A matrix-free, implicit, incompressible fractional-step algorithm for fluid–structure interaction applications

O. F. Oxtoby*, A. G. Malan[1]

Aeronautic Systems, Council for Scientific and Industrial Research, Building 12, Box 395, Pretoria 0001, South Africa

Abstract

In this paper we detail a fast, fully-coupled, partitioned fluid–structure interaction (FSI) scheme. For the incompressible fluid, new fractional-step algorithms are proposed which make possible the fully implicit, but matrix-free, parallel solution of the entire coupled fluid–solid system. These algorithms include artificial compressibility pressure-poisson solution in conjunction with upwind velocity stabilisation, as well as simplified pressure stabilisation for improved computational efficiency. A dual-timestepping approach is proposed where a Jacobi method is employed for the momentum equations while the pressures are concurrently solved via a matrix-free preconditioned GMRES methodology. This enables efficient sub-iteration level coupling between the fluid and solid domains. Parallelisation is effected for distributed-memory systems. The accuracy and efficiency of the developed technology is evaluated by application to benchmark problems from the literature. The new schemes are shown to be efficient and robust, with the developed preconditioned GMRES solver furnishing speed-ups ranging between 50 and 80.

Keywords:
fluid–structure interaction, finite volume method, incompressible split, matrix free, parallelization, preconditioned GMRES

*Corresponding author

Email addresses: `ooxtoby@csir.co.za` (O. F. Oxtoby), `Arnaud.Malan@uct.ac.za` (A. G. Malan)

[1]Present address: Department of Mechanical Engineering, University of Cape Town, Private Bag X3, Rondebosch 7701, South Africa

1. Introduction

Fluid–Structure Interaction (FSI) is a growing field within computational mechanics with important industrial applications. For example, in aeroelastic systems, the drive to model non-linear flutter response has spawned the field of Computational Aeroelastics [1], while in Computational Biomechanics the structural response of cardiac, arterial and respiratory systems plays a cental role in the flow characteristics therein [2, 3]. In real-world systems such as these, geometries are complex and require considerable computing power to accurately resolve. In addition, FSI problems typically call for unsteady analyses where one seeks to identify transient responses or limit-cycle states rather than steady-state behaviour. All of this means that FSI calculations are computationally onerous, and while recent years have seen much progress in FSI modelling technology [4, 5, 6, 7, 8], its biggest hurdle to becoming a realistically viable tool in industry is still its high computational cost. In this paper we develop an implicit, matrix-free solver within the *Elemental* multiphysics code [9, 10, 11, 12, 13, 14] to solve FSI problems in a computationally efficient manner.

In order to robustly solve fluid-solid systems which are physically strongly coupled, it is of value to fully converge the solution at each timestep, so that both dynamic and kinematic continuity – i.e. continuity of forces and velocities – are satisfied at the fluid/solid interface. So-called monolithic methods ensure this by solving the entire coupled system implicitly [15, 16, 17]. Partitioned solvers, on the other hand, are flexible in allowing independent treatment of fluid and solid, but require that care be taken to ensure the stability of the coupling process [18, 19, 20, 21, 22, 23]. In this paper, we propose a system where dual-timestepping [24, 25] is employed for the solid and the fluid momentum equations to effect the fully-coupled, implicit solution of the entire FSI system in a partitioned manner. This allows for the realisation of optimum parallel efficiency as fluid and solid domains are computed simultaneously.

For the fluid, incompressibility has traditionally been dealt with in one of two ways: Either using the pressure projection (PP) methods which originated with Patankar [26], or Chorin's artificial compressibility (AC) [27]. Recently, these two historically opposing methodologies have been combined into one algorithm in the form of the Artificial Compressibility Characteristic-Based Split (CBS-AC) scheme [14, 28, 29], which exhibits the desirable characteristics of both. Building on this, we introduce in this paper an upwind stabilised pressure-projection artificial compressibility (UP-AC) method. In this scheme the characteristic-based term used to stabilise velocity in the CBS-AC method is replaced with higher-order upwinding of the convective velocities. This somewhat simplifies the implementation but more importantly is better suited to stabilising flows with discontinuities in velocity since limiters can be applied. Furthermore, since the time-step size dependence of the characteristic-based term is eliminated, we show that local timestepping can be used to accelerate convergence without impacting accuracy. As a further simplification to the method, we consider a stabilised direct discretisation of the continuity equation which employs the Consistent Numerical Fluxes of Löhner *et al.* [30, 31] (CNF-AC). As this is no longer a pressure-projection method, it consists of two rather than three steps and a consequent reduction in computational complexity, as well as simplified implementation.

The use of artificial compressibility naturally allows for matrix-free solution and efficient parallelisation. However, convergence can be slow if using simple Jacobi iteration due to the fact that pressure waves have to propagate and equalise across the entire fluid domain. Therefore an implicit method of solving the pressure equation, without sacrificing its matrix-free parallel nature, would produce significant savings in computation time. For the treatment of the advective terms, on the other hand, explicit timestepping has been shown to be as fast or faster than implicit matrix-free methods in the case of transient problems [31]. The CBS-AC method has been used with explicit timestepping [28] and dual-timestepping [32, 33] to increase the allowable time-step size, while still treating the pressure equation in an explicit manner. In this paper we propose a new approach in which the momentum equations are solved using Jacobi dual-timestepping while pressures are implicitly integrated using a preconditioned GMRES solver. This recipe combines the computational expediency of explicit momentum advection with the fully matrix-free and implicit treatment of pressures. A novel feature of our method is that the pressure equation is not solved to convergence at each iteration, but converged simultaneously with the momentum equations. Therefore, a computationally inexpensive, iterative advanced solver is essential for the pressure equation.

In this work, both fluid and solid governing equations are spatially discretised via an edge-based finite volume method [11, 34] whose accuracy is notionally of second order. Edge-based methods hold the advantages of generic applicability to hybrid-unstructured meshes, ideal parallelisation properties and improved computational efficiency compared to element-based approaches [35]. In the case of the solid domain, we use a hybrid between the traditional node-based finite volume method (which suffers from locking of high aspect-ratio elements) and the element-based strain method [36] (which suffers from odd-even decoupling), to circumvent both of these problems [37].

The outline of this paper is as follows. In Section 2 we present the governing equations for fluid and solid domains, then describe the spatial discretisation and mesh movement algorithm in Section 3. Section 4 details the numerical solvers and coupling algorithm, and in Section 5 parallelisation of the code is discussed. We present numerical applications in Section 6 before concluding in Section 7.

2. Governing Equations

The physical domain to be modelled consists of a viscous incompressible fluid and homogeneous isotropic elastic solid region. The mechanics of each is governed by the appropriate governing equation set. In this work the fluid mesh boundary is fitted to the deforming solid and the internal nodes moved using the mesh movement algorithm described in Section 3.3.

2.1. Fluid equations

We consider an incompressible viscous fluid undergoing laminar 2D isothermal flow. We write the governing equations for an arbitrary Lagrangian Eulerian (ALE) coordinate frame in weak form as

$$\frac{\partial}{\partial t}\int_{\mathcal{V}(t)} \mathbf{W} d\mathcal{V} + \int_{\mathcal{S}(t)} \left(\mathbf{F}^j + \mathbf{H}^j - \mathbf{G}^j\right) n_j d\mathcal{S} = \int_{\mathcal{V}(t)} \mathbf{Q} d\mathcal{V}, \tag{1}$$

where $\mathcal{V}(t)$ denotes an arbitrary volume translating at mesh velocity $\mathbf{u}^*$ and

$$\mathbf{W} = \begin{pmatrix} W_0 \\ W_1 \\ W_2 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \end{pmatrix}, \quad \mathbf{H}^j = \begin{pmatrix} 0 \\ p\delta_{1j} \\ p\delta_{2j} \end{pmatrix}, \quad \mathbf{G}^j = \begin{pmatrix} 0 \\ \sigma_{1j} \\ \sigma_{2j} \end{pmatrix}, \tag{2}$$

$$\mathbf{F}^j = \mathbf{W}(u_j - u_j^*). \tag{3}$$

In the above, $\mathcal{S}(t)$ denotes the surface of the volume $\mathcal{V}(t)$ with $\mathbf{n}$ being the outward pointing unit normal vector; $\mathbf{Q}$ is a vector of source terms (e.g. body forces), $\mathbf{u}$ denotes velocity, p is the pressure, ρ is density, $\boldsymbol{\sigma}$ stress, and δ_{ij} is the Kronecker delta.

The governing equations are closed via the relationship between stress and rate of strain:

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{4}$$

where μ is the dynamic viscosity and x_i are the fixed (Eulerian) co-ordinates.

2.2. Solid Equations

In this work we model isotropic elastic solids using a finite volume formulation (as has been done, for instance, in [36, 38]). To account for large deformations without accumulating strain errors due to repeated oscillations, the solid equations are cast in a total Lagrangian formulation, i.e. in the undeformed reference frame. The resulting momentum equations are written in weak form as

$$\frac{\partial}{\partial t} \int_{\mathcal{V}_0} \rho_0 v_i d\mathcal{V} = \int_{\mathcal{S}_0} P_{ij} n_j d\mathcal{S} + \int_{\mathcal{V}_0} Q_i d\mathcal{V}, \tag{5}$$

where $\mathbf{v}$ is the solid velocity, $\mathbf{P}$ is the first Piola-Kirchoff stress tensor, $\mathcal{V}_0$ denotes a volume in the material coordinate system and $\mathcal{S}_0$ its surface, with $\mathbf{n}$ being its outward pointing unit normal vector. Further, ρ_0 is the solid density in the undeformed state and $\mathbf{Q}$, again, a vector of source terms.

To account for finite strains, we employ the St. Venant–Kirchoff model with the Green-Lagrange strain tensor

$$E_{ij} = \frac{1}{2} \left(\frac{\partial w_i}{\partial X_j} + \frac{\partial w_j}{\partial X_i} + \frac{\partial w_k}{\partial X_i} \frac{\partial w_k}{\partial X_j} \right) \tag{6}$$

where $\mathbf{w}$ is the total displacement of the solid from equilibrium and X_i are the material coordinates of the solid, so that $\mathbf{x} = \mathbf{X} + \mathbf{w}$.

