

Željko Tuković
Aleksandar Karač
Philip Cardiff
Hrvoje Jasak
Alojz Ivanković

<https://doi.org/10.21278/TOF.42301>

ISSN 1333–1124

eISSN 1849–1391

OpenFOAM FINITE VOLUME SOLVER FOR FLUID-SOLID INTERACTION

Summary

This paper describes a self-contained parallel fluid-structure interaction solver based on a finite volume discretisation, where a strongly coupled partitioned solution procedure is employed. The incompressible fluid flow is described by the Navier-Stokes equations in the arbitrary Lagrangian-Eulerian form, and the solid deformation is described by the Saint Venant-Kirchhoff hyperelastic model in the total Lagrangian form. Both the fluid and the solid are discretised in space using the second-order accurate cell-centred finite volume method, and temporal discretisation is performed using the second-order accurate implicit scheme. The method, implemented in open-source software OpenFOAM, is parallelised using the domain decomposition approach and the exchange of information at the fluid-solid interface is handled using *global face zones*. The performance of the solver is evaluated in standard two- and three-dimensional cases and excellent agreement with the available numerical results is obtained.

Key words: fluid-structure interaction, finite volume method, partitioned, strongly coupled, parallel, OpenFOAM

1. Introduction

Broadly speaking, there are three different coupling procedures that can be used to solve fluid-structure interaction (FSI) problems, as shown in Fig. 1 [1]:

1. separate analysis method, or hybrid method,
2. single analysis method,
3. single fluid-structure domain.

The first two belong to the so-called *partitioned methods*, where the solid and fluid domains are solved separately with their respective mesh discretisation and numerical method (see [2] for a comprehensive review). The coupling is performed through the passing of informations over fluid-solid interface. The difference between the two approaches is in the method used to discretise the separate domains; in the former, two different methods are used (*e.g.* Finite Element (FE) for the solid domain and Finite Volume (FV) for the fluid domain), whereas in the latter the same method is used for both domains (*e.g.* FE or FV). The third method is nowadays commonly known as the *monolithic approach*, where the coupled problem is solved as one entity using a single numerical procedure [3, 4, 5, 6]. There are benefits of and drawbacks to

using both the partitioned and monolithic approaches (see *e.g.* [5, 6] for details) and the choice may depend on the FSI problem.

The present study is concerned with the development of a self-contained fluid-structure interaction solver for analysis of the interaction between an incompressible fluid and an elastic compressible structure exhibiting large displacements and rotations. The solver is based on the cell-centred finite volume discretisation method for both domains and strongly coupled partitioned solution procedure (Fig. 1b). To the authors' best knowledge, there are few partitioned FV-based FSI solvers found in literature [7, 8, 9, 10, 11, 12], with the current work being a significant extension of the authors' previous contributions [13, 14, 15, 16, 17]. The main contribution of this work is the development of an FSI solver that is second-order accurate in both space and time and capable of exploiting arbitrary polyhedral meshes and modern parallel computing architectures.

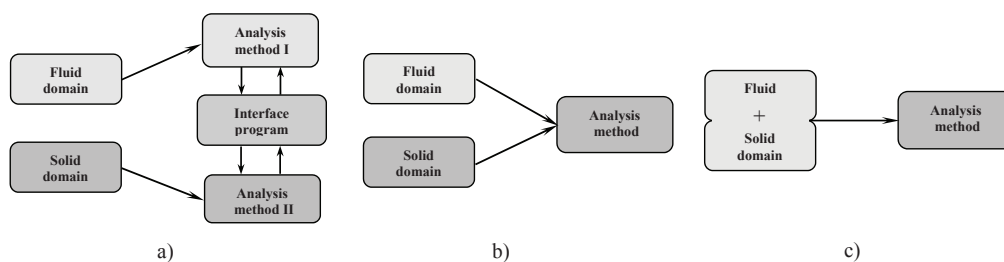


Fig. 1 Fluid-structure coupling procedures.

The fluid part of the model is described by the Navier-Stokes equations in the arbitrary Lagrangian-Eulerian form. The model is discretised on a moving polyhedral mesh using a second-order accurate FV method; the resulting pressure-velocity coupled system is solved using the iterative PISO [18] algorithm, where the discretised pressure equation is derived by combining the discretised momentum and continuity equations with the Rhie-Chow momentum interpolation method [19]. A deficiency of such an approach is temporal inconsistency of the numerical model, caused by the application of the standard Rhie-Chow interpolation; the authors have recently proposed a remedy to this problem by deriving a consistent Rhie-Chow interpolation on moving FV meshes [20, 21]. A detailed interpretation of the proposed consistent Rhie-Chow interpolation and its extension to third-order temporal accuracy was later given by Gillebaart et al. [7]. As the fluid component of the current FSI solver is well described in [20, 21], only a concise overview of the method will be given here.

The focus of the present paper is on the structural component of the partitioned FSI procedure, as well as on the coupling of the fluid and solid regions in the context of the parallel computing. As has been stated earlier, the mathematical model governing the structural deformation is discretised using the cell-centred second-order accurate FV method, echoing closely the fluid model discretisation. The origin of the current structural numerical approach is the model firstly proposed by Demirdžić and Muzaferija [22, 23]; this FV structural model was later extended and implemented in the OpenFOAM framework and used as a part of the OpenFOAM toolbox by many researchers [24, 25, 26, 27, 28, 29, 30, 31, 32, 33]. In the aforementioned works, the second-order accuracy in space is achieved by using the mid-point rule to approximate the surface and volume integrals of the mathematical model. A deficiency in the approach that affects the accuracy of the method is the interpolation of the cell-centred displacement gradient to the faces, rather than the direct calculation of the gradient at the faces; this has a negative impact on the accuracy, especially at the boundary faces, where the tangential gradient is extrapolated

using zero-order extrapolation. In the present work, an improvement over previous methods is proposed, which aims to increase the accuracy of the discretisation on arbitrary polyhedral meshes. The new discretisation is achieved by directly calculating the gradient at the internal and boundary faces using central differencing for the normal gradient and a vertex-based Gauss-Green method for the tangential gradient. This approach requires an interpolation of the displacement field from the cell-centres to the vertices, which is far from a trivial task in the context of parallel computing. The proposed approach has proven to provide significant increases in the accuracy on tetrahedral as well as arbitrary polyhedral meshes.

An additional contribution of the current work is the development of a new concept of the FSI model set-up using the domain decomposition parallel execution. The fluid and structural parts of the computational mesh are constructed and separately decomposed, which means that one fluid and one structural sub-mesh are assigned to each CPU processor. The interface between the fluid and the structure regions is represented by a pair of so-called *global face zones* that are present on all processors. The global face zone point and face fields are reconstructed individually on each processor and, subsequently, global synchronisation is performed for points/faces shared by multiple processors. The impact of such an approach on the parallel calculation performance is investigated in one of the presented cases.

Finally, it should be noted that the presented solver is a part of an open-source framework (OpenFOAM). As such, it is open to scrutiny by other researchers and available for further developments and modifications, *e.g.* for new types of FSI problems (such as [34], where the turbulent flow of compressible fluid is considered), with no additional costs.

The article is organised as follows: Section 2 outlines the mathematical model derived from the governing momentum and continuity equations and constitutive relations, where a total Lagrangian mathematical model is employed for the structural model. The discretisation of the mathematical model is given in Section 3, where the newly developed tangential gradient calculation method is presented; Section 4 presents the application of the method to a number of benchmark cases, where the accuracy, order of accuracy and parallel efficiency of the method are demonstrated.

2. Mathematical model

In the current article, the interaction between an incompressible Newtonian fluid and a hyperelastic solid is considered, where a FV discretisation and a partitioned approach are employed. Mathematical models governing the mechanical behaviour of the fluid and the solid are solved separately, and coupling is achieved by the enforcement of proper boundary conditions at the interface.

2.1. Fluid governing equations

Fluid flow is considered on a spatial domain whose shape is changing in time due to the deformation of the fluid-solid interface. Isothermal flow of an incompressible Newtonian fluid, inside an arbitrary volume V bounded by a closed moving surface S , is governed by the mass and the linear momentum conservation laws:

$$\oint_S \mathbf{n} \cdot \mathbf{v} \, dS = 0 \quad (1)$$

$$\begin{aligned} & \frac{d}{dt} \int_V \mathbf{v} \, dV + \oint_S \mathbf{n} \cdot (\mathbf{v} - \mathbf{v}_s) \mathbf{v} \, dS \\ &= \oint_S \mathbf{n} \cdot (\nu \nabla \mathbf{v}) \, dS - \frac{1}{\rho} \int_V \nabla p \, dV, \end{aligned} \quad (2)$$

where \mathbf{n} is the outward pointing unit normal to S , \mathbf{v} is the fluid velocity, \mathbf{v}_s is the velocity of surface S , ν is the fluid kinematic viscosity, p is the fluid pressure, and ρ is the fluid density. The relationship between the rate of change of the volume V and the velocity \mathbf{v}_s is defined by the *geometric (space) conservation law* (GCL, see [35, 36]):

$$\frac{d}{dt} \int_V dV - \oint_S \mathbf{n} \cdot \mathbf{v}_s \, dS = 0. \quad (3)$$

The mathematical model presented in Eq. (3) is usually referred to as the arbitrary Lagrangian-Eulerian (ALE) formulation.

2.2. Solid governing equations

The deformation of the solid, assumed to be elastic and compressible, can be described by the linear momentum conservation law in the total Lagrangian form:

$$\int_{V_0} \rho_0 \frac{\partial}{\partial t} \left(\frac{\partial \mathbf{u}}{\partial t} \right) \, dV = \int_{S_0} \mathbf{n} \cdot (\boldsymbol{\Sigma} \cdot \mathbf{F}^T) \, dS + \int_{V_0} \rho_0 \mathbf{b} \, dV, \quad (4)$$

where subscript 0 indicates the quantities related to the initial (undeformed) configuration, \mathbf{u} is the displacement vector, $\mathbf{F} = \mathbf{I} + (\nabla \mathbf{u})^T$ is the deformation gradient tensor, \mathbf{I} is the second-order identity tensor, and $\boldsymbol{\Sigma}$ is the second Piola-Kirchhoff stress tensor, which is related to the Cauchy stress tensor $\boldsymbol{\sigma}$ by the following expression:

$$\boldsymbol{\sigma} = \frac{1}{\det \mathbf{F}} \mathbf{F} \cdot \boldsymbol{\Sigma} \cdot \mathbf{F}^T. \quad (5)$$

To account for finite strains, the St. Venant-Kirchhoff constitutive material model is used; this model connects the second Piola-Kirchhoff stress tensor with the Green-Lagrange strain tensor as follows

$$\boldsymbol{\Sigma} = 2\mu \mathbf{E} + \lambda \operatorname{tr}(\mathbf{E}) \mathbf{I}, \quad (6)$$

where μ and λ are the Lamé coefficients. The Green-Lagrange strain tensor is defined as

$$\mathbf{E} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T + \nabla \mathbf{u} \cdot (\nabla \mathbf{u})^T]. \quad (7)$$

Substituting the constitutive relation, Eq. (6), into the governing equation, Eq. (4), one obtains the linear momentum conservation equation for a St. Venant-Kirchhoff hyperelastic solid in the total Lagrangian form, where the displacement vector \mathbf{u} is the primitive variable:

$$\rho_0 \int_{V_0} \frac{\partial}{\partial t} \left(\frac{\partial \mathbf{u}}{\partial t} \right) \, dV - \oint_{S_0} \mathbf{n} \cdot (2\mu + \lambda) \nabla \mathbf{u} \, dS = \oint_{S_0} \mathbf{n} \cdot \mathbf{q} \, dS + \rho_0 \int_{V_0} \mathbf{b} \, dV. \quad (8)$$

where

$$\begin{aligned} \mathbf{q} = & \mu(\nabla\mathbf{u})^T + \lambda \operatorname{tr}(\nabla\mathbf{u}) \mathbf{I} - (\mu + \lambda)\nabla\mathbf{u} \\ & + \mu \nabla\mathbf{u} \cdot (\nabla\mathbf{u})^T + \frac{1}{2}\lambda \operatorname{tr}[\nabla\mathbf{u} \cdot (\nabla\mathbf{u})^T] \mathbf{I} + \Sigma \cdot \nabla\mathbf{u}. \end{aligned} \quad (9)$$

Tensor \mathbf{q} consists of nonlinear and displacement component coupling terms that are treated explicitly after discretisation, allowing the resulting linear system to be solved using a segregated algorithm. The diffusivity $(2\mu + \lambda)$ in the Laplacian at the left-hand side of Eq. (8) is used to maximise the implicit part of the discretised equation [24].

2.3. Conditions at the fluid-solid interface

The fluid and solid models are coupled by kinematic and dynamic conditions which must be satisfied at the fluid-solid interface. The kinematic condition states that the velocity and displacement must be continuous across the interface:

$$\mathbf{v}_{F,i} = \mathbf{v}_{S,i}, \quad (10)$$

$$\mathbf{u}_{F,i} = \mathbf{u}_{S,i} \quad (11)$$

where subscripts F and S represent the quantities corresponding to the fluid and solid regions of the model, respectively, and subscript i represents the quantities at the fluid-solid interface.

The dynamic condition follows from the linear momentum conservation law and states that forces are in equilibrium at the interface:

$$\mathbf{n}_i \cdot \boldsymbol{\sigma}_{F,i} = \mathbf{n}_i \cdot \boldsymbol{\sigma}_{S,i}, \quad (12)$$

where \mathbf{n}_i is the unit normal vector at the interface.

