200
28. Takizawa K, Tezduyar TE, Buscher A, Asada S (2014) Space–time interface-tracking with topology change (ST-TC). Comput Mech 54:955–971
29. Takizawa K, Tezduyar TE, Buscher A, Asada S (2014) Space–time fluid mechanics computation of heart valve models. Comput Mech 54:973–986
30. Takizawa K, Tezduyar TE, Buscher A (2015) Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping. Comput Mech 55:1131–1141
31. Takizawa K, Tezduyar TE, Asada S, Kuraishi T (2016) Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC). Comput Fluids 141:124–134
32. Takizawa K, Tezduyar TE, Terahara T, Sasaki T (2016) Heart valve flow computation with the integrated Space–Time VMS, Slip Interface, Topology Change and Isogeometric Discretization methods. Comput Fluids. doi:10.1016/j.compfluid.2016.11.012
33. Hauke G, Hughes TJR (1998) A comparative study of different sets of variables for solving compressible and incompressible flows. Comput Methods Appl Mech Eng 153(1):1–44
34. Hauke G (2001) Simple stabilizing matrices for the computation of compressible flows in primitive variables. Comput Methods Appl Mech Eng 190(51):6881–6893
35. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 32:199–259
36. Hughes TJR, Tezduyar TE (1984) Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. Comput Methods Appl Mech Eng 45:217–284
37. Le Beau GJ, Ray SE, Aliabadi SK, Tezduyar TE (1993) SUPG finite element computation of compressible flows with the entropy and conservation variables formulations. Comput Methods Appl Mech Eng 104:397–422
38. Tezduyar TE, Senga M (2006) Stabilization and shock-capturing parameters in SUPG formulation of compressible flows. Comput Methods Appl Mech Eng 195:1621–1632
39. Hughes TJR, Scovazzi G, Tezduyar TE (2010) Stabilized methods for compressible flows. J Sci Comput 43:343–368. doi:10.1007/s10915-008-9233-5
40. Hughes TJR, Mallet M, Mizukami A (1986) A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. Comput Methods Appl Mech Eng 54:341–355
41. Hughes TJR, Mallet M (1986) A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. Comput Methods Appl Mech Eng 58:329–339
42. Tezduyar TE, Senga M, Vicker D (2006) Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing. Comput Mech 38:469–481
43. Tezduyar TE, Senga M (2007) SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing. Comput Fluids 36:147–159
44. Rispoli F, Saavedra R, Corsini A, Tezduyar TE (2007) Computation of inviscid compressible flows with the V-SGS stabilization and YZβ shock-capturing. Int J Numer Methods Fluids 54:695–706
45. Rispoli F, Saavedra R, Menichini F, Tezduyar TE (2009) Computation of inviscid supersonic flows around cylinders and spheres with the V-SGS stabilization and YZβ shock-capturing. J Appl Mech 76:021209
46. Rispoli F, Delibra G, Venturini P, Corsini A, Saavedra R, Tezduyar TE (2015) Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing. Comput Mech 55:1201–1209
47. Löhner R, Luo H, Baum JD, Rice D (2008) Improvements in speed for explicit, transient compressible flow solvers. Int J Numer Methods Fluids 56:2229–2244
48. Löhner R, Baum JD, Mestreau E, Sharov D, Charman C, Pelessone D (2004) Adaptive embedded unstructured grid methods. Int J Numer Methods Eng 60(3):641–660
49. Löhner R, Cebral RJ, Camelli FE, Appanaboyina S, Baum JD, Mestreau EL, Soto OA (2008) Adaptive embedded and immersed unstructured grid techniques. Comput Methods Appl Mech Eng 197:2173–2197
50. Hansbo A, Hansbo P, Larson MG (2003) A finite element method on composite grids based on Nitsche's method. ESAIM Math Model Numer Anal 37(3):495–514
51. Glowinski R, Pan TW, Hesla TI, Joseph DD, Periaux J (2001) A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. J Comput Phys 169(2):363–426
52. Burman E, Hansbo P (2012) Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. Appl Numer Math 62(4):328–341
53. Benson DJ, Okazawa S (2004) Contact in a multi-material Eulerian finite element formulation. Comput Methods Appl Mech Eng 193(39):4277–4298
54. Puso MA, Kokko E, Settgast R, Sanders J, Simpkins B, Liu B (2015) An embedded mesh method using piecewise constant multipliers with stabilization: mathematical and numerical aspects. Int J Numer Methods Eng 104(7):697–720
55. Wang X, Liu WK (2004) Extended immersed boundary method using FEM and RKPM. Comput Methods Appl Mech Eng 193:1305–1321
56. Sotiropoulos F, Yang X (2014) Immersed boundary methods for simulating fluid–structure interaction. Prog Aerosp Sci 65:1–21
57. Liu WK, Kim DW, Tang S (2007) Mathematical foundations of the immersed finite element method. Comput Mech 39(3):211–222
58. Zhang LT, Gerstenberger A, Wang X, Liu WK (2004) Immersed finite element method. Comput Methods Appl Mech Eng 193:2051–2067