Under the plane-strain assumption, the second Piola-Kirchoff stress tensor $\mathbf{S}$ is related to the strain tensor by

$$S_{ij} = 2G(E_{ij} - \tfrac{1}{3} E_{kk} \delta_{ij}) + K E_{kk} \delta_{ij} \tag{7}$$

where G is the shear modulus and K the bulk modulus (summation over k implied).

Finally, we convert from the second to the first Piola-Kirchoff stress tensor, $\mathbf{P}$, with

$$P_{ij} = F_{ik} S_{kj}, \text{ where } F_{ik} = \delta_{ik} + \frac{\partial w_i}{\partial X_k}. \tag{8}$$

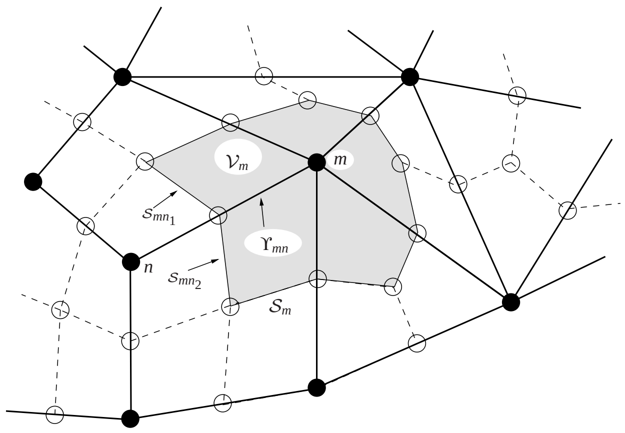

Figure 1: Schematic diagram of the construction of the median dual-mesh on hybrid grids. Here, Υ_{mn} depicts the edge connecting nodes m and n.

To close the system of equations, the solid velocity $\mathbf{v}$ and displacement $\mathbf{w}$ are related in the obvious way:

$$\frac{Dw_j}{Dt} = v_j, \tag{9}$$

where D/Dt denotes the derivative with respect to a fixed point in the reference (initial) configuration.

3. Spatial Discretisation

3.1. Hybrid-Unstructured Discretisation

We use a vertex-centred, edge-based finite volume algorithm for the purpose of spatial discretization, where a compact stencil method is employed for second-derivative terms in the interests of both stability and accuracy [11, 34]. The method allows natural generic mesh applicability, second-order accuracy without odd-even decoupling, and computational efficiency which is factors greater than element-based approaches [35]. The edge-based scheme is also particularly well suited to computation on parallel hardware architectures due to the constant computational cost per edge.

Considering the fluid ALE governing equations (1), all surface integrals are calculated in an edge-wise manner. For this purpose, bounding surface information is similarly stored per edge and termed *edge-coefficients*. The latter, for a given internal edge Υ_{mn} connecting nodes m and n, is defined as a function of time as

$$\mathbf{C}_{mn}(t) = \mathbf{n}^{mn_1}\mathcal{S}_{mn_1}(t) + \mathbf{n}^{mn_2}\mathcal{S}_{mn_2}(t) \tag{10}$$

where $\mathcal{S}_{mn_1}$ is a bounding surface-segment intersecting the edge (Fig. 1) and the normal unit vectors are similarly a function of time. The discrete form of the surface integral in Eq. (1), computed for the volume surrounding the node m, now follows as

$$\int_{\mathcal{S}_m(t)} \left\{\mathbf{F}^j + \mathbf{H}^j - \mathbf{G}^j\right\} n_j d\mathcal{S} \approx \sum_{\Upsilon_{mn}\cap\mathcal{V}_m(t)} \left\{\overline{\mathbf{F}^j}_{mn} + \overline{\mathbf{H}^j}_{mn} - \overline{\mathbf{G}^j}_{mn}\right\} C^j_{mn} \tag{11}$$

where all $\overline{\bullet}_{mn}$ quantities denote edge-averaged values which are calculated such that second-order accuracy of the overall scheme is ensured [11]. In the case of the fluid, $\overline{\mathbf{G}^j}_{mn} = \left[\overline{\mathbf{G}^j}_{mn}\Big|_{tang} + \overline{\mathbf{G}^j}_{mn}\Big|_{norm}\right]$, where $\mathbf{G}^j\Big|_{tang}$ is calculated by employing directional derivatives and $\mathbf{G}^j\Big|_{norm}$ is approximated by employing the standard finite volume first derivative terms.

When considering the solid governing equations (5), the above method of discretising the stress term would enforce continuous gradients at nodes and element boundaries and therefore suffers from stiffness problems on stretched elements. To remedy this we use a hybrid nodal/elemental strain technique [37] in which strains E_{11} and E_{22} are evaluated at nodes and averaged to obtain face values (as per the above method), whereas E_{12} is evaluated at the element centre and averaged with the neighbouring element's value to obtain the value at the shared face.

3.2. Geometric Conservation

In order for the fluid solution to be as transparent as possible to the movement of the mesh, an identity known as the Geometric Conservation Law (GCL) should be obeyed [39, 40, 41]. It asserts that the momentum flux into a cell due to the motion of the faces should be consistent with the change in momentum of the cell due to its changing volume. That is, the discretised version of

$$\frac{\partial}{\partial t}\int_{\mathcal{V}(t)} d\mathcal{V} = \int_{\mathcal{S}(t)} u^*_j n_j d\mathcal{S} \tag{12}$$

should hold exactly, which implies that constant spatial fields will be unaffected by arbitrary mesh deformations. The GCL can therefore be said to impose a specific relationship between mesh deformation and the mesh-velocity field. To enforce the GCL we could discretise the equation above at node m as

$$\frac{V_m^{t+\Delta t} - V_m^t}{\Delta t} = \sum_{\Upsilon_{mn} \cap \mathcal{V}_m(t)} \frac{\delta V_{mn}^t}{\Delta t}, \tag{13}$$

where δV_{mn}^t is the volume swept out by the face lying between nodes m and n between time-steps t and $t + \Delta t$, and V_m is the volume of the dual-cell containing node m. Alternatively, to second order accuracy

$$\frac{3V_m^{t+\Delta t} - 4V_m^t + V_m^{t-\Delta t}}{2\Delta t} = \sum_{\Upsilon_{mn} \cap \mathcal{V}_m(t)} \frac{3\delta V_{mn}^t - \delta V_{mn}^{t-\Delta t}}{2\Delta t}. \tag{14}$$

So, in order for our discretisation to be consistent with the GCL, the mesh velocity flux $\overline{u_j^* C_{mn}^j}$ in the discretisation of (1) is set equal to $(3\delta V_{mn}^t - \delta V_{mn}^{t-\Delta t})/2\Delta t$, similar to the expression used in [41].

3.3. Dynamic Mesh Movement

For the purposes of mesh movement, we employ an interpolation procedure which, while offering no guarantees about element quality, has no significant computational cost and is well suited to parallel computing. This approach entails redistributing internal fluid nodes according to the distance from two fixed boundary regions, for instance internal and external boundaries. The following interpolation function is used:

$$\Delta \mathbf{x} = r\Delta \mathbf{x}_1 + (1 - r)\Delta \mathbf{x}_2,$$

where $\Delta \mathbf{x}_1$ and $\Delta \mathbf{x}_2$ respectively denote the displacement of the closest internal and external boundary nodes from their initial locations, and r, which varies between zero and one, is computed as

$$r = \frac{D_2^p}{D_1^p + D_2^p} \quad \text{with } p = 3/2.$$

Here, D_1 and D_2 are the distances to the identified boundary points in the undeformed configuration. The value $p = 3/2$ was found to produce the best results for the largest displacements of the block with flexible tail problem considered below, and may not be optimal for different geometries. Problems involving smaller displacements will be less sensitive to the value chosen, however. The closest point and the values of r are only calculated at the start of the analysis. Thus, they are based only on the initial configuration in order to ensure that the the mesh does not deteriorate due to repeated oscillations. As a result, the application of the mesh movement function is essentially instantaneous.

4. Temporal Discretisation and Solution Procedure

The solution procedure must allow for fully coupled implicit solution of all discritised equations while allowing independence in terms of discretisation and solution strategy employed for the fluid and solid domains. We therefore advocate a partitioned, matrix-free iterative solution process where fluid-solid interface nodes communicate velocities and tractions at each iteration. The resulting proposed solution procedure is detailed below.

4.1. Temporal Discretisation

For the purpose of transient calculations, a dual-timestepping [24, 25] temporal discretisation [9] is employed such that second-order temporal accuracy is achieved while ensuring that all equations are iteratively solved simultaneously to obtain the implicit solution. The real-time temporal term is accordingly discretised and added as a source term to the right-hand-side of the discretised fluid equation as

$$Q_i V\Big|^{\tau} = -\frac{3W_i^{\tau}V^{\tau} - 4W_i^{t}V^{t} + W_i^{t-\Delta t}V^{t-\Delta t}}{2\Delta t} \quad \text{for } i = 1, 2 \tag{15}$$

where Δt denotes the real-time-step size, the t superscript is the previous (existing) real time-step and τ denotes the latest known solution to the time-step being solved for viz. $t + \Delta t$.

For the solid equation, a similar source term is added to the right-hand side of the discretised version of Eq. (5), namely

$$Q_i^{\tau} V_0 = -\rho_0 \frac{3v_i^{\tau} - 4v_i^{t} + v_i^{t-\Delta t}}{2\Delta t} V_0 \tag{16}$$

and also to the discrete form of Eq. (9), to give

$$\frac{Dw_i}{Dt} = v_i - \frac{3w_i^{\tau} - 4w_i^{t} + w_i^{t-\Delta t}}{2\Delta t} \tag{17}$$

where the nomenclature is as previously defined.