In the partitioned computational approach, force (traction) is usually calculated at the fluid side of the interface and applied as a boundary condition at the solid side of the interface. The traction is calculated by using the fluid stress tensor consisting of the isotropic and viscous components. The viscous (deviatoric) component, defined by Newton's law of viscosity, is:

$$\boldsymbol{\tau} = \mu [\nabla\mathbf{v} + \nabla\mathbf{v}^T] \quad (13)$$

where $\mu = \rho\nu$ is the fluid dynamic viscosity. Hence, the stress tensor for an incompressible Newtonian fluid reads as follows:

$$\boldsymbol{\sigma}_{F,i} = -p\mathbf{I} + \boldsymbol{\tau} = -p\mathbf{I} + \mu [\nabla\mathbf{v} + \nabla\mathbf{v}^T]. \quad (14)$$

and the traction at the interface reads:

$$\mathbf{t}_{F,i} = \mathbf{n}_i \cdot \boldsymbol{\sigma}_{F,i} = -p\mathbf{n}_i + \mu\mathbf{n}_i \cdot \nabla\mathbf{v}_t - 2\mu(\nabla_s \cdot \mathbf{v})\mathbf{n}_i + \mu\nabla_s v_n, \quad (15)$$

where $\mathbf{v}_t = (\mathbf{I} - \mathbf{nn}) \cdot \mathbf{v}$ is the tangential velocity component, $\nabla_s = \nabla - \mathbf{nn} \cdot \nabla$ is the surface tangential gradient operator and $v_n = \mathbf{n} \cdot \mathbf{v}$ is the normal velocity component. The third and the fourth term on the right hand side of equation (15) can usually be omitted; otherwise, these terms can be calculated directly at the solid side of the interface taking that the kinematic condition is valid.

3. Numerical model

The discretisation procedure is separated into two distinct parts: discretisation of the computational domain and discretisation of the governing equations.

3.1. Discretisation of the computational domain

The time interval is split into a finite number of time-steps Δt and the equations are solved in a time-marching manner. In general, the computational space is divided into a finite number of convex polyhedral control volumes (CVs) or cells bounded by convex polygons. Cells do not overlap and fill the spatial domain completely. Figure 2 shows a polyhedral control volume V_P with the computational point P located in its centroid, f is an arbitrary face with area S_f and unit normal vector \mathbf{n}_f , and it is shared with the neighbouring CV with the centroid N . The geometry of the CV is exactly determined by the position of its vertices.

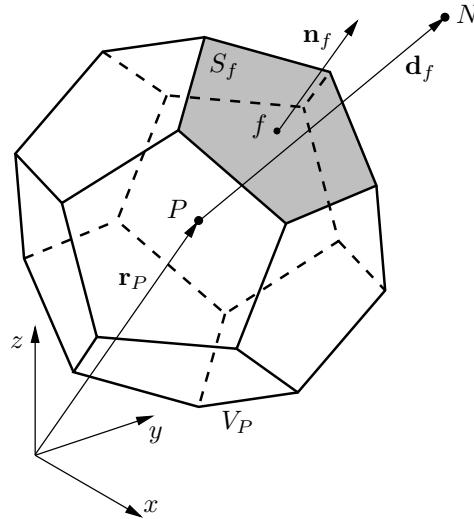


Fig. 2 Polyhedral control volume (cell).

While the fluid flow mesh changes over time due to the deformation of the fluid-solid interface, the solid mesh is always in its initial (undeformed) configuration due to the use of the total Lagrangian formulation.

3.2. Discretisation of the governing equations

The discretised fluid mathematical model with a moving polyhedral mesh consists of the discretised momentum equation and the discretised pressure equation, which is derived from the discretised continuity equation using the Rhie-Chow momentum interpolation method [19]. The FV discretisation of the fluid model is described in detail in [20]. However, it should be noted that in [20], emphasis was on the moving interface between two fluid phases, and boundary conditions on the moving walls were not considered. A comprehensive description of moving wall boundary conditions used in this work can be found in [7]. One exception is the calculation of the pressure at walls; instead of the typical practice of calculating pressure using the zero-gradient extrapolation, pressure is calculated using the linear extrapolation from the neighbouring cell centre. Fluid mesh deformation is performed using the Laplace mesh motion equation with variable diffusivity, as described in [20, 37]. Instead of using an FE discretisation for the Laplace mesh motion equation, this work discretises the motion equation using the cell-centred FV method, by which vertex displacements are obtained using a reconstruction procedure described in Appendix A.

In terms of the solid model, the second-order FV discretisation of the integral conservation equation, Eq. (8), transforms the surface integrals into sums of face integrals; the face integrals and volume integrals are then approximated using the second-order accurate mid-point rule. The spatially discretised counterpart of the momentum equation, Eq. (8), for the control volume

V_P reads:

$$\rho_{0P} \left[\frac{\partial}{\partial t} \left(\frac{\partial \mathbf{u}}{\partial t} \right) \right]_P V_P - \sum_f (2\mu_f + \lambda_f) \mathbf{n}_f \cdot (\nabla \mathbf{u})_f S_f = \sum_f \mathbf{n}_f \cdot \mathbf{q}_f S_f + \rho_{0P} \mathbf{b}_P V_P, \quad (16)$$

where subscript P represents the cell-centre value and subscript f represents the face-centre value.

The temporal discretisation of Eq. (16) is performed using the second-order accurate three-time-level implicit scheme [38], referred to as the backward scheme. All terms in Eq. (16) are evaluated at a new time instance as $t^{[m]} = t^{[m-1]} + \Delta t$. The acceleration $\frac{\partial}{\partial t} \left(\frac{\partial \mathbf{u}}{\partial t} \right)$ and the velocity $\frac{\partial \mathbf{u}}{\partial t}$ are discretised at the new time instance using the three-time-level finite difference formula as follows:

$$\left[\frac{\partial}{\partial t} \left(\frac{\partial \mathbf{u}}{\partial t} \right) \right]_P^{[m]} = \frac{3 \left(\frac{\partial \mathbf{u}}{\partial t} \right)_P^{[m]} - 4 \left(\frac{\partial \mathbf{u}}{\partial t} \right)_P^{[m-1]} + \left(\frac{\partial \mathbf{u}}{\partial t} \right)_P^{[m-2]}}{2\Delta t}, \quad (17)$$

$$\left(\frac{\partial \mathbf{u}}{\partial t} \right)_P^{[m]} = \frac{3\mathbf{u}_P^{[m]} - 4\mathbf{u}_P^{[m-1]} + \mathbf{u}_P^{[m-2]}}{2\Delta t}. \quad (18)$$

The face normal gradient of displacement $\mathbf{n}_f \cdot (\nabla \mathbf{u})_f$ is discretised using the central scheme with non-orthogonal and skewness correction (see Fig.2):

$$\mathbf{n}_f \cdot (\nabla \mathbf{u})_f = \frac{\mathbf{u}_N - \mathbf{u}_P}{d_{fn}} + \frac{\mathbf{k}_N \cdot (\nabla \mathbf{u})_N - \mathbf{k}_P \cdot (\nabla \mathbf{u})_P}{d_{fn}}, \quad (19)$$

where d_{fn} is the normal distance between points N and P ,

$$d_{fn} = \mathbf{n}_f \cdot \mathbf{d}_f = \mathbf{n}_f \cdot (\mathbf{r}_N - \mathbf{r}_P), \quad (20)$$

and \mathbf{k}_P and \mathbf{k}_N are the correction vectors calculated as follows:

$$\mathbf{k}_P = (\mathbf{I} - \mathbf{n}_f \mathbf{n}_f) \cdot (\mathbf{r}_f - \mathbf{r}_P), \quad (21)$$

$$\mathbf{k}_N = (\mathbf{I} - \mathbf{n}_f \mathbf{n}_f) \cdot (\mathbf{r}_N - \mathbf{r}_f). \quad (22)$$

The first term on the right hand side of Eq. (19) is treated implicitly, while the correction term is explicit. The cell-centre displacement gradient used for the calculation of the correction term in Eq. (19) is calculated using the vertex-based Gauss-Green method, which gives a gradient of second-order accuracy irrespective of the mesh quality. Here, the face-centre displacement is calculated by averaging corresponding vertex displacements.

The face-centre gradient needed for the evaluation of tensor \mathbf{q}_f in the first term on the right hand side of Eq. (16) is calculated separately in the normal and tangential directions. The face-centre displacement gradient in the normal direction is calculated using Eq. (19), while the face-centre gradient in the tangential direction is calculated by applying the vertex-based Gauss-Green method to a flat polygonal face. The vertex-based Gauss-Green gradient calculation procedure is described in Appendix A.

The approximation of the displacement gradient at the centres of the cell faces coinciding with the spatial domain boundary depends on the applied boundary conditions. In the case of

the Dirichlet boundary condition (specified displacement \mathbf{u}_b), the displacement gradient in the tangential direction is calculated in the same way as for the internal faces, while the gradient in the normal direction is calculated as follows:

$$\mathbf{n}_b \cdot (\nabla \mathbf{u})_b = \frac{\mathbf{u}_b - \mathbf{u}_P}{d_{bn}} - \frac{\mathbf{k}_b \cdot (\nabla \mathbf{u})_P}{d_{bn}}, \quad (23)$$

where the second term on the right hand side represents the non-orthogonal correction (see Fig.3).

For the boundary cell faces with specified traction, the displacement gradient in the normal direction is calculated using the constitutive relation as follows:

$$\mathbf{n}_b \cdot (\nabla \mathbf{u})_b = \frac{\mathbf{n}'_b \cdot \boldsymbol{\sigma}_b \frac{S'_b}{S_b} - \mathbf{n}_b \cdot \mathbf{q}_b}{2\mu + \lambda}, \quad (24)$$

where $\mathbf{n}'_b \cdot \boldsymbol{\sigma}_b$ is the Cauchy traction at the boundary face b , \mathbf{n}'_b is the normal, and S'_b is the area of the boundary face b in the current (deformed) configuration. Displacement \mathbf{u}_b at the traction boundary face is calculated using Eq. (23) with the displacement gradient in the normal direction calculated from Eq (24).

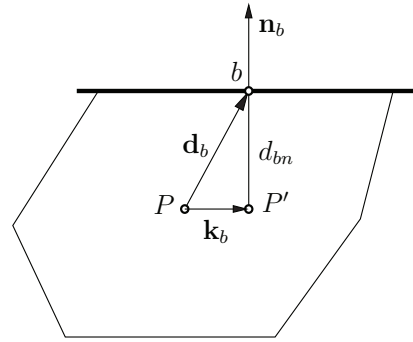


Fig. 3 Displacement gradient calculation for boundary condition implementation

The vertex displacements are reconstructed from the cell-centre displacements of the cells surrounding the vertex using the weighted least-squares method and linear fitting function. Description of the vertex displacement reconstruction procedure is given in Appendix B.

When Eqs. (17), (18), and (19) are substituted into Eq. (16), the fully discretised form of the linear momentum conservation law Eq. (8) can be written in the form of a linear algebraic equation, which for cell P reads:

$$a_P \mathbf{u}_P^{[m]} + \sum_N a_N \mathbf{u}_N^{[m]} = \mathbf{R}_P, \quad (25)$$

where the diagonal coefficient a_P , the neighbour coefficient a_N , and the source term \mathbf{R}_P are defined by the following expressions:

$$a_P = \frac{9\rho_{0P}V_P}{4\Delta t^2} + \sum_f (2\mu_f + \lambda_f) \frac{S_f}{d_{fn}}, \quad (26)$$

$$a_N = -(2\mu_f + \lambda_f) \frac{S_f}{d_{fn}}, \quad (27)$$

$$\begin{aligned}
\mathbf{R}_P = & \rho_{0P} V_P \left[\frac{3\mathbf{u}_P^{[m-1]}}{\Delta t^2} - \frac{3\mathbf{u}_P^{[m-2]}}{4\Delta t^2} + \frac{2}{\Delta t} \left(\frac{\partial \mathbf{u}}{\partial t} \right)_P^{[m-1]} - \frac{1}{2\Delta t} \left(\frac{\partial \mathbf{u}}{\partial t} \right)_P^{[m-2]} \right] \\
& - \sum_f (2\mu_f + \lambda_f) \frac{\mathbf{k}_N \cdot (\nabla \mathbf{u})_N^{[m]} - \mathbf{k}_P \cdot (\nabla \mathbf{u})_P^{[m]}}{d_{fn}} S_f \\
& + \sum_f \mathbf{n}_f \cdot \mathbf{q}_f^{[m]} S_f + \sum_f \rho_{0P} \mathbf{b}_P^{[m]} V_P.
\end{aligned} \tag{28}$$

Remark. *The Laplacian terms in the momentum and pressure equations of the fluid model and in the fluid mesh motion equation are discretised in the same way as the Laplacian term in the solid momentum equation.*

3.3. Solution procedure for fluid and solid models

The incompressible fluid flow model, which is discretised on a moving finite volume mesh, is solved for velocity and pressure using the PISO algorithm [18]. A detailed description of the fluid model solution procedure is given in [20].

For solution of the discretised solid model, Eq. (25) is assembled for all CVs in the computational mesh, resulting in the following system of linear algebraic equations:

$$[A] \cdot \{\mathbf{u}\} = \{\mathbf{R}\}, \tag{29}$$

where $[A]$ is a sparse matrix, with coefficients a_P on the diagonal and a_N off the diagonal, $\{\mathbf{u}\}$ is the solution vector consisting of displacements \mathbf{u}_P for all CVs and $\{\mathbf{R}\}$ is the right-hand side vector consisting of source terms \mathbf{R}_P for all CVs.