59. Sulsky D, Chen Z, Schreyer HL (1994) A particle method for history-dependent materials. Comput Methods Appl Mech Eng 118(1):179–196
60. Sadeghirad A, Brannon RM, Burghardt J (2011) A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. Int J Numer Methods Eng 86:1435–1456
61. Parvizian J, Düster A, Rank E (2007) Finite cell method. Comput Mech 41(1):121–133
62. Casquero H, Bona-Casas C, Gomez H (2015) A NURBS-based immersed methodology for fluid–structure interaction. Comput Methods Appl Mech Eng 284:943–970
63. Hsu M-C, Kamensky D, Bazilevs Y, Sacks MS, Hughes TJR (2014) Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. Comput Mech 54:1055–1071

64. Kamensky D, Hsu M-C, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR (2015) An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. Comput Methods Appl Mech Eng 284:1005–1053
65. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput Methods Appl Mech Eng 194:4135–4195
66. Cottrell JA, Hughes TJR, Bazilevs Y (2009) Isogeometric analysis: toward integration of CAD and FEA. Wiley, New York
67. Chen J-S, Belytschko T (2015) Meshless and meshfree methods. In: Engquist B (ed) Encyclopedia of applied and computational mathematics. Springer, Berlin, pp 886–894
68. Hauke G, Hughes TJR (1994) A unified approach to compressible and incompressible flows. Comput Methods Appl Mech Eng 113:389–396
69. Belytschko T, Liu WK, Moran B (2000) Nonlinear finite elements for continua and structures. Wiley, New York
70. Lubliner J (1990) Plasticity theory. Macmillan Publishing Company, London
71. Simo JC, Hughes TJR (1998) Computational inelasticity. Springer, New York
72. Casquero H, Bona-Casas C, Gomez H (2015) A NURBS-based immersed methodology for fluid–structure interaction. Comput Methods Appl Mech Eng 284:943–970
73. Xu F, Moutsanidis G, Kamensky D, Hsu M-C, Murugan M, Ghoshal A, Bazilevs Y (2017) Compressible flows on moving domains: stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling. Comput Fluids. doi:10.1016/j.compfluid.2017.02.006
74. Von Neumann J, Richtmyer RD (1950) A method for the numerical calculation of hydrodynamic shocks. J Appl Phys 21:232–237
75. Noh WF (1987) Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. J Comput Phys 72:78–120
76. Hughes TJR (1980) Generalization of selective integration procedure to anisotropic and nonlinear media. Int J Numer Methods Eng 15:1413–1418
77. Belytschko T, Bindeman LP (1991) Assumed strain stabilization of the 4-node quadrilateral with 1-point quadrature for nonlinear problems. Comput Methods Appl Mech Eng 88(3):311–340
78. Zhu YY, Cescotto S (1995) Unified and mixed formulation of the 4-node quadrilateral elements by assumed strain method: application to thermomechanical problems. Int J Numer Methods Eng 38(4):685–716
79. Elguedj T, Bazilevs Y, Calo VM, Hughes TJR (2008) B-bar and F-bar projection methods for nearly incompressible linear and nonlinear elasticity and plasticity using higher-order NURBS elements. Comput Methods Appl Mech Eng 197:2732–2762
80. Massing A, Larson MG, Logg A, Rognes ME (2014) A stabilized Nitsche fictitious domain method for the Stokes problem. J Sci Comput 61(3):604–628
81. Chung J, Hulbert GM (1993) A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. J Appl Mech 60:371–375
82. Jansen KE, Whiting CH, Hulbert GM (2000) A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. Comput Methods Appl Mech Eng 190(3):305–319
83. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid–structure interaction: theory, algorithms, and computations. Comput Mech 43:3–37
84. Shakib F, Hughes TJR, Johan Z (1991) A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations. Comput Methods Appl Mech Eng 89(1):141–219
85. Oñate E, Owen R (2011) Particle-based methods: fundamentals and applications, vol 25. Springer, New York
86. Idelsohn SR, Marti J, Becker P, Oñate E (2014) Analysis of multifluid flows with large time steps using the particle finite element method. Int J Numer Methods Fluids 75(9):621–644
87. Benson DJ (1992) Computational methods in Lagrangian and Eulerian hydrocodes. Comput Methods Appl Mech Eng 99(2):235–394
88. Benson DJ (1997) The numerical simulation of the dynamic compaction of powders. Springer, New York
89. Giordano J, Jourdan G, Burtschell Y, Medale M, Zeitoun DE, Houas L (2005) Shock wave impacts on deforming panel, an application of fluid-structure interaction. Shock Waves 14(1–2):103–110
90. Dadvand P, Rossi R, Oñate E (2010) An object-oriented environment for developing finite element codes for multi-disciplinary applications. Arch Comput Methods Eng 17(3):253–297
91. Coll A, Ribó R, Pasenau M, Escolano E, Perez J, Melendo A, Monros A, Gárate J (2016) GiD v.13 reference manual