4.2. Solution Procedure: Fluid

The solution of the incompressible fluid equations (2) presents two numerical difficulties. Firstly, the spatial discretisation of the convective terms via linear interpolation results in destabilising odd-even decoupling, and secondly, the incompressibility of the fluid demands that the pressure field evolve such that the continuity equation $\nabla \cdot \mathbf{u} = 0$ is satisfied. Since this equation does not involve pressure, solving for it in a matrix-free manner is not straightforward. We investigate two algorithms to overcome these difficulties. The first is an Upwind Pressure-Projection Artificial Compressibility (UP-AC) algorithm. This method builds on the Artificial Compressibility Characteristic Based Split (CBS-AC) algorithm of Nithiarasu [14, 28, 29], but stabilises convective velocities using upwinding rather than the characteristic-based approach. Secondly, we introduce a more economical approach to pressure-stabilisation, in which artificial compressibility is added to the divergence-free constraint to solve for pressure, along with forth-order stabilisation terms similar to those employed by Löhner *et al.* [30, 31] to prevent its consequent chequerboarding. These two techniques are described in detail below.

For incompressible flow it is usually advocated that the pressures are solved implicitly [31] while momentum advection terms are often explicitly integrated [28, 31] since the advective timescales are those of interest, whereas pressure waves propagate instantaneously. Löhner *et al.* [31] have undertaken a thorough investigation of different strategies of explicit and implicit integration of pressure and momentum equations and the system as a whole, with the fastest solution times for transient problems achieved using explicit integration of the advective terms. However, for fluid–structure interaction problems, explicit timestepping is not ideal as it can lead to instabilities if used in a staggered (weakly-coupled) mode [42]. Here we propose an approach that effects strong coupling while retaining the efficiency of Jacobi solution of the momenum equations.

Dual-timesteping has long been used to increase the allowable real-timestep size without introducing the extra computational cost of an implicit solver [9, 11, 24, 25, 32]. In the context of FSI, it confers the additional advantage of allowing the coupled system to be solved simultaneously but with the iterative solution process serving the extra purpose of effecting coupling between fluid and solid. In other words, it allows for coupling information to be exchanged on a fine-grained level while both fluid and solid are concurrently converged, nonetheless keeping the discretisation and solver methodologies of the two entirely partitioned. We therefore propose the use of dual-timestepping for the momentum equations combined with implicit integration of pressures. We now describe in detail the two split-step incompressible flow algorithms.

4.2.1. UP-AC algorithm

To account for moving meshes we consider Eq. (1) with a time-dependent dual-cell $\mathcal{V}(t)$ which is moving at the mesh velocity $\mathbf{u}^*$, from which the first incremental solution step written in semi-discrete form follows as:

$$\frac{\Delta W_i^*}{\Delta \tau} V^\tau = - \int_{\mathcal{S}(t)} (F_i^j - G_i^j) n_j dS \bigg|^\tau + Q_i V|^\tau \quad \text{for } i = 1, 2, \tag{18}$$

where the τ superscript denotes the previous (existing) solution or pseudo time-step and $\Delta\tau$ is calculated as in Eq. (36). Convective odd-even decoupling is circumvented by upwinding momenta in the flux term $\mathbf{F}$ via 3rd order interpolation as in the MUSCL scheme [43]. Similar upwinding is used to construct left and right states in compressible flow schemes [44], artificial-compressibility characteristic schemes [45], as well as in constructing consistent numerical fluxes [46], although the present scheme is somewhat simpler in being a one-sided extrapolation from the upwind node. As mentioned, this technique of stabilising the convective velocity has an advantage over the second-derivative characteristic-based stabilisation when considering flows with sharp velocity gradients, such as jet-flows and free-surface flows, since limiters may be applied to the extrapolated velocities, while the second-order stabililsation term can introduce spurious oscillations. ΔW_i^* is an auxiliary variable which is used in the second step as:

$$\frac{1}{c_\tau^2} \frac{p^{\tau+\Delta\tau_2} - p^\tau}{\Delta \tau_2} V^\tau = - \int_{\mathcal{S}(t)} \left[\rho u_k \Big|^\tau + \Delta\tau \left(\frac{\Delta W_k^*}{\Delta \tau} - \frac{\partial H_k^j}{\partial x_j} \bigg|^{\tau+\alpha\Delta\tau_2} \right) \right] n_k dS. \tag{19}$$

Here c_τ denotes the pseudo-acoustic velocity which is given by

$$c_\tau^2 = \max[\varepsilon^2; 1.2 u_j u_j]$$

and ε is typically chosen as $0.1 u_{\max}$ where $u_{\max}$ is the peak flow velocity in the domain [9]. By selecting $\alpha = 0$ we obtain an explicit solution method and with $\alpha = 1$ an implicit method. We shall solve the implicit form of the equation in a matrix-free manner by using a preconditioned GMRES routine to be described later. The pressure term above (involving $\mathbf{H}$) is discretised using a compact stencil to compute the derivative at the edge centre as described in Section 3.1. For efficiency, we only consider the tangential component of the compact derivative implicitly [13], i.e. we write

$$\left.\frac{\partial \mathbf{H}^j}{\partial x_j}\right|^{\tau+\alpha\Delta\tau_2} \approx \left.\overline{\mathbf{H}^j}_{mn}\right|_{tang}^{\tau+\alpha\Delta\tau_2} + \left.\overline{\mathbf{H}^j}_{mn}\right|_{norm}^{\tau}. \tag{20}$$

This implies that the Jacobian matrix at a node only involves that node and its nearest neighbours rather than next-nearest neighbours as well. On a 2D unstructured grid this implies a roughly sixfold decrease in the number of potentially nonzero entries in the matrix. On structured grids the Jacobian is exact. Since the equations are iterated to convergence, the inexact nature of the Jacobian does not affect the solution. However, if the mesh is highly skewed or if the system is stiff due to large discontinuities in the domain, the artificial compressibility provides a degree of under-relaxation to ensure robust convergence under these conditions.

The third and final incremental solution step written in semi-discrete form now follows:

$$\frac{W_i^{\tau+\Delta\tau} - W_i^{\tau}}{\Delta\tau} V^\tau = \frac{\Delta W_i^*}{\Delta\tau} V^\tau - \left.\int_{\mathcal{S}(t)} H_i^j n_j dS\right|^{\tau+\Delta\tau_2} \equiv R_i(\mathbf{W}) \quad \text{for } i = 1, 2. \tag{21}$$

Finally, the fluid mesh velocity $\mathbf{u}^*$ is calculated as follows:

$$u_j^* = \frac{3x_j^\tau - 4x_j^t + x_j^{t-\Delta t}}{2\Delta t}, \tag{22}$$

again using second order backward difference.

4.2.2. CNF-AC algorithm

In contrast to the preceding method which involves a 3 step solution procedure, the CNF-AC scheme addresses the continuity equation directly:

$$\frac{1}{c_\tau^2}\frac{p^{\tau+\Delta\tau_2} - p^\tau}{\Delta\tau_2} V^\tau = -\int_{\mathcal{S}(t)} \left[\left.\rho u_k\right|^\tau + \left.\frac{\Delta\tau}{\ell}(p_R - p_L) n_k\right|^{\tau+\alpha\Delta\tau_2} \right] n_k dS \tag{23}$$

Here, p_R denotes the pressure extrapolated from the node outside the control volume (node n in Fig. 1), p_L denotes the pressure extrapolated from the node inside it (node m in Fig. 1) and ℓ is the length of the edge connecting nodes

m and n. This stabilisation term is essentially the same as the 'consistent numerical flux' of Löhner *et al.* [46], but in a three-step pressure-projection method. Using constant interpolation (i.e. the nodal values for p_R and p_L) yields an effective second-order (Laplacian) stabilisation term, whereas an effective fourth-order stabilisation term results from the third-order accurate linear interpolation used in the MUSCL scheme [43]:

$$p_L = p_m + \frac{1}{2}\big[(1-\kappa)\nabla p|_m \cdot \boldsymbol{\ell} + \kappa(p_n - p_m)\big], \quad \kappa = 1/3. \tag{24}$$

Here, $\boldsymbol{\ell}$ is the vector from node m to node n. It can be shown that on a structured mesh, Eq. (23) is identical to (19) once the momentum equation has converged.

In the interest of computational efficiency, we use the implicit formulation ($\alpha = 1$) to approximate the interpolated pressures in such a way as to use only nearest-neighbour values of the increment $\Delta p \equiv p^{\tau+\Delta\tau_2} - p^{\tau}$ being solved for:

$$(p_R - p_L)^{\tau+\Delta\tau_2} \approx (p_R - p_L)^{\tau} + (\Delta p_n - \Delta p_m). \tag{25}$$

Via numerical experiementation it is found that the computational cost of additional iterations due to the inexact Jacobian is more than offset by the saving due to the greater sparsity of the Jacobian.

Having solved (23) for $p^{\tau+\Delta\tau_2}$, a momentum iteration follows as

$$\frac{W_i^{\tau+\Delta\tau} - W_i^{\tau}}{\Delta\tau} V^{\tau} = -\int_{\mathcal{S}(t)} (F_i^j - G_i^j) n_j dS\Big|^{\tau} - \int_{\mathcal{S}(t)} H_i^j n_j dS\Big|^{\tau+\Delta\tau_2} + Q_i V|^{\tau} \equiv R_i(\mathbf{W}), \tag{26}$$

where the nomenclature is as defined previously.

The CNF-AC scheme involves only two sweeps over the computational mesh in contrast to three for the UP-AC scheme and other pressure-projection schemes, e.g. [31], and is therefore computationally cheaper per iteration, as well as being more straightforward to implement. Comparative performance in terms of accuracy and rate of convergence will be investigated below. The CNF-AC scheme will produce the same results as the method of Löhner *et al.* [31] upon convergence, but differs in that it does not have a pressure-projection step, and includes artificial compressibility. As mentioned, the main virtue of the artificial compressibility term is to reduce the numerical stiffness of the pressure equation system, thus making it more amenable to solution with fast iterative implicit solvers, particularly in the presence of skewed grids or discontinuities in the domain.

4.2.3. Preconditioned GMRES Routine

As mentioned, we wish to solve Eqs (19) and (23) implicitly in order to overcome the time-step-size restriction on $\Delta\tau_2$. This is of particular importance when considering incompressible flow, as pressure waves propagate throughout the domain instantaneously. However, in order to scale efficiently to large problems, the procedure must be matrix-free. A popular approximate matrix solver is the Generalised Minimum Residual (GMRES) method of Saad and Schultz [47], which finds an optimum solution within a reduced subspace of the column space of a matrix, viz. the Krylov space. The Krylov vectors must however be preconditioned to yield suitably fast solution times [48].