The system of equations Eq. (29) is solved using a segregated algorithm, where the three components of the displacement vector are temporarily decoupled and solved separately. Since non-linear and coupling terms depending on the unknown displacement vector are placed in the right-hand side vector, the system is solved in an iterative manner, where the right hand-side vector $\{\mathbf{R}\}$ is updated at the beginning of each outer iteration by using the displacement vector increment from the previous iteration. When the solution changes less than some pre-defined tolerance, the system is considered to be solved. This is performed for every time-step of the transient simulation.

The sparse matrix $[A]$ from Eq. (29) is symmetric and weakly diagonally dominant and the corresponding system of equations is solved using the preconditioned conjugate gradient iterative solver [39, 40]. There is no need to solve the system to a fine tolerance since the right-hand side vector is only an approximation based on the displacement vector increment from the previous iteration; reduction of the residuals by an order of magnitude normally suffices.

3.4. Solution procedure for fluid-structure interaction

As mentioned earlier, the partitioned approach is adopted for the fluid-structure interaction solution procedure, where the flow model and the structural model are solved separately using different solvers in the FV framework. The coupled fluid-structure interaction problem is decomposed using the Dirichlet-Neumann procedure, where the flow model is solved for a given velocity (or displacement) of the fluid-structure interface, while the structural model is solved for a given force exerted on the interface.

The equilibrium of the force and velocity (or displacement) on the fluid-structure interface is enforced in each time step using a strongly coupled procedure by performing iterations between the fluid and solid solvers. To this end, one can choose between the Gauss-Seidel iteration scheme, with fixed relaxation or with convergence acceleration using Aitken relaxation [41,

42], and the interface quasi-Newton procedure, with the approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS) [43, 5]. The Aitken relaxation procedure and the IQN-ILS procedure are preceded by two fixed-relaxation iterations.

The fluid-structure interaction solution procedure is summarised in Algorithm 1. The procedure is mostly self-explanatory, except perhaps for the interface residual calculation, defined as the difference between the solid side interface displacement, obtained by solving the structural model, and the fluid side interface displacement, used to move the fluid mesh before solving the fluid model:

$$\{\mathbf{r}\}_i^k = \{\mathbf{u}\}_{S,i}^k - \{\mathbf{u}\}_{F,i}^k, \quad (30)$$

where superscript k represents the iteration number, $\{\mathbf{u}\}_{F,i}$ is the vector consisting of vertex-displacements of the fluid side of the interface, and $\{\mathbf{u}\}_{S,i}$ is the vector consisting of displacements of the solid side of the interface mapped to the vertices of the fluid side of the interface by a vertex-interpolation, described in the next section. The calculation procedure for the vertex displacements of the fluid side of the interface, used to move the fluid mesh, depends on the selected fluid-structure iteration procedure. If the Gauss-Seidel-type procedure is used, the fluid side vertex displacements are calculated as follows:

$$\{\mathbf{u}\}_{F,i}^{k+1} = \{\mathbf{u}\}_{F,i}^k + \omega^{k+1} \{\mathbf{r}\}_i^k, \quad (31)$$

where ω^{k+1} is the fixed or the dynamic under-relaxation factor, for the fixed or the Aitken relaxation schemes, respectively. The calculation of $\{\mathbf{u}\}_{F,i}^{k+1}$ using the IQN-ILS procedure is described in [5].

At the end of each fluid-structure interaction iteration, the L^2 -norm of the interface residual vector is calculated and it is checked whether a converged solution is reached.

Algorithm 1 Fluid-structure interaction iterative solution procedure

- 1: Switch to the next time step.
 - 2: Predict interface displacement and calculate initial interface residual.
 - 3: Start the FSI strongly coupled iterative procedure.
 - 4: Switch to the next iteration.
 - 5: Calculate the vertex-displacements of the fluid side of the interface.
 - 6: Solve mesh motion equation.
 - 7: Move the fluid mesh.
 - 8: Solve the fluid model.
 - 9: Transfer the face-centre forces from the fluid to the solid side of the interface.
 - 10: Solve the structural model.
 - 11: Transfer the vertex-displacements from the solid to the fluid side of the interface.
 - 12: Calculate interface residual at the fluid side of the interface.
 - 13: **if** converged **then**
 - 14: Go to next time step (line 1)
 - 15: **else**
 - 16: Go to next iteration (line 4)
 - 17: **end if**
-

3.5. Interpolation at the fluid-structure interface

Traction (pressure and viscous forces) calculated at the boundary faces of the fluid side of the interface must be transferred to the boundary faces of the solid side of the interface; it is

applied as the boundary condition for the solid model. In general, meshes at the solid and fluid side of the interface are not conformal and an interpolation procedure between the models is required. To this end, two different interpolation techniques are used:

1. *face-interpolation procedure* for the interpolation from the boundary cell faces of the fluid side of the interface to the boundary cell faces of the solid side of the interface,
2. *vertex-interpolation procedure* for the interpolation of the vertex-displacement field from the solid side of the interface to the vertices of the fluid side of the interface.

3.5.1. Face-interpolation

Interpolation of face-centre fields between the fluid and solid side of the interface is performed using the Generalised Grid Interface (GGI) interpolation [44]. A face-centre value at the solid side of the interface is calculated using face-centre values at the fluid side of the interface as follows:

$$\phi_{S,j} = \sum_k w_{j,k} \phi_{F,k}, \quad (32)$$

where $\phi_{S,j}$ is the value of the field at the centre of the solid face j and $\phi_{F,k}$ is the value of the field at the centre of the fluid face k . Summation is performed over all fluid side faces k which have an intersection with the solid side face j . The weighting factor $w_{j,k}$ for the fluid face k is defined as

$$w_{j,k} = \frac{S_{S,j} \cap S_{F,k}}{S_{S,j}}, \quad (33)$$

where $S_{S,j} \cap S_{F,k}$ is the surface intersection area between the solid face j and the fluid face k and $S_{S,j}$ is the area of the solid face j .

The surface intersection area is calculated using the Sutherland-Hodgman algorithm [45] which is fast, robust, and can operate on any convex n -sided polygon. The face neighbourhood determination is performed using the Axis Aligned Bounding Box quick rejection test [46] and the Separating Axis Theorem algorithm [47]. The face interpolation procedure is conservative in a sense that it preserves the total force at the two sides of the interface.

3.5.2. Vertex-interpolation

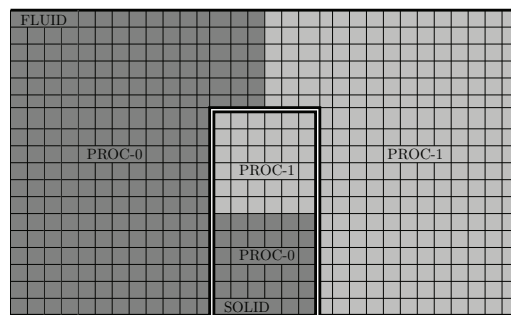
Interpolation of vertex displacements from the solid to the fluid side of the interface is performed using the following procedure:

1. Polygonal faces at the solid side of the interface are decomposed into triangles using an additional central point (centroid of the polygonal face).
2. Vertices at the fluid side of the interface are projected to the nearest triangle at the solid side of the interface.
3. Displacement values at the projection points are calculated by linear interpolation using known displacements at the triangle vertices.

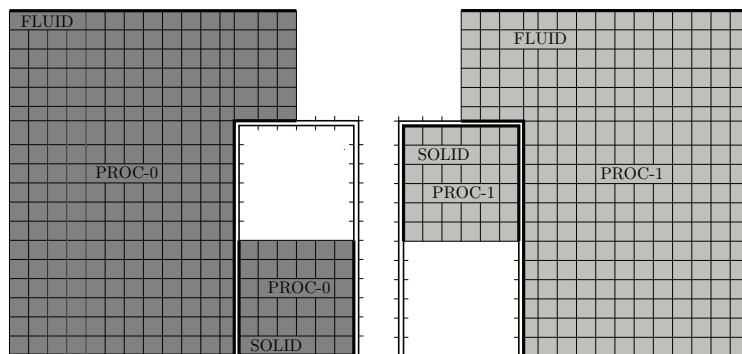
Searching the nearest triangle for each vertex at the fluid side of the interface is substantially accelerated using the neighbourhood determined by the GGI interpolation.

3.6. Parallelisation

Parallel computing in OpenFOAM is performed using the domain decomposition approach. Fluid and solid meshes and fields are split into a number of sub-domains, which are allocated to separate processors (see Fig. 4(a)). Solvers are run in parallel on separate sub-domains and communication between processors is performed using the Message Passing Interface communication protocol [48]. In order to allow independent decomposition of the fluid and solid models in the FSI simulation, the *global face zone concept* is introduced, where the entire fluid and solid sides of the interface (all interface points and faces) are present on all processors (see Fig. 4(b) and (c)). Prior to the exchange of data between fluid and solid models, interface fields (point or face-centre fields) are reconstructed on all processors so that interpolation can be performed on each processor independently. Interface field reconstruction is performed using the gather-scatter procedure. Here, each processor sends its part of the interface field to the master (zero) processor, where the entire interface field is reconstructed and sent back to the remaining processors.



(a) Fluid and solid domains decomposed independently into two sub-domains.



(b) Fluid and solid sub-domains on processor 0. (c) Fluid and solid sub-domains on processor 1.

Fig. 4 Parallel fluid-structure interaction computed using the domain decomposition and the global interface face zones to exchange information between the fluid and the solid at the interface.

4. Numerical results

4.1. Flat plate with a circular hole subjected to uniform tension

This case is used to investigate the spatial accuracy of the structural part of the numerical model proposed in the paper. The model is compared to the one given by Demirdžić and Muzaferija [22], which may still be considered as the state-of-the-art model for numerical stress analysis based on the cell-centred (collocated) finite volume method supporting arbitrary poly-

hedral meshes. It is also compared to the numerical model given by Jasak and Weller [24], which can be considered as an OpenFOAM implementation of the model given by Demirdžić and Muzaferija.

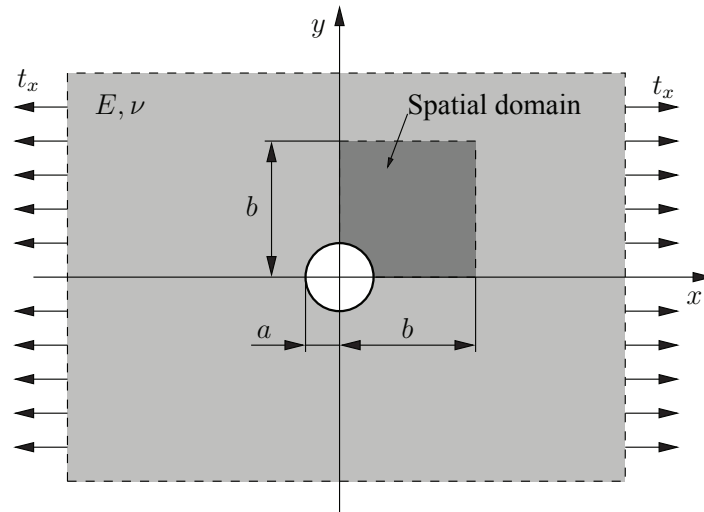


Fig. 5 Geometry of the spatial computational domain for the flat plate with a circular hole ($a = 0.5$ m, $b = 2$ m, $E = 1 \times 10^7$ Pa, $\nu = 0.3$, $t_x = 10$ kPa).

A plate with a circular hole in its centre is loaded by uniform tension in one direction (Fig. 5). The analytical solution to this plane stress linear elastic problem, obtained for an infinitely large plate, can be found in [49]. Due to the symmetry of the problem, only a quarter of the plate is modelled, as shown in the figure (darker area). Following [50], the exact solution corresponding to $t_x = 10$ kPa is prescribed on all traction specified boundaries (top and right dashed boundaries) to remove the effects of the finite size of the computational domain.

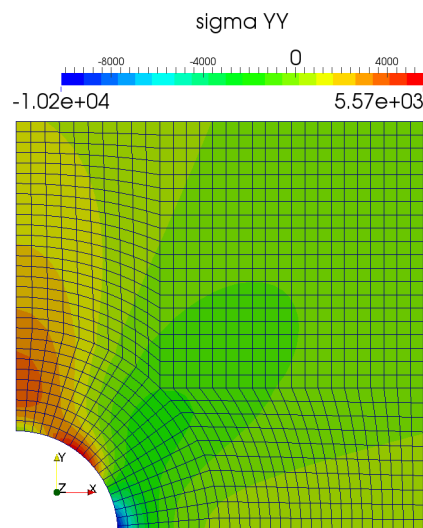


Fig. 6 Discretised spatial domain for a plate with a hole case with the yy -component of Cauchy stress.

Grid sensitivity study is performed by simulations on four systematically refined grids, consisting of 1 000, 4 000, 16 000, and 64 000 quadrilateral cells¹. The coarsest grid, consisting of

¹In reality, the mesh is 3-D and consists of one layer of prismatic cells with quadrilateral base.

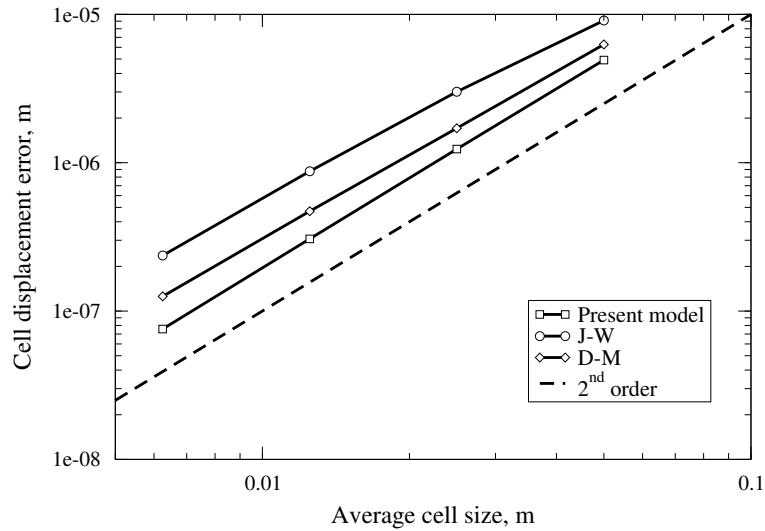


Fig. 7 Maximum displacement error as a function of average cell size for the present model and the models given by Demirdzic and Muzaferija [22] (D-M) and Jasak and Weller [24] (J-W).

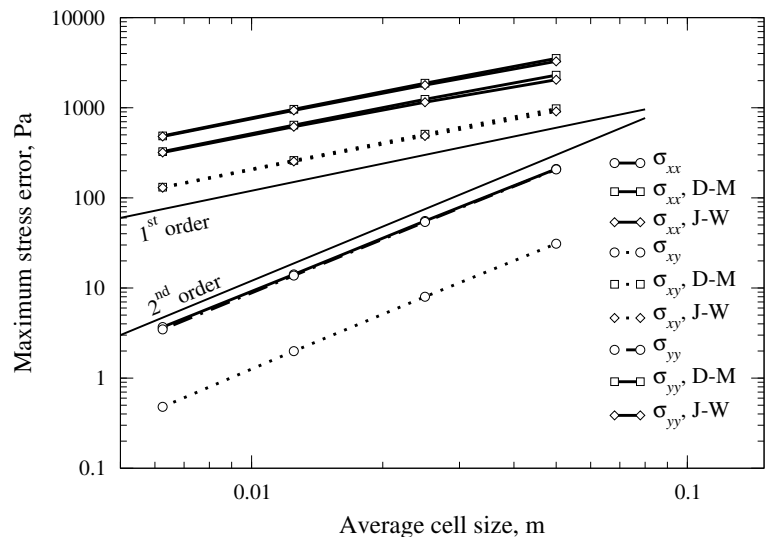


Fig. 8 Maximum stress error as a function of average cell size for the present model and the models given by Demirdzic and Muzaferija [22] (D-M) and Jasak and Weller [24] (J-W).

1 000 cells, is shown in Fig. 6 together with the yy -component of the Cauchy stress distribution. For each grid, a maximum error is calculated for the cell-centre displacement field and the face-centre Cauchy stress field at the hole surface with respect to the corresponding analytical solution.

Figure 7 shows maximum error of the cell-centre displacement field as a function of average cell size, while the maximum error of the face-centre Cauchy stress components at the hole surface is shown in Fig. 8. It can be noticed that the cell-centre displacement error reduces at a second-order rate for all three numerical models considered, with the smallest error for the present model (it also seems that the present model follows the second-order trend best). It is interesting to notice that the numerical model given by Jasak and Weller gives a greater displacement error than the model by Demirdzic and Muzaferija. This can be attributed to the inconsistency between the discretisation of the implicit and explicit parts of the traction force at the cell boundary in the Jasak-Weller model. The maximum stress error at the hole

surface reduces at the second-order rate for the present model, but at the first-order rate for the Demidzic-Muzaferija and Jasak-Weller models. This can be attributed to the second-order approximation of the displacement gradient in the tangential direction to the boundary cell face obtained by applying the vertex-based Gauss-Green method at the boundary cell faces, although the displacement gradient in the normal direction is approximated with first-order accuracy. The present numerical model also gives a substantially smaller stress error for the same grid resolution.

4.2. Channel flow over a cavity with a flexible bottom

The first FSI case considered is laminar flow of incompressible fluid through a two-dimensional channel with a cavity in the bottom wall of the channel. This case is proposed in [51] and it is extended for the purpose of the presented FSI solver validation by adding a flexible wall at the cavity bottom. The geometry of the fluid and solid parts are shown in Fig. 9. The height of the channel for all performed calculations is $H = 1$ m.

A fluid with a density of 1 kg/m^3 and a kinematic viscosity of $0.01 \text{ m}^2/\text{s}$ enters the channel from the left-hand side with a parabolic velocity profile. The mean inlet velocity is 1 m/s , corresponding to a Reynolds number of 100 with respect to the channel height. A constant pressure is imposed at the outlet of the channel and a no-slip boundary condition is applied to the channel walls. The elastic plate has a density of 1000 kg/m^3 , Young's modulus of 500 N/m^2 , Poisson's ratio of 0.4 and its deformation is described by the Saint Venant-Kirchhoff constitutive model with the plane strain assumption. The coupling between the fluid and the solid is performed using the Gauss-Seidel iteration scheme with the Aitken relaxation.

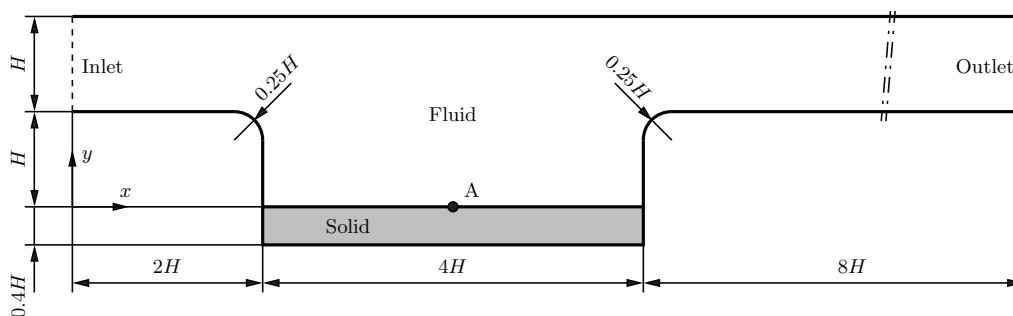


Fig. 9 Geometry of the spatial computational domain for the channel flow over a cavity with a flexible bottom.

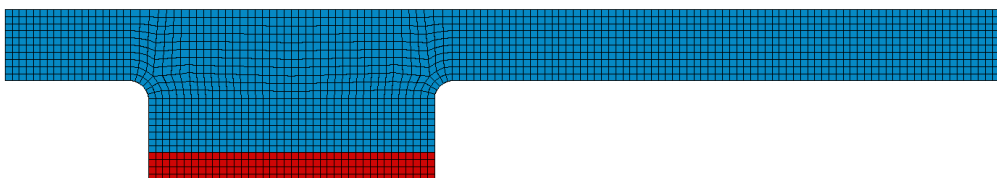


Fig. 10 Discretised spatial domain for the channel flow over a cavity with flexible bottom.

Spatial accuracy is demonstrated by calculating the steady-state solution with different finite volume mesh resolutions. Four block-structured meshes are used with cell sizes of 0.1 m, 0.05 m, 0.025 m, and 0.0125 m (Fig. 10 shows the coarsest mesh). The steady state velocity field and equivalent stress in the plate, calculated on the third mesh, can be seen in Fig. 11. Convergence of the y -component of the displacement at the point A (see Fig. 9) and the y -component of the total force at the interface with regard to the cell size is shown in Fig. 12(a)

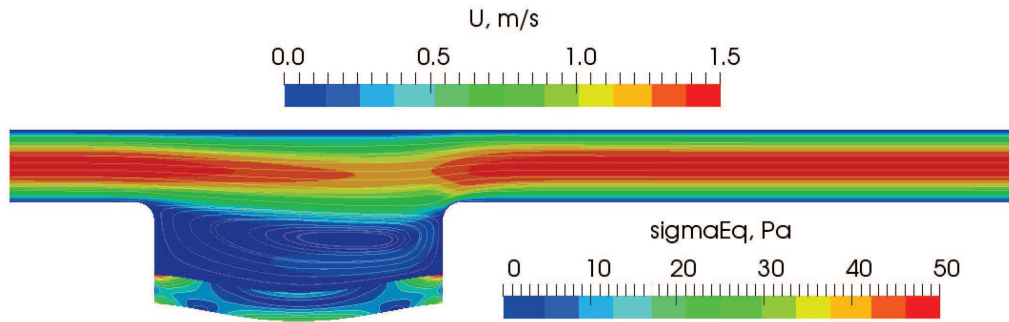
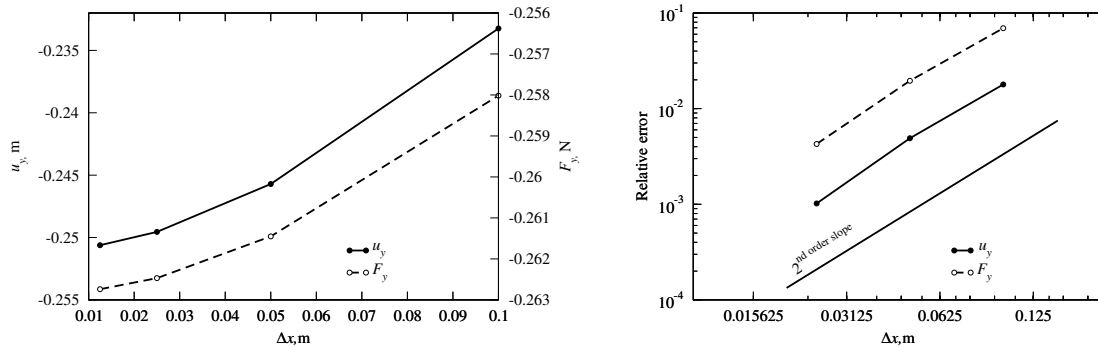


Fig. 11 Velocity field in the fluid and equivalent stress in the solid part of the spatial computational domain at steady state for the channel flow over a cavity with a flexible bottom.

with the corresponding relative errors (Fig. 12(b)). The relative error is calculated with respect to the results from the finest mesh. The order of spatial accuracy is calculated as [52]:

$$p = \frac{\ln \left[\frac{\phi_3 - \phi_2}{\phi_2 - \phi_1} \right]}{\ln(r)} \quad (34)$$

where ϕ_1 , ϕ_2 , and ϕ_3 are the numerical solutions with the three finest meshes, refined using the constant refinement ratio $r = \Delta x_2 / \Delta x_1 = \Delta x_3 / \Delta x_2 = 2$. The order of spatial accuracy calculated using the y -component of displacement is 1.85, whereas it is 1.91 when calculated using the y -component of total force; this is in good agreement with the expected second-order spatial accuracy of the fluid and solid models.



(a) Displacement and force as a function of cell size

(b) Relative error as a function of cell size

Fig. 12 Calculating spatial accuracy for the channel flow over a cavity with a flexible bottom – displacement of point A and force at the plate.

4.3. Elastic plate behind a rigid cylinder

Elastic plate behind a rigid cylinder is a well known benchmark FSI case documented in [53]. The geometry (Fig. 13) consists of a horizontal channel of 0.41 m in height and 2.5 m in length, containing a rigid cylinder with a radius of 0.05 m; the centre of the cylinder is positioned 0.2 m away from the bottom and inlet (left) boundaries of the channel. An elastic plate of 0.35 m in length and 0.02 m in height is attached to the right-hand side of the rigid cylinder.

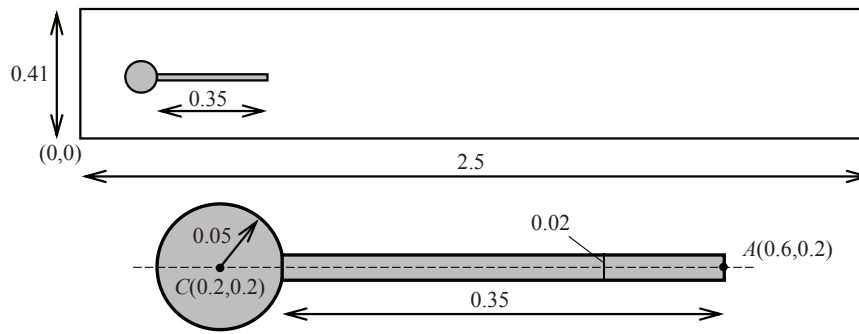


Fig. 13 Computational domain with a structural detail for the elastic plate case. All dimensions are in m.

Fluid enters the channel from the left-hand side with a parabolic velocity profile. A constant pressure is imposed at the outlet of the channel and a no-slip boundary condition is applied on the walls. The fluid flow is assumed to be laminar. Deformation of the elastic plate is described by the Saint Venant-Kirchhoff constitutive model with the plane strain assumption. Unsteady variants of this case, designated as FSI2 and FSI3 in [53], are considered. Properties of the fluid and elastic plate material are given in Table 1 together with the mean inlet velocity.

Table 1 Fluid and solid properties and mean inlet velocity \bar{u} for the elastic plate behind a rigid cylinder: ρ_F is the fluid density, ν_F is the fluid kinematic viscosity, ρ_S is the solid density, E_S is Young's modulus and ν_S is Poisson's ratio.

	FSI2	FSI3
$\rho_F, \text{kg/m}^3$	1000	1000
$\nu_F, \text{m}^2/\text{s}$	0.001	0.001
$\bar{u}, \text{m/s}$	1	2
$\rho_S, \text{kg/m}^3$	10000	1000
E_S, Pa	1.4×10^6	5.6×10^6
ν_S	0.4	0.4

Figure 14 shows a section of the discretised spatial domain. The fluid part of the mesh consists of 21 344 quadrilateral finite volumes, while the solid part consists of 328 finite volumes. The numerical solution is obtained with a time step size $\Delta t = 1.5 \times 10^{-3}$ s for the FSI2 case and $\Delta t = 0.75 \times 10^{-3}$ s for the FSI3 case.