The choice of the preconditioner to be applied to the GMRES Krylov-subspace vectors is of utmost importance. It must ensure a good condition number while, importantly, preserving the matrix-free nature of the solution scheme and being computationally efficient. Luo *et al.* [49] were the first to employ LU-SGS [50, 51, 52] as a preconditioner. As compared to its main competitor, Incomplete Lower-Upper decomposition (ILU), it does not require the storage of any part of the preconditioning matrix. Furthermore, it has been applied with great success to the nonlinear heat conduction equation (which is similar to the pressure-Poisson equation) [13], showing orders of magnitude performance improvement over Jacobi iterations and even over both GMRES and LU-SGS in isolation.

The discretised form of Eqs (19) and (23) can collectively be written in algebraic form as

$$\mathbf{A}\Delta\mathbf{p} = \mathbf{Res} \tag{27}$$

where the length of the vectors $\Delta\mathbf{p}$ and $\mathbf{Res}$ is equal to the number of nodes N, and $\mathbf{A}$ is a sparse $N \times N$ coefficient matrix. An outline of the LU-SGS preconditioned, restarted GMRES procedure follows. At each iteration the change in the primitive variable vector $\mathbf{p}$ is calculated from $\Delta\mathbf{p} = \mathbf{v}^l a_l$, $l = 1 \ldots L$. Here L denotes the number of preconditioned Krylov-space vectors $\mathbf{v}^l$, and the coefficients a_l are calculated such that they minimise the residual $\mathbf{Res}$ for a given set of Krylov-space vectors (the expression from which a_l is calculated follows below). The latter are calculated via the following GMRES procedure, which is invoked iteratively for a pre-specified number of GMRES iterations.

1. Initialise $\Delta\mathbf{p}_0 = \mathbf{0}$.

2. Starting Krylov-subspace vector:

$$\mathbf{v}^1 = \left[\mathbf{P}^{-1}(\mathbf{Res} - \mathbf{A}\Delta\mathbf{p}_0)\right] \left|\mathbf{P}^{-1}(\mathbf{Res} - \mathbf{A}\Delta\mathbf{p}_0)\right|^{-1} \tag{28}$$

   where $\mathbf{P}^{-1}$ denotes the preconditioning matrix, or $\mathbf{P}^{-1} \approx \mathbf{A}^{-1}$, and $|\cdot|$ denotes the Euclidian norm.

3. For $l = 1, 2, \ldots, L-1$, compute

$$\mathbf{w}^{l+1} = \mathbf{P}^{-1}\mathbf{A}\mathbf{v}^l - \sum_{k=1}^{l} h^{kl}\mathbf{v}^k, \quad h^{kl} = \mathbf{v}^k \cdot \mathbf{P}^{-1}\mathbf{A}\mathbf{v}^l \tag{29}$$

$$\mathbf{v}^{l+1} = \frac{\mathbf{w}^{l+1}}{|\mathbf{w}^{l+1}|} \tag{30}$$

4. The change in $\mathbf{p}$ is now calculated from the expression given previously, namely

$$\Delta\mathbf{p} = \Delta\mathbf{p}_0 + \mathbf{v}^l a_l \tag{31}$$

   where a_l are calculated such that the residual is minimised:

$$\left(\mathbf{A}\mathbf{v}^k\right) \cdot \left(\mathbf{A}\mathbf{v}^l\right) a_l = \left(\mathbf{A}\mathbf{v}^k\right) \cdot \left(\mathbf{Res} - \mathbf{A}\Delta\mathbf{p}_0\right) \tag{32}$$

5. Restart from Step 2 using $\Delta\mathbf{p}_0 = \Delta\mathbf{p}$ until required number of iterations complete.

We now outline the preconditioning procedure. The generic vector ω is preconditioned by the LU-SGS procedure by performing two computational sweeps over the mesh:

- Sweep 1: Calculate ω^* from $(\mathbf{D} + \mathbf{L})\omega^* = \omega$, and
- Sweep 2: Calculate ω^p from $(\mathbf{D} + \mathbf{U})\omega^p = \mathbf{D}\omega^*$,

where $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{U}$ are the strict lower, diagonal and upper parts of $\mathbf{A}$. Further, ω^* and ω^p respectively denote the intermediary and preconditioned versions of ω. Note that this procedure may be implemented in a completely matrix-free form.

As mentioned, a fixed number of GMRES iterations (restarts) are used rather than running the GMRES procedure to convergence; this dramatically reduces the computation time since the pressure equation is converged in concert with the momentum equations. After extensive numerical tests it was found that optimal convergence of the overall scheme is seemingly attained with three GMRES iterations (i.e. two restarts). This is because the residual is rapidly reduced at first, with diminishing returns for further restarts.

For fastest convergence of the pseudo-timestep iterations, $\Delta\tau_2$ should be made as large as possible, reducing the amount of artificial compressibility. On the other hand, if it is made too large the performance of GMRES suffers as the $\mathbf{A}$ matrix loses diagonal dominance. This is particularly important since the pressure equation is not solved to convergence for every iteration of the momentum equations. For the purposes of this work a scaling factor of order 10^5 was found to yield satisfactory performance, i.e. $\Delta\tau_2 = 10^5\Delta\tau$.

4.3. Solution Method: Solid

For the solid domain, the solution procedure simply involves the pseudo-temporal discretisation of Eqs (5) and (17). Traction boundary conditions are realised numerically by excluding external boundaries from the surface integral in (5) and adding in surface integrals of the applied tractions $\boldsymbol{\tau}$. Thus, the equation becomes

$$\frac{\partial}{\partial t}\int_{\mathcal{V}_0} \rho_0 v_i d\mathcal{V} = \int_{\mathcal{S}_{0\text{internal}}} P_{ij} n_j d\mathcal{S} + \int_{\mathcal{S}_{\text{boundary}}} \tau_i d\mathcal{S} + \int_{\mathcal{V}_0} Q_i d\mathcal{V} \equiv R_i(\mathbf{w}), \quad (33)$$

This work considers FSI systems for which the structure is considerably less computationally expensive than the fluid. Therefore, for pseudo-temporal discretisation a matrix-free Jacobi method is employed. For stability the single-step procedure proposed in [53] is used:

$$\begin{aligned} w_i^{\tau+\Delta\tau} &= w_i^\tau + \Delta\tau\left[v_i^\tau - \tfrac{3w_i^\tau - 4w_i^t + w_i^{t-\Delta t}}{2\Delta t} + \tfrac{1}{2}\Delta\tau R_i(\overline{\mathbf{w}}^\tau)/(\rho_0^{\tau+\Delta\tau} V_0^{\tau+\Delta\tau})\right] \\ v_i^{\tau+\Delta\tau} &= v_i^\tau + \Delta\tau R_i(\overline{\mathbf{w}}^\tau)/(\rho_0^{\tau+\Delta\tau} V_0^{\tau+\Delta\tau}) \end{aligned} \quad (34)$$

where V_0 is the volume of dual-cell $\mathcal{V}_0$ in the undeformed configuration and $\overline{w}^\tau$ denotes a projected displacement which is calculated as

$$\overline{w_i}^\tau = w_i^\tau + \Delta\tau v_i^\tau \quad (35)$$

for a stable iterative procedure.

4.4. Pseudo-timestep Calculations

The pseudo-timestep $\Delta\tau$ local to each computational cell is to be determined in the interest of optimal convergence while ensuring a stable solution process. An accurate estimation is therefore required for which the following expression is employed:

$$\Delta\tau = \mathrm{CFL}\left[\frac{|u_i - u_i^*| + c_{\text{unified}}}{\Delta x_i} + \kappa\frac{2\mu}{\rho\Delta x_i^2}\right]^{-1} \tag{36}$$

where CFL denotes the Courant-Friedrichs-Lewy number, Δx_i is the effective mesh spacing in direction i and κ is equal to 1 in the fluid domain and 0 in the solid domain. Further,

$$c_{\text{unified}} = \kappa c_\tau + (1-\kappa)\left(\sqrt{K/\rho_0} + \sqrt{\eta/\rho_0}\right). \tag{37}$$

Finally, in the case of transient analyses, we treat the dual-timestepping term implicitly, i.e. change τ to $\tau + \Delta\tau$ in (15) to (17), in order to maintain stability in cases where $\Delta\tau$ is comparable to or larger than Δt, as per [25].

4.5. Solution Procedure

To achieve simultaneous solution of the discretised fluid-solid equations in a manner which effects strong coupling, the following solution sequence is employed in an iterative fashion:

1. The fluid and solid discrete equations are solved concurrently via (18)–(21) or (23)–(26), and (33). Due to the accelleration of the fluid solver with GMRES, convergence of the fluid domain is typically faster than the solid. Therefore, in order to achieve convergence of the coupled system at an optimum rate, an adjustable number of iterations of Eq. (33) may be performed for every iteration of the fluid equations.
2. At the interface, the calculated fluid traction is applied to the solid boundary and the solid velocities to the fluid boundary. That is, the following equations for traction and velocity at the boundary are prescribed:

$$\begin{aligned} \tau_j &= pn_j - \sigma_{ij}n_i \\ \mathbf{u} &= \mathbf{v} \end{aligned} \tag{38}$$

   where $\mathbf{n}$ is the related normal unit vector pointing outward from the fluid domain.

3. The mesh is only moved if a solid mesh boundary node displacement exceeds a certain proportion of the element size (here chosen as 30%) or the residual of the fluid or solid mesh has been reduced to less than the convergence tolerance (a real-time-step is considered converged when the residual of all fluid and solid equations have dropped by at least 5 orders of magnitude).
4. The residuals are recalculated following the mesh movement, and if larger than the convergence tolerance steps 1–3 are repeated. This ensures a fully-converged solution.
5. If the residuals are below the convergence tolerance, the real-timestep is terminated, and the next time-step entered by proceeding to step 1.

The above is repeated for all real-time steps.

5. Parallelisation

Because of the fully matrix-free nature of the numerical method at solver sub-iteration level, the mesh can be decomposed into separate subdomains for parallel computing. Firstly, in composing the right-hand side, all loops are over edges, with the operation count for each edge being very nearly identical. Secondly, in performing the various sparse-matrix–vector dot products that make up the preconditioned GMRES algorithm, there is one dot product per *node*; however, the number of nonzero entries in the corresponding row of the Jacobian matrix is equal to the number of edges surrounding the node, plus one. Therefore, to balance the operation count for efficient parallelisation of not only the right-hand side calculation but also the LU-SGS/GMRES routine, the number of edges in each domain was balanced. This was done by weighting each node with an integer equal to the number of edges which connect to it, followed by applying the METIS library [54] to its connectivity graph. For interdomain communication, a system of "ghost nodes" is used in this work, with one layer of overlapping nodes at domain boundaries, where 'slave' nodes are updated with the values from corresponding 'master' nodes in the neighbouring domain. For efficiency, data transfer is consolidated into the largest possible packets and communicated using MPI.

An example of the overlap of domains due to the inclusion of ghost nodes is shown in Fig. 2. The inset shows extended overlap near the boundary due to the larger stencil required for certain types of boundary conditions.

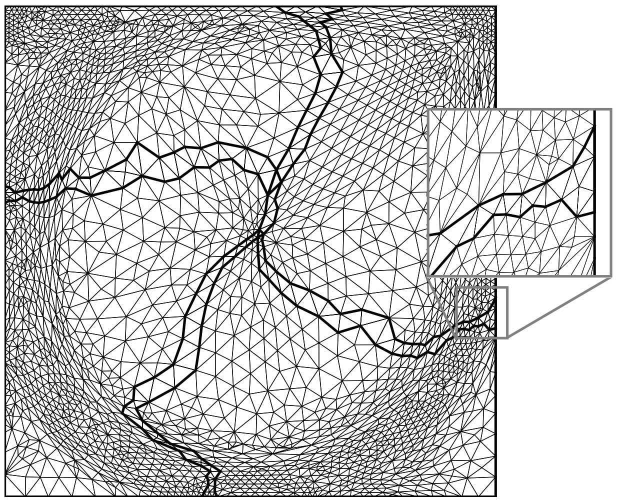

Figure 2: Example of domain decomposition showing one-element overlap due to ghost nodes and extended overlap near the boundaries.

The GMRES routine consists of sparse-matrix–vector products and dot products which may be computed in parallel to yield the serial computation result. However, the LU-SGS preconditioning steps are sequential in nature. They involve sweeps across the mesh, with each subsequent nodal value calculated from the newly updated values of the preceding ones. While this process may be vectorised on shared-memory architectures [55], on distributed-memory machines excessive communication would be required during each sweep over the mesh, severely impairing performance. In this work we have therefore elected to perform the LU-SGS preconditioning only within each parallel subdomain, eliminating the need for interdomain communication. Though this violates the formal serial nature of the procedure, being merely a preconditioning step it only impacts speed of convergence. However, this effect was found to diminish as the number of parallel domains increases, while being dwarfed by the performance gain due to parallelisation. Relevant results are presented below.

6. Application and Evaluation

6.1. Fluid solver validation

In order to validate the new UP-AC and CNF-AC flow solver algorithms we consider the popular lid-driven cavity test case. This consists of a square

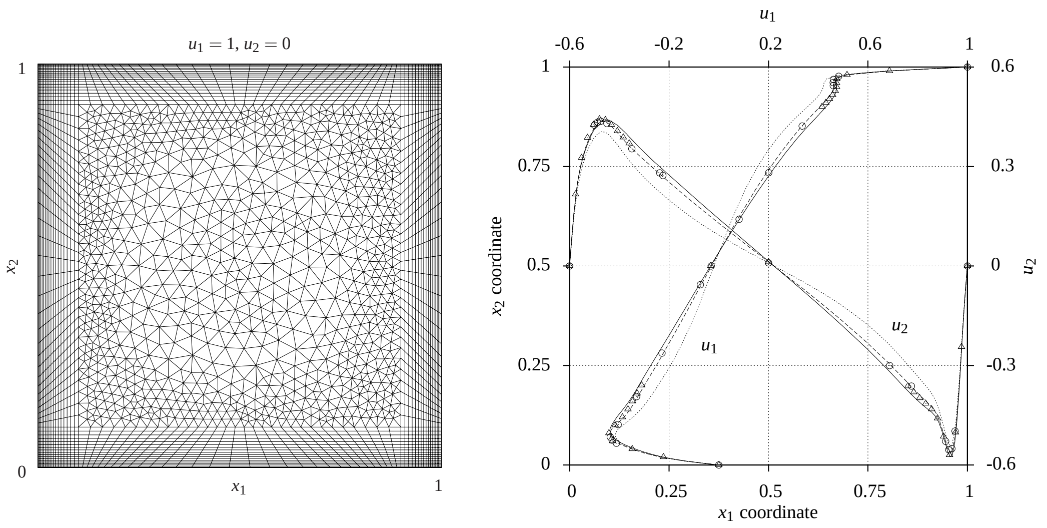

Figure 3: Lid-driven cavity. Left: mesh and boundary conditions. Sides and bottom are no-slip boundaries. Right: vertical velocity (u_2) through line $x_2 = 0.5$ and horizontal velocity (u_1) through line $x_1 = 0.5$. Datapoints are benchmark results from [56] (circles) and [57] (triangles). Solid line: UP-AC and CNF-AC results; dashed line: CBS-AC with global timestepping; dotted line: CBS-AC with local timestepping.

two-dimensional box with no-slip boundary conditions and a lid which moves at a horizontal velocity of 1. The geometry and mesh used are shown on the left-hand side of Fig. 3. As shown a hybrid-unstructured mesh containing 5046 elements is employed. This is to highlight the invariance of the developed methodology to element type. Here we consider the problem with a Reynolds number of 5 000.

To verify the spatial accuracy of the schemes we consider the steady-state solution; temporal discretisation will be validated by application to an FSI problem. The solutions predicted via the two schemes were found to be within 0.1% of each other, with selected vertical and horizontal velocity profiles comparing well to benchmark data [56, 57] (Fig. 3, solid line). As a comparison, results generated using the CBS-AC method are also shown in Fig. 3. For the CBS-AC scheme, the results show a dependence on timestep size due to the characteristic-based stabilisation term. When using local timestepping to accelerate convergence (dotted line), the results are not as accurate as the UP-AC or CNF-AC schemes for this mesh spacing; however, when using global timestepping, the value of $\Delta\tau$ is dramatically reduced in most regions, due to the very small cell size present near the boundaries. This

Mesh cells	Refinement factor r	Minimum centreline velocity (u_1) UP-AC	CNF-AC
8722		-0.443098	-0.444031
18254	1.447	-0.445640	-0.446006
36354	1.411	-0.446824	-0.446916
Convergence order p		1.94	1.97
Grid Convergence Index		0.0035	0.0026

Table 1: Results of mesh independence study for lid-driven cavity, verifying second-order spatial accuracy of the UP-AC and CNF-AC schemes.

results in a much more accurate solution, but at the expense of a significant reduction in speed of convergence. The UP-AC and CNF-AC methods, by contrast, do not suffer from this restriction.

Further, in order to verify the order of convergence of the spatial discretisation, a detailed mesh-independence study has been performed with three refined versions of the mesh in Fig. 3. Each mesh was refined by a factor of approximately $\sqrt{2}$, resulting in a doubling of the number of nodes. The exact refinement factor in each case was calculated as the square root of the ratio of the number of nodes in the two meshes. The order of convergence was then calculated based on measurement of the maximum negative velocity over the vertical centre-line of the cavity (the 'nose' at the bottom-left in Fig. 3). These values are given in Table 1, together with the order of convergence calculated by solving

$$\frac{u_2 - u_3}{r_{23}^p - 1} = r_{12}^p \frac{u_1 - u_2}{r_{12}^p - 1}$$

for p as per Roache [58], where $f_{1,2,3}$ are the measured values on the fine, medium and coarse meshes respectively, and $r_{12,23}$ are the associated mesh refinement factors as given in Table 1. Also given are the Grid Convergence Indices [59] for the finest meshes, whose small values indicate that the asymptotic mesh refinement region has indeed been reached. As shown, the order of convergence calculated is very close to 2, validating the claimed second-order spatial accuracy of the methods.

6.2. FSI accuracy assessment

The FSI problem we consider comprises a block with a flexible tail in cross-flow as shown in Fig. 4. This is a popular problem for testing new FSI algorithms [4, 36, 15, 60, 20]. In these publications, various different combinations of material properties, initial conditions and Reynolds numbers

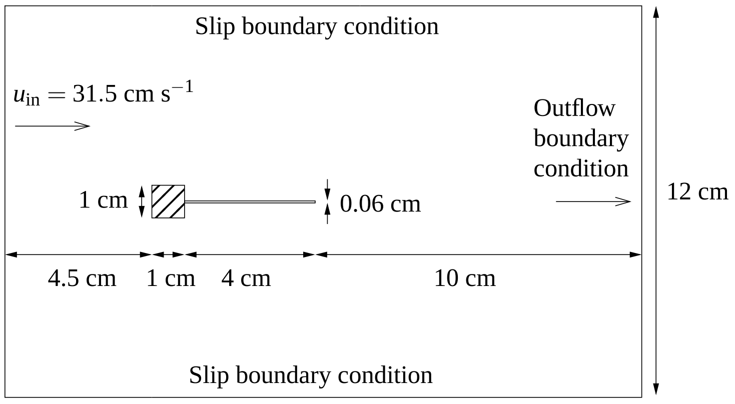

Figure 4: Block with flexible tail: geometry and boundary conditions

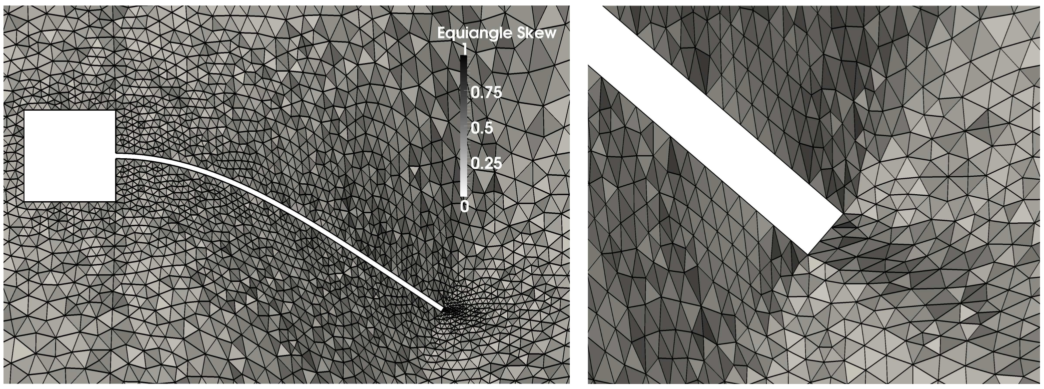

Figure 5: Block with flexible tail: Meshes at maximum deformation shaded according to element quality. Left: 6 000 cell mesh; max equiangle skew = 0.786, mean = 0.205. Right: Close-up of 50 000 cell mesh at beam tip. Max equiangle skew = 0.751, mean = 0.204.

have been considered. For the purpose of this work, we select a variant from [15]. The material properties are as follows: Incompressible fluid density is 1.18×10^{-3} g cm^{-3} and viscosity $\mu = 1.82 \times 10^{-4}$ g cm^{-1} s^{-1}. For the tail, density = 2.0 g cm^{-3}, Young's modulus $E = 2.0 \times 10^{6}$ g cm^{-1} s^{-2}, and Poisson's ratio ν is 0.35.