In order to quickly reach the periodic motion of the plate, the inlet velocity profile is applied without a gradual increase in the mean velocity and the coupling between the fluid and the solid is activated after 2 s. The coupling between the fluid and the solid is performed using the IQN-ILS procedure. The residual $\|\mathbf{r}\|_2$ is reduced in each time step by six orders of magnitude with respect to its maximum value.

Figure 15 shows simulation snapshots for the FSI2 case at the instant of time when the plate tip point (A) is at its highest position. Temporal variation in the plate tip point displacement, after the periodic solution has been reached, is shown in Figs. 16 and 18. On the other hand, temporal variation in the total force exerted on the plate and cylinder in the same time frame is shown in Figs. 17 and 19. A quantitative comparison with the corresponding benchmark solutions is given in Table 2. Mean values and amplitudes are calculated using the maximum and the minimum values after the periodic solution has been reached (around 7 s for the FSI2

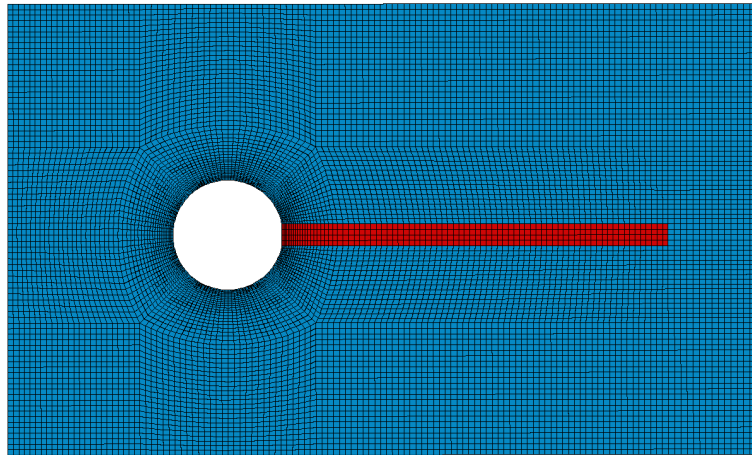


Fig. 14 Discretised spatial domain for the elastic plate case.

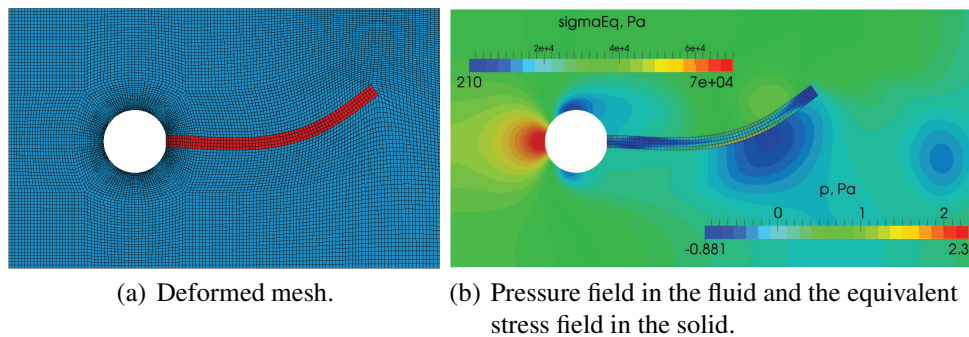


Fig. 15 FSI2 case – plate tip point at its highest position.

case and 4 s for the FSI3 case), where frequencies are calculated using the Fast Fourier Transform algorithm. The difference between the calculated and the benchmark results is around 3% in average for the amplitude and frequency of force and displacement. The relative difference for the mean value of the force y -component goes up to 40%, which can be attributed to the difficulty in calculating the mean value in the case when it is close to zero.

4.4. Wave propagation in an elastic tube

Wave propagation in an elastic tube is a standard FSI case intended to demonstrate the capability of the numerical model to predict blood flow in large arteries [5]. The spatial computational domain consists of a straight flexible tube with a radius (r) of 0.005 m, a length (L) of 0.05 m, and a thickness (t) of 0.001 m. The tube wall is the Saint Venant-Kirchhoff material with a density (ρ_S) of 1200 kg/m³, Young's modulus (E) of 3×10^5 N/m², and Poisson's ratio (ν_S) of 0.3. The tube is clamped in all directions at the inlet and outlet. The fluid is incompressible, with a density (ρ_F) of 1000 kg/m³, and a dynamic viscosity (ν_F) of 0.003 Pa.s. Both the fluid and the structure are initially at rest. During the first 0.003 s, a uniform overpressure of 1333.2 N/m² is applied at the inlet.

Figure 20 shows the spatial domain discretised by a hexahedral finite volume mesh. The fluid part of the mesh consists of 449 600 finite volumes, while the solid part consists of 288 000 finite volumes. The numerical solution presented here is obtained using the time step size $\Delta t = 5 \times 10^{-5}$ s. FSI coupling in this case is performed using the IQN-ILS procedure.

Figure 21 shows the velocity field both in the fluid and solid parts of the spatial domain at time instant 0.005 s, while Fig. 22 shows the pressure field in the fluid part and the equivalent

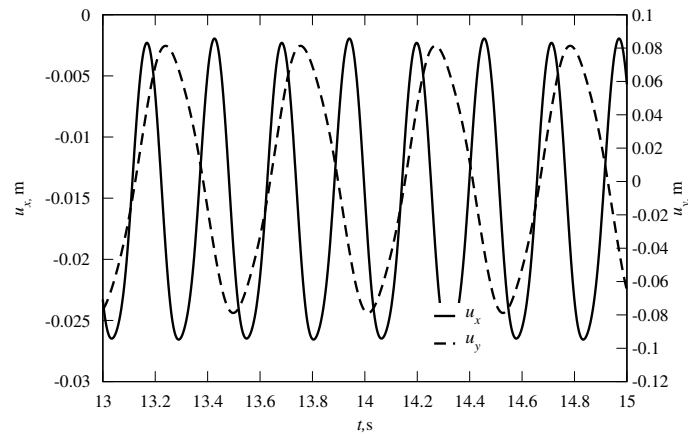


Fig. 16 Displacement of the plate tip point A for the elastic plate behind a rigid cylinder case (FSI2).

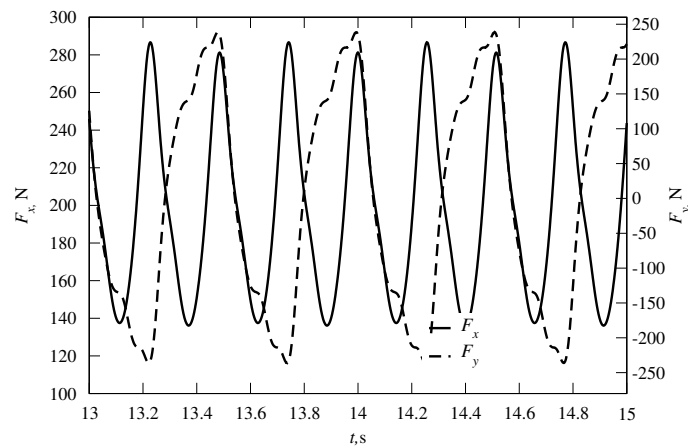


Fig. 17 Force on the plate and the cylinder for the elastic plate behind a rigid cylinder case (FSI2).

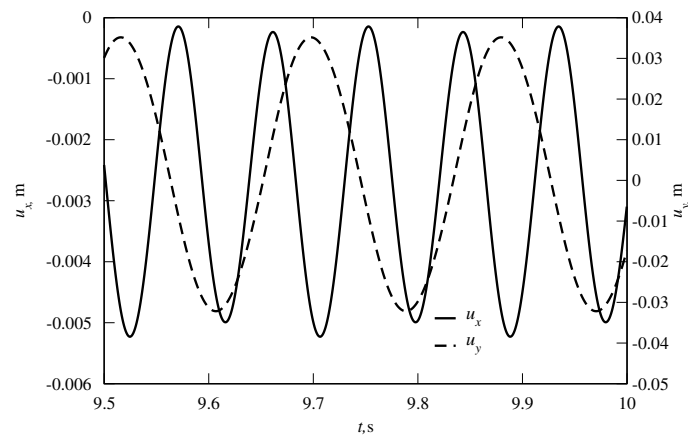


Fig. 18 Displacement of the plate tip point A for the elastic plate behind a rigid cylinder case (FSI3).

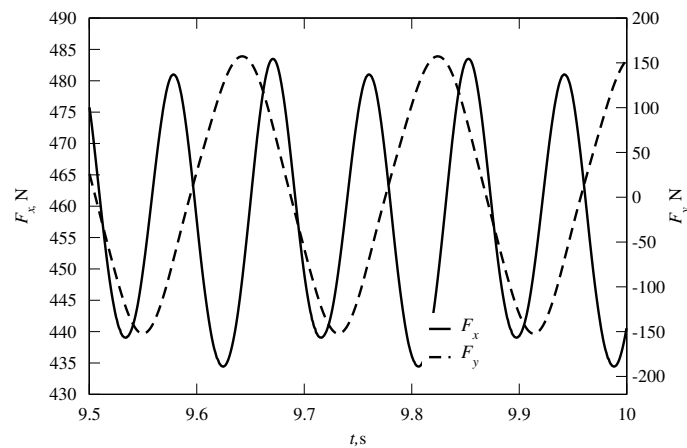


Fig. 19 Force on the plate and the cylinder for the elastic plate behind a rigid cylinder case (FSI3).

Table 2 Displacement and force for the elastic plate behind a rigid cylinder case. The values are given in the same format as in the benchmark paper [53] (mean \pm amplitude [frequency]).

	$u_x \times 10^{-3}$ [m]	$u_y \times 10^{-3}$ [m]
FSI2		
Benchmark	$-14.58 \pm 12.44[3.8]$	$1.23 \pm 80.6[2.0]$
Calculated	$-14.26 \pm 12.34[3.9]$	$1.22 \pm 80.2[1.95]$
FSI3		
Benchmark	$-2.69 \pm 2.53[10.9]$	$1.48 \pm 34.38[5.3]$
Calculated	$-2.72 \pm 2.58[11.07]$	$1.67 \pm 33.84[5.53]$
	F_x [N]	F_y [N]
FSI2		
Benchmark	$208.83 \pm 73.75[3.8]$	$0.88 \pm 234.2[2.0]$
Calculated	$211.34 \pm 75.59[3.9]$	$1.23 \pm 238.35[1.95]$
FSI3		
Benchmark	$457.3 \pm 22.66[10.9]$	$2.22 \pm 149.78[5.3]$
Calculated	$459.18 \pm 24.86[11.07]$	$1.59 \pm 155.9[5.53]$

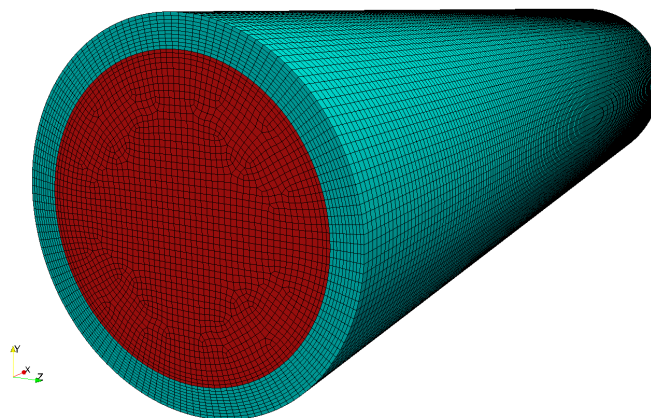


Fig. 20 Discretised spatial domain for the elastic tube case.

stress in the solid part of the spatial domain at the same time instant.

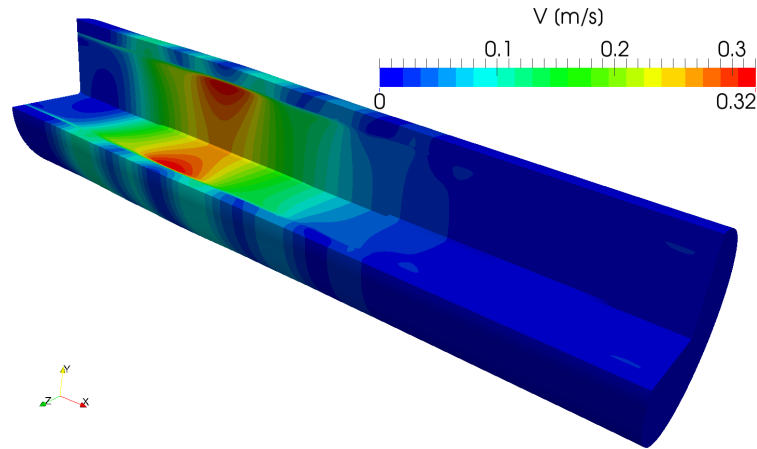


Fig. 21 The field of velocity magnitude in the fluid and the solid part of the spatial computation domain at the time instant 0.005 s for the elastic tube case.

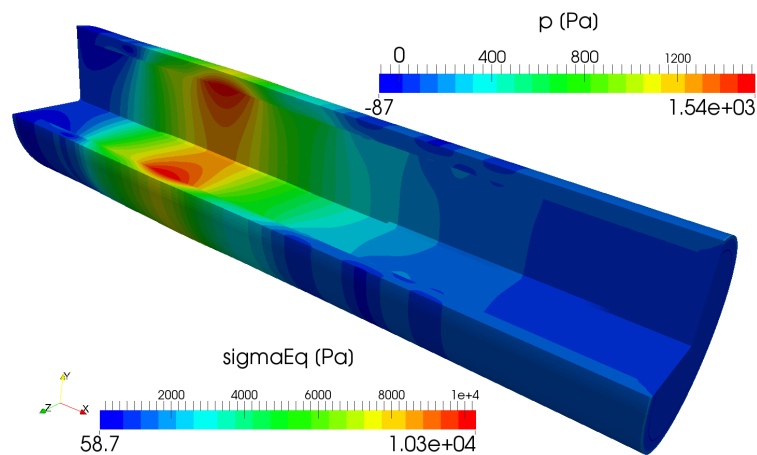


Fig. 22 Pressure field in the fluid and the equivalent stress in the solid part of the spatial computational domain at the time instant 0.005 s for the elastic tube case.

Pressure and axial velocity histories along the pipe axis for a few time instants are shown in Figs. 23 and 24. Using the procedure described in [17], one can obtain the simulated pressure wave speed of 4.54 m/s. The analytical solution of the pressure wave speed, c_F , can be found in [54]; for an incompressible fluid it reads:

$$c_F = \sqrt{\frac{Et}{2\rho_F r} \left[\frac{t}{r}(1 + \nu_S) + \frac{2r}{2r + t} \right]^{-1}} \quad (35)$$

Taking into account axial stress waves in the tube wall, the final expression for the analytical wave speed can be calculated as [17]:

$$\tilde{c}_F = c_F \sqrt{1 - \frac{\nu_S^2}{1 - \frac{t\rho_F}{2r\rho_S}}} \quad (36)$$

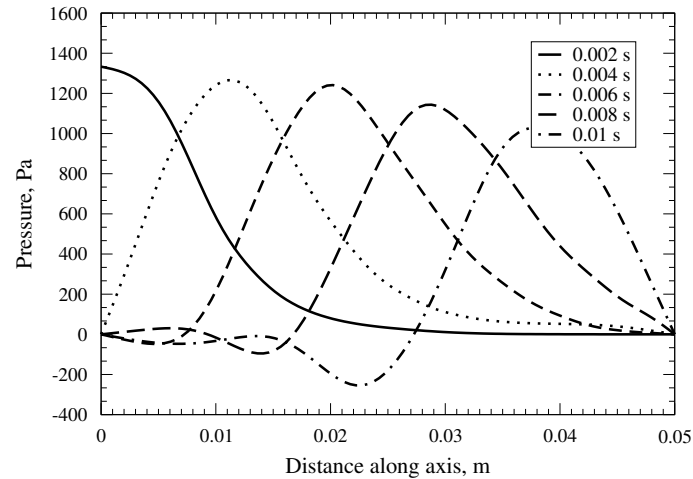


Fig. 23 Pressure variation along the tube axis as a function of time for the elastic tube case.

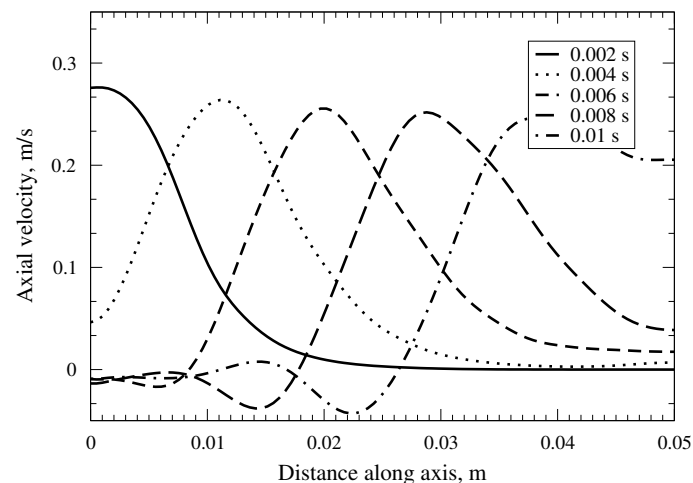


Fig. 24 Axial velocity variation along the tube axis as a function of time for the elastic tube case.

Using the case data, a value of 4.81 m/s can be obtained, which is in good agreement with simulation.

In order to evaluate the temporal accuracy of the method, the calculation is carried out for three different time step sizes: 1×10^{-4} s, 5×10^{-5} s, and 2.5×10^{-5} s, and the radial displacement is monitored at the inner side of the tube midsection. Results are shown in Fig. 25. Temporal accuracy is evaluated for the time instant when the radial displacement reaches its maximum. The order of temporal accuracy is calculated using Eq. (34) and its value is 3.11, *i.e.* above the theoretical order of accuracy for the backward temporal discretisation scheme.

In order to estimate the efficiency of parallel calculation, the same calculation is performed on a various number of processors and execution times are compared to the execution time on a single processor. Figure 26 shows the simulation speed-up as a function of the number of processors. Parallel performance is firstly tested for the fluid and the solid part separately, where one can see that the solid solver scales better than the fluid solver. By analysing the parallel performance of the FSI solver, it may be concluded that parallel scaling is actually limited by the fluid solver.

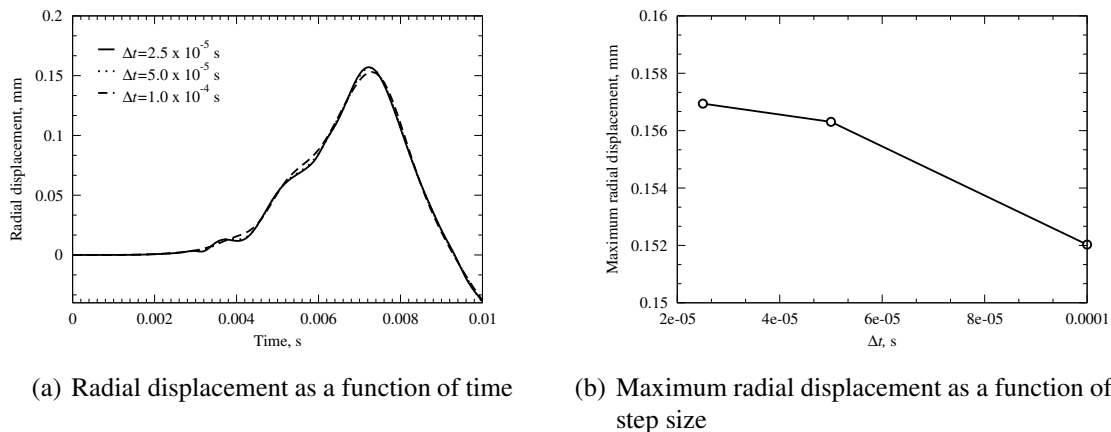


Fig. 25 Calculating temporal accuracy – the inner wall at the midsection of the tube.

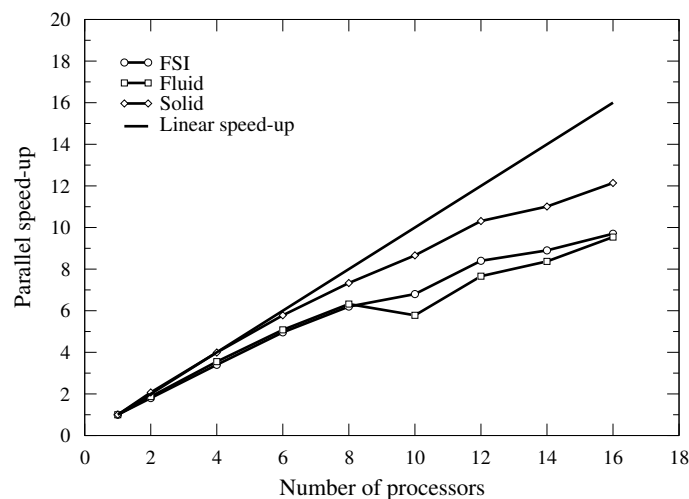


Fig. 26 Parallel calculation performance.

4.5. Channel flow over an elastic thick plate

Channel flow over an elastic thick plate is a steady-state three-dimensional FSI case, used in this study to show the performance of the proposed FSI solver on unstructured polyhedral meshes. The case, originally proposed by Richter [55], consists of an elastic thick plate attached to the bottom surface of a rectangular channel. The geometry of the spatial domain is given in Fig. 27. Due to symmetry, only half of the spatial domain is considered. An incompressible viscous fluid with a density of 1000 kg/m^3 and a kinematic viscosity of $0.001 \text{ m}^2/\text{s}$ enters the channel from the left-hand side with a parabolic velocity profile. The peak inlet velocity is 0.2 m/s , corresponding to a Reynolds number of 40 with respect to the plate height ($h = 0.2 \text{ m}$). The peak inlet velocity is gradually increased from zero at $t = 0 \text{ s}$ to its maximum value at $t = 4 \text{ s}$ using the following transition function: $0.2 [1 - \cos(\pi t/4)] / 2$. A constant pressure is applied at the outlet of the channel and a no-slip boundary condition is applied to the channel walls. The elastic plate has a density of 1000 kg/m^3 , Young's modulus of $1.4 \times 10^6 \text{ N/m}^2$ (a shear modulus of $5 \times 10^5 \text{ N/m}^2$), Poisson's ratio of 0.4, and its deformation is described by the Saint Venant-Kirchhoff constitutive model.

The spatial computational domain is discretised using an arbitrary polyhedral mesh consist-

ing of 273 539 cells in the fluid part of the domain and 6 661 cells in the solid part of the domain. The structure of the mesh at the bottom wall, symmetry plane, and fluid-solid interface can be seen in Fig. 28 together with the velocity and displacement fields at a steady state. These results are obtained in the same way as in [7], where an increased peak inlet velocity (0.3 m/s) and a reduced Young's modulus (10^4 N/m²) are used in order to show that the presented numerical model can handle large solid displacements.

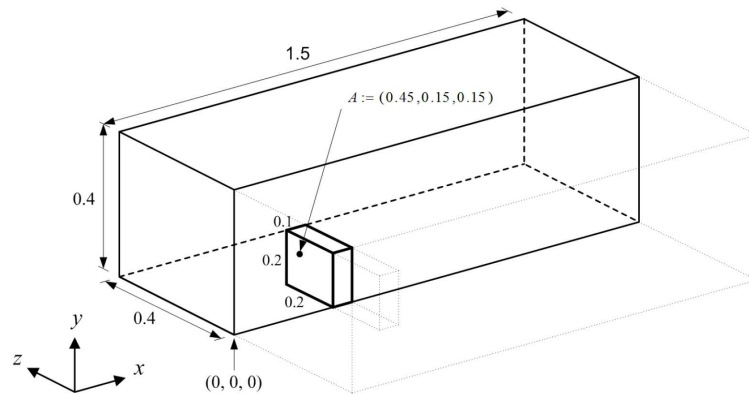


Fig. 27 Geometry of the spatial domain for the channel flow over an elastic thick plate case [55]. All dimensions are in m.

Displacement of the interface point *A* (see Fig. 27) and the total fluid force acting on the plate are shown in Tab. 3 together with the corresponding benchmark solutions [55].

Table 3 Comparison of the calculated and benchmark [55] displacements of the interface point *A* and the total forces on the plate.

	u_x [m]	u_y [m]	F_x [N]	F_y [N]
Calculated	5.93×10^{-5}	2.40×10^{-5}	1.31	0.1055
Benchmark	5.95×10^{-5}	-	1.33	-

To demonstrate the temporal accuracy, the simulation is performed with four different time-step sizes: 0.005 s, 0.0025 s, 0.00125 s, and 0.000625 s. The accuracy analysis is carried out at the time instant $t = 0.08$ s for the x -component of the displacement of point *A* (see Fig. 27). It has to be noted that in this analysis the same transition function was used, but the peak velocity was set to be reached after 0.1 s. Figure 29(a) shows the convergence of the solution with the time-step size. The order of temporal accuracy is calculated using Eq. (34) with the three finest time-step sizes and the refinement ratio $r = \Delta t_1/\Delta t_2 = \Delta t_2/\Delta t_3 = 2$. The calculated temporal accuracy is 2, corresponding to the formal order of accuracy of the backward temporal discretisation scheme. This is also graphically demonstrated in Fig. 29(b), where the relative error is calculated against the results with the smallest time-step size.