For the purpose of assessing accuracy, we compare the results obtained by the UP-AC and CNF-AC schemes to those of [15] via a mesh indepedence study. Three meshes with varying density were employed in the interest of demonstrating a mesh-independent solution. The fluid meshes consisted of 6 000, 25 000 and 50 000 fluid cells respectively. For the first series of analy-

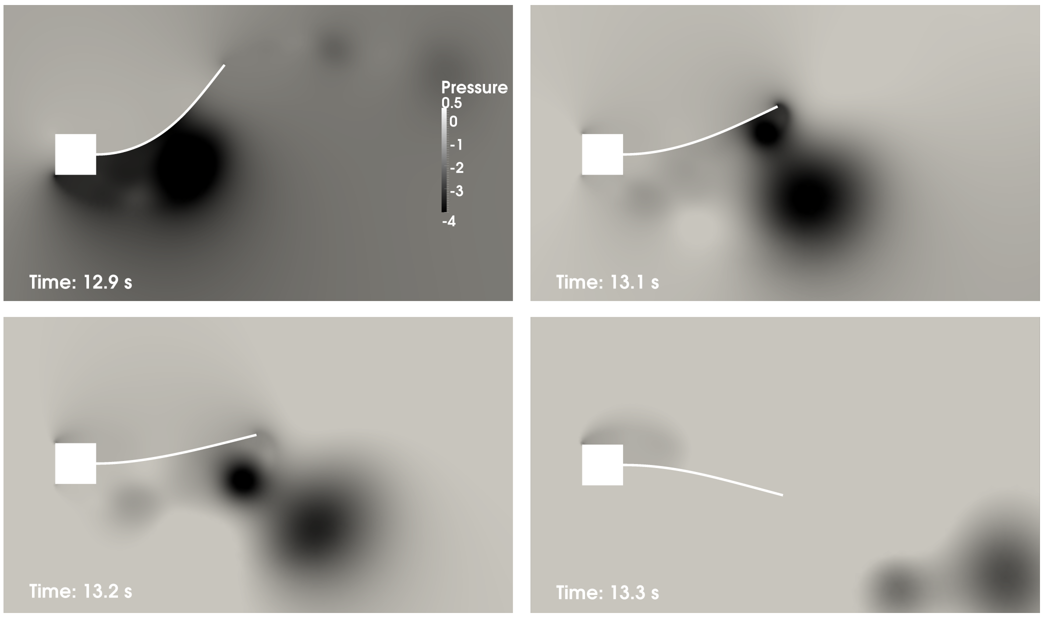

Figure 6: Block with flexible tail: Pressure contours showing primary and secondary vortices forming and detaching.

Time (s):	0–0.1	0.1–0.15	0.15–0.2	0.2–0.3	0.3–0.4	0.4–0.5	0.5–0.6
Force (g cm s^{-2}):	0	0.35	1.5	2.3	3.1	4.2	6.5
Time (s):	0.6–0.7	0.7–0.8	0.8–0.9	0.9–1.0	1.0–1.1	1.1–1.2	1.2–1.3
Force (g cm s^{-2}):	6.5	5.5	5.0	4.0	3.0	2.0	1.0

Table 2: Piecewise-constant force as a function of time applied to tip of beam to reproduce initial deflection in results of [15].

ses, structured solid meshes consisting of 12 elements through the thickness were employed. The number of elements along the length of the beam was respectively selected as 40, 120 and 160 for the three different fluid meshes, in order to line up with the fluid nodes. In addition, to evaluate solution independence with respect to the solid mesh, another analysis was performed with a 320×24 solid mesh – i.e. half the mesh spacing in both directions – in conjunction with the 50 000-element fluid mesh. The time-step size used in the aforementioned analyses was $\Delta t = 0.005$ s, and the accuracy of this was verified by comparing with an additional analysis with $\Delta t = 0.01$ s.

To allow the flow field to form without large transient shocks to the solid, the fluid inflow velocity was linearly ramped up to its final value of 31.5 cm s^{-1} within the first 0.02 s. In order to reproduce the initial deflection of the

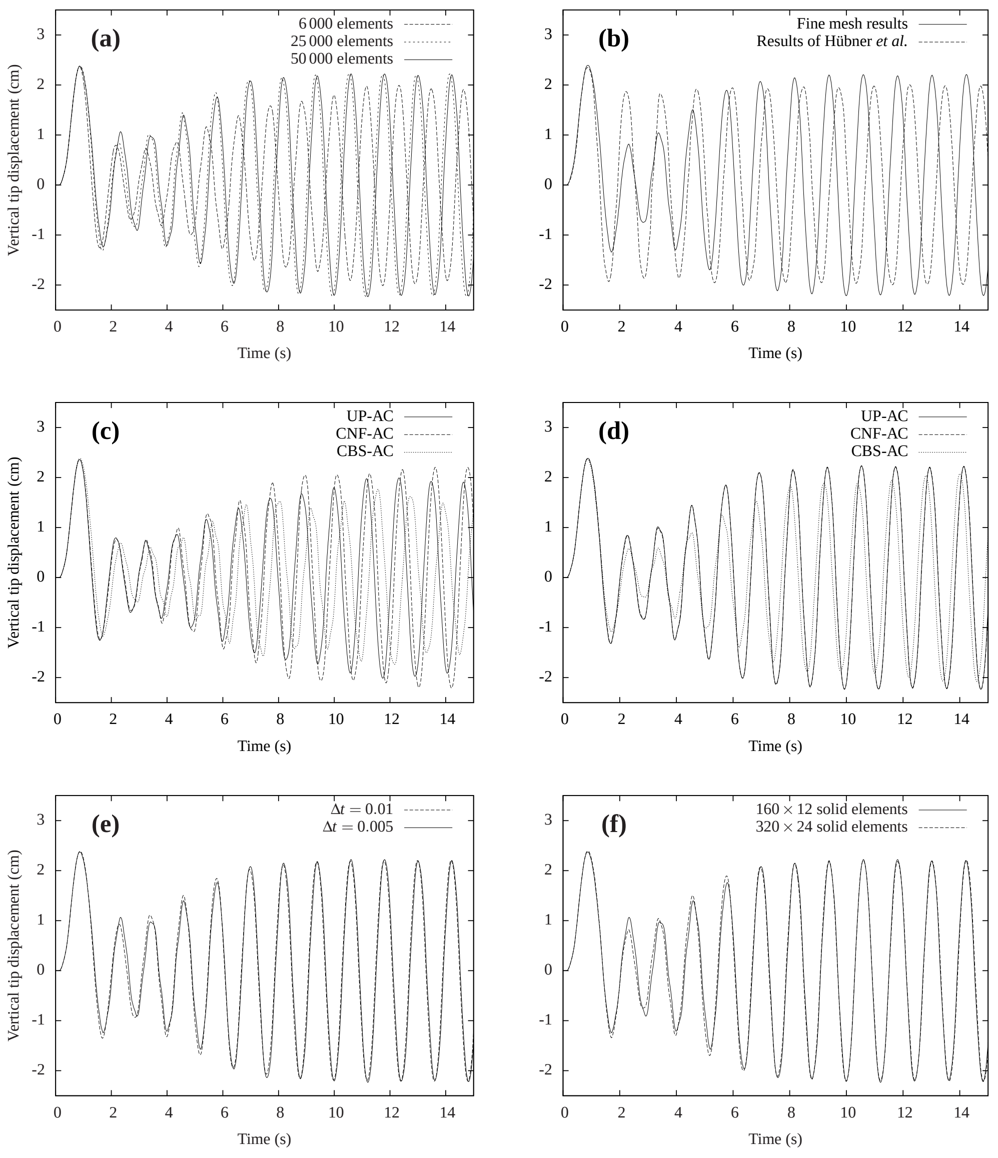

Figure 7: Tip deflection of block with flexible tail with initial tip load given in Table 2. (a) 6 000, 25 000 and 50 000 element fluid meshes with 40×12, 120×12 and 160×12 solid elements respectively, using $\Delta t = 0.005$ s and the UP-AC algorithm. (b) The same 50 000-element result compared with that of Hübner *et al.* [15]. (c-d) The two alternative pseudo-temporal discretisation schemes compared to CBS-AC, on (c) 6 000 and (d) 25 000 element fluid meshes respectively with solid mesh densities and Δt as in (a). (e) $\Delta t = 0.01$ and $\Delta t = 0.005$ s with 50 000 fluid elements and 160×12 solid elements. (f) 160×12 and 320×24 element solid meshes, 50 000 fluid elements and $\Delta t = 0.005$ s.

Fluid elements	Solid elements	Timestep (s)	Solution method	Amplitude (cm)	Frequency (Hz)
6 000	40×12	0.005	UP-AC	1.987	0.85896
			CNF-AC	2.152	0.85398
			CBS-AC	1.634	0.84354
25 000	120×12	0.005	UP-AC	2.219	0.83087
			CNF-AC	2.226	0.83017
			CBS-AC	2.061	0.83052
50 000	160×12	0.01	UP-AC	2.212	0.82922
50 000	160×12	0.005	UP-AC	2.214	0.82863
			CNF-AC	2.214	0.82877
			CBS-AC	2.127	0.82911
50 000	160×12	0.0025	UP-AC	2.219	0.82846
50 000	320×24	0.005	UP-AC	2.199	0.82512

Table 3: Amplitude and frequency of block-tail limit-cycle oscillation for various meshes and timestep sizes. These are calculated as the average values of the last seven cycles of oscillation.

tail seen in the results of Hübner *et al.* [15], we determined empirically the piecewise-constant tip-force as a function of time shown in Table 2. Figure 5 shows the coarse mesh in a state of maximum deflection and a close-up of the fine mesh near the tip of the beam. Shading is applied according to mesh quality of the deformed meshes, measured using the equiangle-skew metric defined as

$$Q = \max\left(\frac{\theta_{max} - \theta_e}{180^\circ - \theta_e}, \frac{\theta_e - \theta_{min}}{\theta_e}\right),$$

where θ_{max} and θ_{min} are the maximum and minimum cell internal angles and θ_e is the angle for which all internal angles are equal (60° for a triangle). Figure 6 shows snapshots of the pressure field. The large vortex underneath the tail is seen detaching and forming a secondary vortex of opposite rotational direction at the tip as it does so.

The calculated time-history of vertical tip displacement for the various meshes and time-step sizes is depicted in Fig. 7, with measurements of the steady-state amplitude and frequency give in Table 3. First, to evaluate mesh independence of our solution, the same simulations carried out on all three fluid meshes are compared in Fig. 7(a). The solution changes markedly when the mesh is refined from 6 000 to 25 000 elements (limit-cycle amplitude and frequency changing by 8.5% and 4% respectively), with little further change as the number of elements is doubled to 50 000 (<0.5% change in limit-cycle amplitude and frequency). Figure 7(b) compares the above results on the finest mesh with those of Hübner *et al.* [15], showing a discrepancy at first but similar steady-state amplitude and frequency (differing by 10% and 3.6% respectively). Figures 7(c) and (d) compare the UP-AC, CNF-AC and existing CBS-AC solution methods on the 6 000 and 25 000 element fluid meshes respectively. As seen, the three methods converge in the limit of mesh refinement – for the 25 000 element mesh, the amplitude and frequency of the UP-AC and CNF-AC methods are within 0.5% of each other, while on the 50 000 element mesh (not shown) they are within 0.2%. CBS-AC also appears to converge to the same values but is less accurate for a given mesh spacing; since local timestepping is being used in the interest of computational speed, this conclusion is in agreement with the that of the previous section.

To assess temporal accuracy the finest fluid mesh was used with the UP-AC algorithm and the analysis repeated with the time-step doubled from 0.005 s to 0.01 s and halved to 0.0025 s. As seen in Fig. 7(e), this has a negligible effect on the solution (the change in limit-cycle amplitude and frequency are again less than 0.5%). To assess the temporal order of convergence of the algorithm, measurements are taken of the average frequency of the last seven cycles of oscillation as shown in Table 3. The order of convergence is established from

$$p = \ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right) / \ln(r)$$

where $r = 2$ in this case and f_1, f_2 and f_3 are the frequency values for the smallest, intermediate and largest Δt values respectively [58]. This yields a convergence rate of $p = 1.85$, consistent with the time-discretisation being second-order accurate. The Grid Convergence Index for the medium timestep $\Delta t = 0.005$ is 0.00041, indicating that time-step-size convergence has been achieved to a good accuracy [59]. Since the form of the time discretisation is identical for the CNF-AC algorithm, we only consider UP-AC here. Finally, mesh independence of the solid mesh is ascertained by repeating the same analysis using the 50 000 node fluid mesh with a solid mesh of half the mesh spacing (320×24). Figure 7(f) shows that solid-mesh independence is likely, with a <1% change in limit-cycle amplitude and frequency. The above results are summarised in Table 3.

6.3. GMRES performance speedup

In this section, we analyse the performance improvement effected by the preconditioned GMRES solver in the context of the fully-coupled FSI problem considered above. As mentioned previously, the implicit solution of the

Mesh (fluid, solid)		6 000, 40×12	25 000, 120×12	50 000, 160×12
Parallel subdomains fluid+solid		2+2	12+4	24+8
	Jacobi	216.4 s (29937)	285.9 s (52070)	366.1 s (65223)
UP-AC	GMRES	4.26 s (233)	3.55 s (211)	4.48 s (248)
	Speedup	50.8 ×	80.5 ×	81.7 ×
	Jacobi	209.4 s (33383)	193.5 s (41263)	218.3 s (44693)
CNF-AC	GMRES	3.92 s (251)	3.19 s (221)	3.28 s (262)
	Speedup	53.4 ×	60.7 ×	66.6 ×

Table 4: Computation time and speedups for a representative timestep of the block-tail problem with various mesh densities. Number of iterations to convergence is shown in brackets.

pressure equation yields faster (but more costly) convergence per iteration of the fluid as compared to the solid mesh (for which Jacobi iteration is employed). To alleviate this bottleneck, we employ solid sub-iteration in such a manner that the fluid and solid residuals reduce at similar rates (to yield optimal convergence). For the block-tail problem considered here we have used 30 solid sub-iterations.

Table 4 shows the computation time, convergence performance and achieved GMRES speed-ups for both the UP-AC and CNF-AC pressure stabilisation algorithms. Here we have run the block-tail problem considered above for 1 second so as to reach a representative time-step, and then compared the time taken to converge the following timestep. Results for the three different mesh densities are shown. Speedups of a factor between 40 and 80 result, with the CNF-AC method converging somewhat faster than the UP-AC method. This is an interesting result as, although the CNF-AC method is more computationally efficient per iteration, one might expect it to take a greater number of iterations to converge due to the fact that it is not a projection method. However, as seen from Table 4, this is not the case and indicates that CNF-AC is in fact more efficient overall than UP-AC.

6.4. Parallel efficiency

The strong-scaling performance of the entire coupled solver was evaluated by assessing the time taken to complete the first time-step of the block-tail problem with a 50 000 element fluid mesh and a 2 000 element solid mesh. Calculations were perfomed on a Sun Microsystems Constellation cluster with 8-core Intel Nehalem 2.9 GHz processors and Infiniband interconnects at the Centre for High Performance Computing (CHPC), Cape Town.

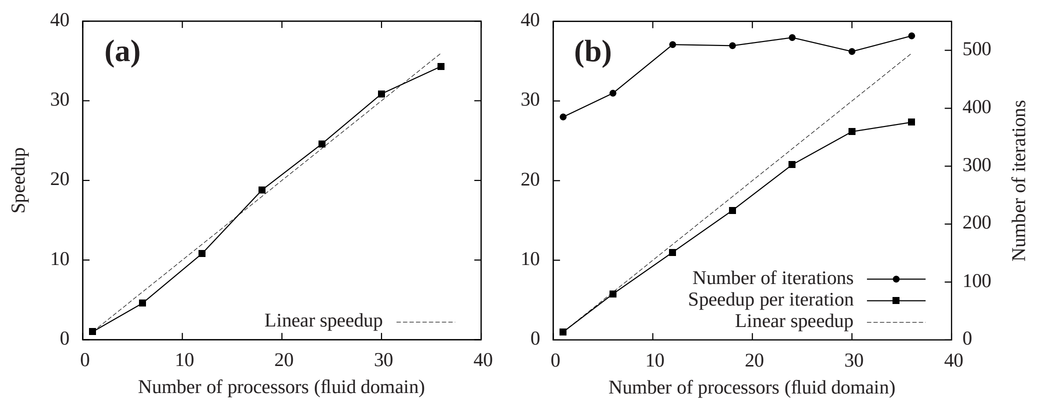

Figure 8: Parallelisation speed-up for the block-tail problem using the 50 000-element fluid mesh with (a) Jacobi solver and (b) preconditioned GMRES solver. The right axis of the latter depicts the increased number of pressure iterations required per time-step.

The results of the study are depicted in Fig. 8, where the number of iterations achieved per second has been normalised to the value for a single processor. Note that, as a consequence of the preconditioning being done separately on each parallel subdomain, the number of iterations taken to converge the time-step is not constant, as plotted in Fig. 8. As seen, there is an initial deterioration in the speed of convergence of about 20%, but this levels off as the number of sub-domains increases further. Also plotted is the speed-up per iteration, which is linear up to circa 30 processors for the problem considered, after which inter-core communication cost begins to dominate. This occurs more quickly for the GMRES than the Jacobi solver due to the increased communication involved.

7. Conclusion

A fast, strongly-coupled, partitioned, parallel FSI scheme was developed. Both fluid and solid domains were discretised via an edge-based finite volume methodology. In the case of the fluid domain, two novel matrix-free implicit fractional step methods were introduced viz. Upwind Pressure-Projection Artificial Compressibility (UP-AC) and Consistent Numerical Flux Artificial Compressibility (CNF-AC) methods. Both were found to be robust and accurate, with CNF-AC being somewhat more computationally efficient. These methods were shown to effect a significant improvement in accuracy compared to the Artificial Compressibility Characteristic-Based Split (CBS-AC) method. Retaining a small artificial compressibility source-term makes the methods more robust. It allows for the pressure equation to be approximately and cheaply solved at every iteration while enhancing convergence on skewed meshes and in the presence of numerical discontinuities. A dual-timestepping solution method was proposed where Jacobi iterations for the momentum equations are performed concurrently with the implicit iterative solution of pressures. This allowed for simultaneous, fully-coupled but partitioned solution of fluid and solid domains which was stable and robust for the problem considered. The fast and efficient matrix-free solution of pressure was done for the first time via a parallel LU-SGS preconditioned GMRES method. Remarkable FSI modeling performance was demonstrated. The achieved computational speed-ups ranged between 50 and 80 times, while preserving linear parallel computing performance.