5. Conclusions

A self-contained parallel fluid-structure interaction solver based on the FV discretisation method and strongly coupled partitioned approach is presented in this paper. Both the fluid and solid models are discretised in space using the second-order accurate cell-centred finite volume method, while the numerical integration of the models in time is performed using a second-order accurate implicit method. Details of the gradient calculation and the vertex field value

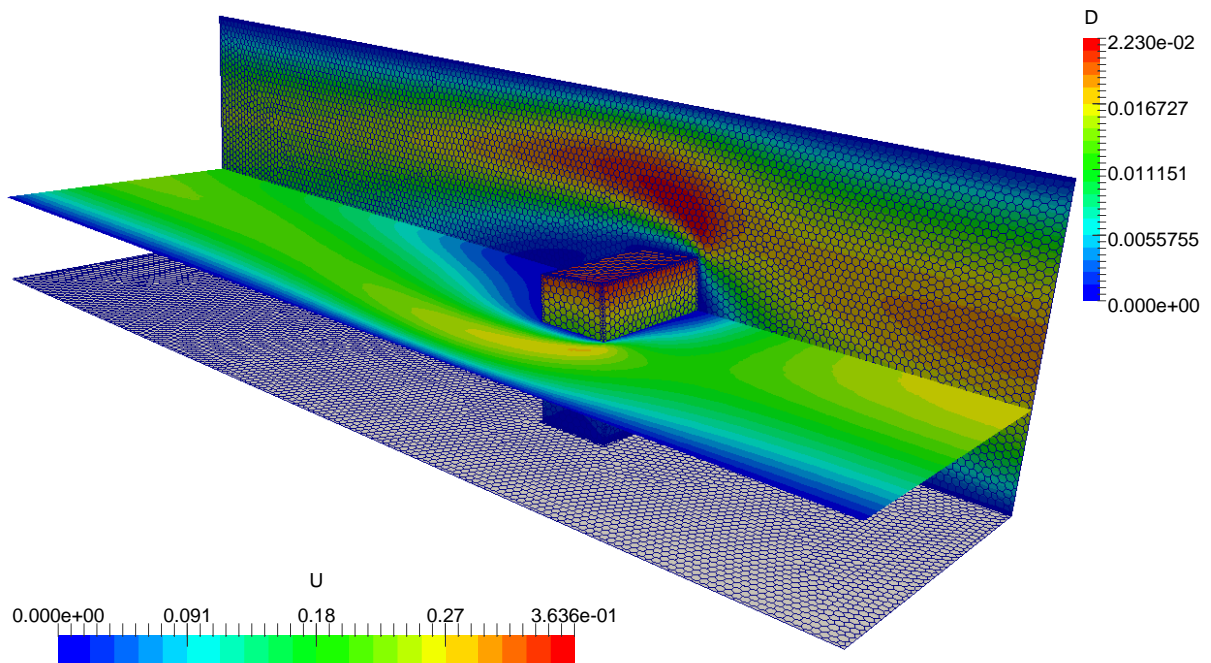
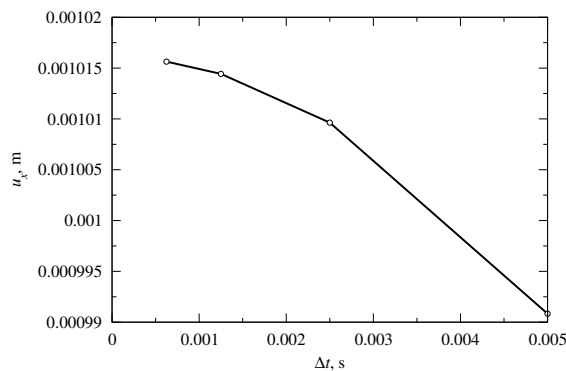
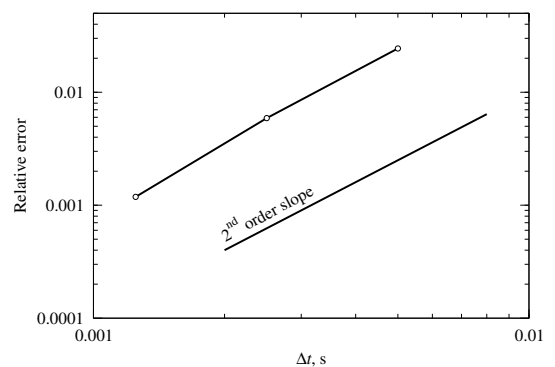


Fig. 28 Fluid velocity magnitude field and solid displacement magnitude field for the channel flow over an elastic thick plate case. In order to obtain a larger beam deformation, an increased peak inlet velocity of 0.3 m/s and reduced Young’s modulus of 10^4 N/m² have been taken in this case. The calculated displacement of point *A* in a steady state is (0.01463, 0.005, -0.000447).



(a) Calculated displacement of point *A* as a function of time-step size



(b) Relative error as a function of time-step size

Fig. 29 Calculating temporal accuracy for the channel flow over an elastic thick plate case.

reconstruction on a polyhedral mesh with special attention paid to parallel computing are also given.

The solver has been tested in four FSI benchmark cases, where the spatial and the temporal accuracy for the solid part together with the parallel computing performance are evaluated (accuracy analysis for the fluid part has been omitted here and can be found in [20]). However, the spatial accuracy is investigated in more detail in the standalone stress analysis case, where the performance of the presented stress model is compared with the existing models in the FV stress analysis. It has been demonstrated that the present model is more accurate and follows the second-order trend better than the existing models when both the displacement and stress fields are analysed. In addition, the spatial accuracy of the solid part has been analysed in the channel flow FSI case, with the overall accuracy corresponding well to the employed second-order accurate scheme.

In terms of temporal accuracy of the solid part, the order of accuracy has been shown to be the same or better than the employed second-order accurate backward scheme for the two cases analysed (see Sections 4.4 and 4.5). The better performance has been demonstrated in the elastic tube case, where the influence of other frequency modes may have caused discrepancies.

A quantitative analysis of two well known FSI cases has been presented, namely, the elastic plate behind a rigid cylinder (Section 4.3) and the channel flow over an elastic thick plate (Section 4.5), where the available results are compared. The latter case has been used to demonstrate the performance of the solver on unstructured polyhedral meshes. In general, excellent agreement has been achieved. It has also been shown that the parallel efficiency is not reduced by introducing an additional parallel exchange of information at the fluid-solid interface but only limited by the parallel efficiency of the fluid solver.

Acknowledgments

The authors wish to acknowledge the support and provision of computational facilities received from the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC).

A. Vertex-based Gauss-Green gradient calculation procedure

The field gradient $\nabla\phi$ at the centre of the polyhedral finite volumes (cells) and the surface field gradient $\nabla_s\phi$ at the centre of the polygonal faces are calculated using the Gauss theorem and the cell-vertex field values. At this point, it is assumed that the field values at the cell vertices are known. The procedure for their calculation using the cell-centre field values is presented in Appendix B.

For the cell-centre gradient calculation, each polygonal face is decomposed into triangles using the average central point (point c in Fig. 1) and the gradient is calculated using the discretised Gauss-Green theorem as follows:

$$(\nabla\phi)_P = \frac{1}{V_P} \sum_{\tau} \mathbf{n}_{\tau} \phi_{\tau} S_{\tau}, \quad (37)$$

where S_{τ} is the triangle area, \mathbf{n}_{τ} is the triangle unit normal vector, and the summation is performed over all triangles enclosing the considered finite volume (cell). The field value ϕ_{τ} at the triangle centre is calculated as an average of the values at the triangle vertices and the field value at the central triangle vertex (f or c in Fig. 1) is calculated as an average of the corresponding polygonal face vertex field values. The volume V_P of the polyhedral cell is also calculated using the Gauss-Green formula:

$$V_P = \frac{1}{3} \sum_{\tau} \mathbf{n}_{\tau} \cdot \mathbf{r}_{\tau} S_{\tau}, \quad (38)$$

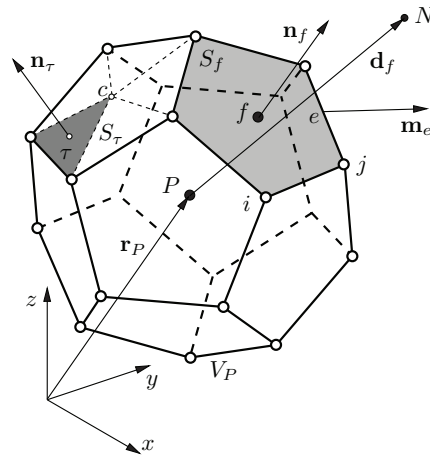


Fig. 1 Polyhedra finite volume (cell) bounded by polygonal faces.

where \mathbf{r}_τ is the position vector of the triangle centroid.

The surface field gradient at the polygonal face centre is calculated by using the Gauss-Green theorem discretised on a flat convex polygonal face f as follows:

$$(\nabla_s \phi)_f = \frac{1}{S_f} \sum_e \mathbf{m}_e \phi_e L_e, \quad (39)$$

where S_f is the face area, L_e is the face edge length, \mathbf{m}_e is the face edge unit bi-normal, and ϕ_e is the field value at the face edge centre calculated as an average of the end points field values (see [20] for more details).

B. Vertex field value reconstruction procedure

Field values at cell vertices are calculated from the cell-centre values using the weighted least squares method with the linear interpolation function. Without loss of generality, the interpolation procedure will be explained on a two-dimensional mesh, shown in Fig. 1, where interpolation stencils for the internal (i) and the boundary (b) vertices are given. The interpolation stencil of the internal vertex i consists of all cells sharing the vertex, whereas in the case of boundary vertex b , the boundary faces sharing the corresponding vertex are also included into the stencil. The following linear interpolation function is assumed in the neighbourhood of each vertex i (the same is valid for b with a slightly different notation):

$$\phi(\mathbf{r}) = \phi_{i0} + \mathbf{C}_i \cdot (\mathbf{r} - \mathbf{r}_{i0}), \quad (40)$$

where \mathbf{C}_i is the unknown coefficient vector (three components), and the reference position \mathbf{r}_{i0} and the field value ϕ_{i0} can be calculated directly as the weighted average of cell-centre positions and cell-centre field values:

$$\mathbf{r}_{i0} = \frac{\sum_{j=1}^n w_{ij} \mathbf{r}_{ij}}{\sum_{j=1}^n w_{ij}}, \quad \phi_{i0} = \frac{\sum_{j=1}^n w_{ij} \phi_{ij}}{\sum_{j=1}^n w_{ij}}, \quad (41)$$

where \mathbf{r}_{ij} is the centre of cell j in the interpolation stencil of the vertex i , ϕ_{ij} is the field value in the centre of cell j in the interpolation stencil of the vertex i , and w_{ij} is the weighting factor calculated as the inverse square distance between the centre of cell j and the position of vertex i . The unknown coefficient vector is determined using the weighted least squares method by inverting the matrix of normal equations:

$$\mathbf{C}_i = [(\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}] \cdot \Phi_i, \quad (42)$$

where Φ_i is the vector consisting of quantity $(\phi_{ij} - \phi_{i0})$ for all cells in the interpolation stencil of the vertex i , \mathbf{W} is the diagonal matrix consisting of weighting factors for all cells in the interpolation stencil of the vertex i , and \mathbf{X} is the $n \times 3$ matrix whose row j is the position vector $(\mathbf{r}_{ij} - \mathbf{r}_{i0})$ of the cell j in the interpolation stencil of the vertex i .

In the case of parallel calculations, special attention should be devoted to the vertices at the processor boundary. For example, the interpolation procedure for the vertex i at the processor boundary (see Fig. 1) is performed independently on two processors, but the field value obtained in such a way should be exactly the same. To accomplish this, all cell-centre field values in the interpolation stencil must be present on both processors. A parallel data exchange in OpenFOAM is performed across inter-processor boundary faces allowing access to the neighbour processor cell-centre field values only for the cells sharing the inter-processor boundary faces with the local processor. Therefore, when the field value is calculated for the vertex i on the processor PROC-1, the cell-centre field values for the cells $i1$ and $i4$ from the neighbour processor PROC-0 can be accessed, but the cells $i2$ and $i3$ cannot because the corresponding cells do not share a face with the inter-processor boundary (see Fig. 1). For these cells, an additional inter-processor data exchange has to be introduced in order to ensure the same vertex reconstructed values on both sides of the inter-processor boundary. A similar problem exists in the case of the boundary vertex b at the inter-processor boundary because the boundary field values from the neighbour processor required for performing the vertex field reconstruction are not available.

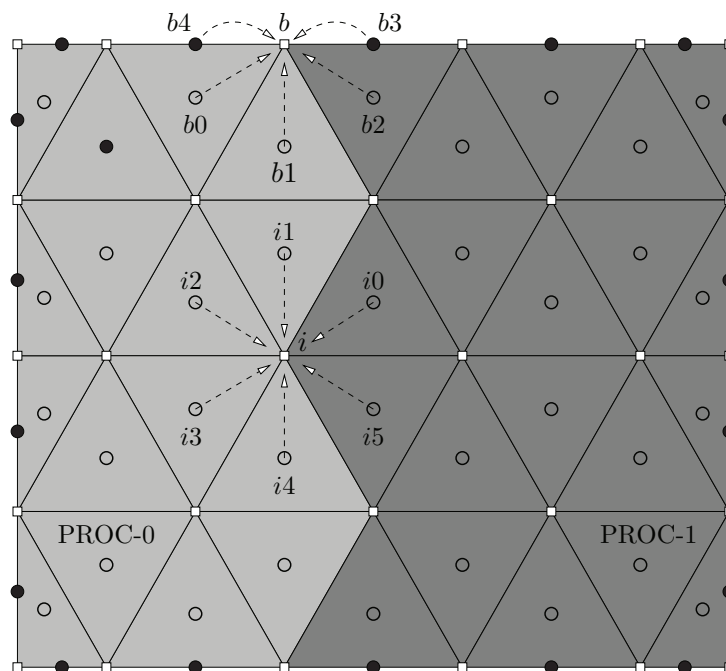


Fig. 1 Two-dimensional FV mesh with an interpolation stencil for the vertex field value reconstruction. The interpolation stencil is given for the internal vertex i and the boundary vertex b , which are also positioned at the inter-processor boundary.

REFERENCES

- [1] Karac, A., Ivankovic, A., Drop impact of fluid-filled plastic containers: Finite volume method for coupled fluid-structure-fracture problems, WCCM V, Fifth World Congress on Computational Mechanics, Vienna, Austria, July 2002.

- [2] Hou, G., Wang, J., Layton, A., Numerical methods for fluid-structure interaction – A review, *Communications in Computational Physics* 12 (2) (2012), pp. 337–377, <https://doi.org/10.4208/cicp.291210.290411s>.
- [3] Slone, A. K., Pericleous, K., Bailey, C., Cross, M., Bennett, C., A finite volume unstructured mesh approach to dynamic fluid-structure interaction: an assessment of the challenge of predicting the onset of flutter, *Applied mathematical modelling* 28 (2004), pp. 211–239, [https://doi.org/10.1016/S0307-904X\(03\)00142-2](https://doi.org/10.1016/S0307-904X(03)00142-2).
- [4] Degroote, J., Vierendeels, J., Multi-solver algorithms for the partitioned simulation of fluid-structure interaction, *Computational Methods in Applied Mechanical Engineering* 200 (2528) (2011), pp. 2195–2210, <https://doi.org/10.1016/j.cma.2011.03.015>.
- [5] Degroote, J., Bathe, K.-J., Vierendeels, J., Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction, *Computers and structures* 87 (2009), pp. 793–801, <https://doi.org/10.1016/j.compstruc.2008.11.013>.
- [6] Richter, T., A monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation, *International journal for numerical methods in engineering* 104 (2015), pp. 372–390, <https://doi.org/10.1002/nme.4943>.
- [7] Gillebaart, T., Blom, D. S., van Zuijlen, A. H., Bijl, H., Time consistent fluid structure interaction on collocated grids for incompressible flow, *Computer Methods in Applied Mechanics and Engineering* 298 (0) (2016), pp. 159–182, <https://doi.org/10.1016/j.cma.2015.09.025>.
- [8] Xia, G., Lin, C.-L., An unstructured finite volume approach for structural dynamics in response to fluid motions, *Computers and Structures* 86 (2008), pp. 684–701, <https://doi.org/10.1016/j.compstruc.2007.07.008>.
- [9] Malan, A., Oxtoby, O., An accelerated, fully-coupled, parallel 3D hybrid finite-volume fluid-structure interaction scheme, *Computer methods in applied mechanics and engineering* 253 (2013), pp. 426–438, <https://doi.org/10.1016/j.cma.2012.09.004>.
- [10] Lv, X., Zhao, Y., Huang, X. Y., Xia, G. H., Wang, Z. J., An efficient parallel/unstructured-multigrid preconditioned implicit method for simulating 3D unsteady compressible flows with moving objects, *Journal of computational physics* 215 (2) (2006), pp. 661–690, <https://doi.org/10.1016/j.jcp.2005.11.012>.
- [11] Lv, X., Zhao, Y., Huang, X. Y., Xia, G. H., Su, X. H., A matrix-free implicit unstructured multigrid finite volume method for simulating structural dynamics and fluid-structure interaction, *Journal of computational physics* 225 (1) (2007), pp. 120–144, <https://doi.org/10.1016/j.jcp.2006.11.023>.
- [12] Moosavi, M., Khelil, A., Nonlinear internal fluid structure interaction by finite volume methods in both domains, *International Journal for Computational Methods in Engineering Science and Mechanics* 16 (2015), pp. 86–97, <https://doi.org/10.1080/15502287.2015.1009578>.
- [13] Greenshields, C., Weller, H., Ivankovic, A., The finite volume method for coupled fluid flow and stress analysis, *Computer Modeling and Simulation in Engineering* 4 (3) (1999), pp. 213–218.
- [14] Karač, A., Ivankovic, A., Investigating the behaviour of fluid-filled polyethylene containers under base drop impact: A combined experimental/numerical approach, *International Journal of Impact Engineering* 36 (4) (2009), pp. 621–631, <https://doi.org/10.1016/j.ijimpeng.2008.08.007>.
- [15] Karač, A., Ivankovic, A., Modelling the drop impact behaviour of fluid-filled polyethylene containers, *ESIS Publication, Fracture of Polymers, Composites and Adhesives II* 32 (2003), pp. 253–264.
- [16] Ivankovic, A., Karač, A., Dendrinos, E., K, P., Towards early diagnosis of atherosclerosis: the finite volume method for fluid-structure interaction, *Biorheology* 39 (2002), pp. 401–407.
- [17] Kanyanta, V., Ivankovic, A., Karač, A., Validation of a fluid-structure interaction numerical model for predicting flow transients in arteries, *Journal of Biomechanics* 42 (11) (2009), pp. 1705–1712, <https://doi.org/10.1016/j.jbiomech.2009.04.023>.
- [18] Issa, R. I., Solution of the implicitly discretised fluid flow equations by operator-splitting, *Journal of computational physics* 62 (1) (1986), pp. 40–65, [https://doi.org/10.1016/0021-9991\(86\)90099-9](https://doi.org/10.1016/0021-9991(86)90099-9).
- [19] Rhie, C. M., Chow, W. L., A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, *AIAA Journal* 21 (1983), pp. 1525–1532, <https://doi.org/10.2514/3.8284>.
- [20] Tuković, Ž., Jasak, H., A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow, *Computers and fluids* 55 (2012), pp. 70–84, <https://doi.org/10.1016/j.compfluid.2011.11.003>.
- [21] Tuković, Ž., Perić, M., Jasak, H., Consistent second-order time-accurate non-iterative PISO-algorithm, *Computers and Fluids* 166 (2018), pp. 78–85, <https://doi.org/10.1016/j.compfluid.2018.01.041>.

- [22] Demirdžić, I., Muzaferija, S., Finite volume method for stress analysis in complex domains, *International journal for numerical methods in engineering* 37 (21) (1994), pp. 3751–3766, <https://doi.org/10.1002/nme.1620372110>.
- [23] Demirdžić, I., Muzaferija, S., Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology, *Computer methods in applied mechanics and engineering* 125 (1-4) (1995), pp. 235–255, [https://doi.org/10.1016/0045-7825\(95\)00800-G](https://doi.org/10.1016/0045-7825(95)00800-G).
- [24] Jasak, H., Weller, H. G., Application of the finite volume method and unstructured meshes to linear elasticity, *International journal for numerical methods in engineering* 48 (2) (2000), pp. 267–287, [https://doi.org/10.1002/\(SICI\)1097-0207\(20000520\)48:2;267::AID-NME884;3.0.CO;2-Q](https://doi.org/10.1002/(SICI)1097-0207(20000520)48:2;267::AID-NME884;3.0.CO;2-Q).
- [25] Tuković, Ž., Jasak, H., Updated lagrangian finite volume solver for large deformation dynamic response of elastic body, *Transactions of FAMENA* 31 (1) (2007), pp. 55–70.
- [26] Tuković, Ž., Ivanković, A., Karač, A., Finite-volume stress analysis in multi-material linear elastic body, *International journal for numerical methods in engineering* 93 (2013), pp. 400–419, <https://doi.org/10.1002/nme.4390>.
- [27] Carolan, D., Tuković, Ž., Murphy, N., A. Ivanković, A., Arbitrary crack propagation in multi-phase materials using the finite volume method, *Computational Materials Science* 69 (2013), pp. 153–159, <https://doi.org/10.1016/j.commatsci.2012.11.049>.
- [28] Cardiff, P., Karač, A., Ivanković, A., A large strain finite volume method for orthotropic bodies with general material orientations, *Computer Methods in Applied Mechanics and Engineering* 268 (1) (2014), pp. 318–335, <https://doi.org/10.1016/j.cma.2013.09.008>.
- [29] Jasak, H., Weller, H., Finite volume methodology for contact problems of linear elastic solids, *Third International Conference of Croatian Society Mechanics, Cavtat/Dubrovnik, Croatia*, pp. 253–260, 2000.
- [30] Cardiff, P., Karač, A., Ivanković, A., Development of a finite volume contact solver based on the penalty method, *Computational Materials Science* 64 (2012), pp. 283–284, <https://doi.org/10.1016/j.commatsci.2012.03.011>.
- [31] Habchi, C., Russeil, S., Bougeard, D., Harion, J. L., Lemenand, T., A., G., Valle, D. D., Peerhossaini, H., Partitioned solver for strongly coupled fluid-structure interaction, *Computers and fluids* 71 (2013), pp. 306–319, <https://doi.org/10.1016/j.compfluid.2012.11.004>.
- [32] Wiedemair, W., Tuković, Ž., Jasak, H., Poulikakos, D., Kurtcuoglu, V., On ultrasound-induced microbubble oscillation in a capillary blood vessel and its implications for the bloodbrain barrier, *Physics in Medicine and Biology* 57 (4) (2012), pp. 1019–1045, <https://doi.org/10.1088/0031-9155/57/4/1019>.
- [33] Ali, S., Habchi, C., Menanteau, S., Lemenand, T., Harion, J.-L., Heat transfer and mixing enhancement by free elastic flaps oscillation, *International Journal of Heat and Mass Transfer* 85 (2015), pp. 250–264, <https://doi.org/10.1016/j.ijheatmasstransfer.2015.01.122>.
- [34] Šekutkovski, B., Kostić, I., Simonović, A., Cardiff, P., Jazarević, V., Three-dimensional fluid-structure interaction simulation with a hybrid RANS-LES turbulence model for applications in transonic flow domain, *Aerospace Science and Technology* 49 (2016), pp. 1–16, <https://doi.org/10.1016/j.ast.2015.11.028>.
- [35] Thomas, P. D., Lombard, C. K., Geometric conservation law and its application to flow computations on moving grids, *AIAA Journal* 17 (1979), pp. 1030–1037, <https://doi.org/10.2514/3.61273>.
- [36] Demirdžić, I., Perić, M., Space conservation law in finite volume calculations of fluid flow, *International journal for numerical methods in fluids* 8 (9) (1988), pp. 1037–1050, <https://doi.org/10.1002/flid.1650080906>.
- [37] Jasak, H., Tuković, Ž., Automatic mesh motion for the unstructured finite volume method, *Transactions of FAMENA* 30 (2) (2006), pp. 1–20.
- [38] Ferziger, J. H., Perić, M., *Computational methods for fluid dynamics*, Springer Verlag, Berlin-New York, 1995.
- [39] Kershaw, D., The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations, *Journal of Computational Physics* 26 (1) (1978), pp. 43–65, [https://doi.org/10.1016/0021-9991\(78\)90098-0](https://doi.org/10.1016/0021-9991(78)90098-0).
- [40] Jacobs, D. A. H., *Preconditioned Conjugate Gradient methods for solving systems of algebraic equations*, Technical Report RD/L/N193/80, Central Electricity Research Laboratories, 1980.
- [41] Irons, B., Tuck, R. C., A version of the Aitken accelerator for computer implementation, *International journal for numerical methods in engineering* 1 (1969), pp. 275–277, <https://doi.org/10.1002/nme.1620010306>.

- [42] Küttler, U., Wall, W. A., Fixed-point fluid-structure interaction solvers with dynamic relaxation, *Computational mechanics* 43 (1) (2008), pp. 61–72, <https://doi.org/10.1007/s00466-008-0255-5>.
- [43] Degroote, J., Development of Algorithms for the Partitioned Simulation of Strongly Coupled Fluid-Structure Interaction Problems, Ph.D. thesis, University of Ghent, 2010.
- [44] Beaudoin, M., Jasak, H., Development of a generalized grid interface for turbomachinery simulations with OpenFOAM, Open Source CFD International Conference, Berlin, Germany, pp. 1–11, December 2008.
- [45] Sutherland, I. E., Hodgman, G. W., Reentrant polygon clipping, *Commun. ACM* 17 (1) (1974), pp. 32–42, <https://doi.org/10.1145/360767.360802>.
- [46] Verth, J., Bishop, L., *Essential Mathematics for Games and Interactive Applications: Programmer's guide*, Morgan Kaufmann Publishers Inc., San Francisco, 2004.
- [47] Gottschalk, S., Separating axis theorem, Technical Report TR96-024, Department of Computer Science, UNC Chapel Hill, 1996.
- [48] Gropp, W., Lusk, E., Doss, N., Skjellum, A., A high performance, portable implementation of the MPI message passing interface standard, *Parallel computing* 22 (1996), pp. 789–828, [https://doi.org/10.1016/0167-8191\(96\)00024-5](https://doi.org/10.1016/0167-8191(96)00024-5).
- [49] Timoshenko, S. P., J. N. Goodier, J. N., *Theory of Elasticity*, McGraw-Hill, London, 1970.
- [50] Demirdžić, I., Muzaferija, S., Perić, M., Benchmark solutions of some structural analysis problems using finite volume method and multigrid acceleration, *International journal for numerical methods in engineering* 40 (10) (1997), pp. 1893–1908, [https://doi.org/10.1002/\(SICI\)1097-0207\(19970530\)40:10<1893::AID-NME146>3.0.CO;2-L](https://doi.org/10.1002/(SICI)1097-0207(19970530)40:10<1893::AID-NME146>3.0.CO;2-L).
- [51] Juretić, H., Gosman, A. D., Error analysis of the finite-volume method with respect to mesh type, *Numerical heat transfer, Part B* 57 (6) (2010), pp. 414–439, <https://doi.org/10.1080/10407791003685155>.
- [52] Roache, P. J., Quantification of uncertainty in computational fluid dynamics, *Annual Review of Fluid Mechanics* 29 (1997), pp. 123–160, <https://doi.org/10.1146/annurev.fluid.29.1.123>.
- [53] Turek, S., Hron, J., Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, H.-J. Bungartz, M. Schäfer (Editors) *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, Springer Berlin Heidelberg, ISBN 978-3-540-34595-4, pp. 371–385, 2006.
- [54] Wylie, E. B., Streeter, V. L., *Fluid Transients in Systems*, Englewood Cliffs, New York, 1993.
- [55] Richter, T., Goal-oriented error estimation for fluid-structure interaction problems, *Computer Methods in Applied Mechanics and Engineering* 223-224 (2012), pp. 28–42, <https://doi.org/10.1016/j.cma.2012.02.014>.

Submitted: 19.6.2017

Accepted: 08.5.2018

Željko Tuković
Zeljko.Tukovic@fsb.hr
Hrvoje Jasak
University of Zagreb,
Faculty of Mechanical Engineering
and Naval Architecture,
Ivana Lučića 5,
Zagreb, Croatia

Aleksandar Karač
University of Zenica,
Polytechnic Faculty,
Fakultetska 1, Zenica,
Bosnia and Herzegovina

Philip Cardiff
Alojz Ivanković
University College Dublin,
Bekaert University Technology Centre,
School of Mechanical and
Materials Engineering,
Belfield, Dublin, Ireland