Acknowledgements

The authors wish to thank the Centre for High Performance Computing (CHPC) in Cape Town for access to computing hardware. This work was funded by the Council for Scientific and Industrial Research (CSIR) on Thematic Type A Grant nr. TA-2009-013.

References

[1] R. M. Bennet, J. W. Edwards, An overview of recent developments in computational aeroelasticity, in: Proceedings of the 29th AIAA fluid dynamics conference, Albuquerque, NM, 1998.

[2] R. van Loon, F. N. van de Vosse, Special Issue: Fluid–structure interaction in biomedical applications, Int. J. Numer. Meth. Biomed. Engng. 26 (3-4) (2010) 273–275, doi:10.1002/cnm.1371.

[3] C. Taylor, C. Figueroa, Patient-Specific Modeling of Cardiovascular Mechanics, Annu. Rev. Biomed. Eng. 11 (1) (2009) 109–134, doi: 10.1146/annurev.bioeng.10.061807.160521.

[4] W. A. Wall, E. Ramm, Fluid–structure interaction based upon a stabilized (ALE) finite element method, in: S. Idelsohn, E. Oñate, E. Dvorkin (Eds.), Computational mechanics – new trends and applications, Proceedings of WCCM IV, CIMNE, Barcelona, 1998.

[5] K. J. Bathe, H. Zhang, S. Ji, Finite element analysis of fluid flows fully coupled with structural interactions, Comput. Struct. 72 (1-3) (1999) 1–16, doi:10.1016/S0045-7949(99)00042-5.

[6] E. H. Dowell, K. C. Hall, Modeling of fluid–structure interaction, Annu. Rev. Fluid Mech. 33 (1) (2001) 445–490, doi: 10.1146/annurev.fluid.33.1.445.

[7] R. Ohayon, C. E. Felippa, Special Issue: Advances in Computational Methods for Fluid–Structure Interaction and Coupled Problems, Comput. Method Appl. M. 190 (2001) 2977–3292, doi:10.1016/S0045-7825(00)00376-5.

[8] T. Tezduyar, Y. Bazilevs, Special issue on fluid–structure interaction, Comput. Mech. 43 (2008) 1–189, doi:10.1007/s00466-008-0317-8.

[9] A. G. Malan, R. W. Lewis, P. Nithiarasu, An Improved Unsteady, Unstructured, Artificial Compressibility, Finite Volume Scheme for Viscous Incompressible Flows: Part I. Theory and Implementation, Int. J. Numer. Meth. Engng. 54 (5) (2002) 695–714.

[10] A. G. Malan, R. W. Lewis, P. Nithiarasu, An Improved Unsteady, Unstructured, Artificial Compressibility, Finite Volume Scheme for Viscous Incompressible Flows: Part II. Application, Int. J. Numer. Meth. Engng. 54 (5) (2002) 715–729.

[11] A. G. Malan, R. W. Lewis, Modeling Coupled Heat and Mass Transfer in Drying Non-Hygroscopic Capillary Particulate Materials, Commun. Numer. Meth. Engng. 19 (9) (2003) 669–677.

[12] J. Pattinson, A. G. Malan, J. P. Meyer, A Cut-cell non-conforming Cartesian Mesh Method for Compressible and Incompressible Flow, Int. J. Numer. Meth. Engng. 72 (11) (2007) 1332–1354.

[13] A. G. Malan, J. P. Meyer, R. W. Lewis, Modelling Non-Linear Heat Conduction via a Fast Matrix-Free Implicit Unstructured-Hybrid Algorithm, Comput. Method Appl. M. 196 (45-48) (2007) 4495–4504.

[14] A. G. Malan, R. W. Lewis, An artificial compressibility CBS method for modelling heat transfer and fluid flow in heterogeneous porous materials, Int. J. Numer. Meth. Engng. 87 (1–5) (2011) 412–423, doi: 10.1002/nme.3125.

[15] B. Hübner, E. Walhorn, D. Dinkler, A monolithic approach to fluid–structure interaction using spacetime finite elements, Comput. Method Appl. M. 193 (2004) 2087–2104.

[16] C. J. Greenshields, H. G. Weller, A unified formulation for continuum mechanics applied to fluid-structure interaction in flexible tubes, Int. J. Numer. Meth. Engng. 64 (2005) 1575–1593.

[17] M. W. Gee, U. Küttler, W. A. Wall, Truly monolithic algebraic multigrid for fluid–structure interaction, Int. J. Numer. Meth. Engng. 85 (8) (2011) 987–1016, doi:10.1002/nme.3001.

[18] P. L. Tallec, J. Mouro, Fluid structure interaction with large structural displacements, Comput. Method Appl. M. 190 (2001) 3039–3067.

[19] D. P. Mok, W. A. Wall, Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures, in: W. A. Wall, K.-U. Bletzinger (Eds.), Trends in computational structural mechanics, Barcelona: CIMNE, 689–698, 2011.

[20] W. Dettmer, J. D. Peric, A computational framework for fluid–structure interaction: Finite element formulation and application, Comput. Method Appl. M. 195 (2006) 5754–79.

[21] W. A. Wall, S. Genkinger, E. Ramm, A strong coupling partitioned approach for fluid–structure interaction with free surfaces, Comput. Fluids 36 (2007) 169–183.

[22] U. Küttler, W. A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, Comput. Mech. 43 (2008) 61–72.

[23] M. von Scheven, E. Ramm, Strong coupling schemes for interaction of thin-walled structures and incompressible flows, Int. J. Numer. Meth. Engng. 87 (2011) 214–231.

[24] A. Jameson, Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings, AIAA Paper 91-1596.

[25] V. Venkatakrishnan, D. J. Mavriplis, Implicit Method for the Computation of Unsteady Flows on Unstructured Grids, J. Comput. Phys. 127 (1996) 380–397.

[26] S. V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York, 1980.

[27] A. J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, J. Comput. Phys. 2 (1967) 12–26.

[28] P. Nithiarasu, An efficient artificial compressibility (AC) scheme based on the characteristic based split (CBS) method for incompressible flow, Int. J. Numer. Meth. Engng. 56 (13) (2003) 1815–1845.

[29] P. Nithiarasu, An arbitrary Lagrangian Eulerian (ALE) formulation for free surface flows using the characteristic-based split (CBS) scheme, Int. J. Numer. Meth. Fl. 48 (2005) 1415–1428.

[30] R. Löhner, A fast finite element solver for incompressible flows, AIAA Paper 90-0398.

[31] R. Löhner, C. Yang, J. Cebral, F. Camelli, O. Soto, J. Waltz, Improving the speed and accuracy of projection-type incompressible flow solvers, Comput. Method Appl. M. 195 (2006) 3087–3109.

[32] P. Nithiarasu, C.-B. Liu, An artificial compressibility based characteristic based split (CBS) scheme for steady and unsteady turbulent incompressible flows, Comput. Method Appl. M. 195 (2006) 2961–2982.

[33] N. Massarotti, F. Arpino, R. W. Lewis, P. Nithiarasu, Explicit and semi-implicit CBS procedures for incompressible viscous flows, Int. J. Numer. Meth. Engng. 66 (2006) 1618–1640.

[34] P. I. Crumpton, P. Moinier, M. B. Giles, An Unstructured Algorithm for High Reynolds Number Flows on Highly Stretched Meshes, in: C. Taylor, J. T. Cross (Eds.), Numerical Methods in Laminar and Turbulent Flow, Pineridge Press, 561–572, 1997.

[35] Y. Zhao, B. Zhang, A High-Order Characteristics Upwind FV Method for Incompressible Flow and Heat Transfer Simulation on Unstructured Grids, Int. J. Numer. Meth. Engng. 37 (1994) 3323–3341.

[36] G. Xia, C.-L. Lin, An unstructured finite volume approach for structural dynamics in response to fluid motions, Comput. Struct. 86 (2008) 684–701.

[37] R. Suliman, O. Oxtoby, A. G. Malan, S. Kok, An enhanced matix-free edge-based finite volume approach to model structures, in: 7th South African Conference on Computational and Applied Mechanics (SACAM10), Pretoria, 399–406, 10-13 January 2010.

[38] A. Slone, K. Pericleous, C. Bailey, M. Cross, Dynamic fluid–structure interaction using finite volume unstructured mesh procedures, Comput. Struct. 80 (2002) 371–390.

[39] P. D. Thomas, C. K. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA J. 17 (1979) 1030–1037.

[40] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, Comput. Method Appl. M. 134 (1996) 71–90.

[41] K. A. Sørensen, O. Hassan, K. Morgan, N. P. Weatherill, Agglomerated Multigrid on Hybrid Unstructured Meshes for Compressible Flow, Int. J. Numer. Meth. Fl. 40 (3-4) (2002) 593–603.

[42] C. Förster, W. A. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, Comput. Method Appl. M. 196 (1997) 1278–1293.

[43] B. van Leer, Towards the Ultimate Conservative Difference Scheme IV: A New Approach to Numerical Convection, J. Comput. Phys. 23 (1977) 276.

[44] J. Blazek, Computational Fluid Dynamics: Principles and Applications, Elsevier Science, Oxford, first edn., 2001.

[45] D. Drikakis, P. A. Govatsos, D. E. Papantonis, A characteristic-based method for incompressible flows, Int. J. Numer. Meth. Fl. 19 (1994) 667.

[46] R. Löhner, C. Yang, E. Oñate, S. Idelssohn, An unstructured grid-based, parallel free surface solver 31 (1999) 271–293.

[47] Y. Saad, M. H. Schultz, GMRES: A Generalized Mimimal Residual Algorithm for Solving Nonsymmetric Linear Systems, Siam J. Sci. Stat. Comp. 7 (3) (1986) 856–869.

[48] R. Löhner, Applied CFD Techniques, John-Wiley and Sons Ltd., Chichester, 2001.

[49] H. Luo, J. D. Baum, R. Löhner, A Fast, Matrix-Free Implicit Method for Compressible Flows on Unstructured Grids, J. Comput. Phys. 146 (1998) 664–690.

[50] A. Jameson, S. Yoon, Lower-upper Implicit Schemes with Multiple Grids for the Euler Equations, AIAA J. 25 (7).

[51] I. Men'shov, Y. Nakamura, Implementation of the LU-SGS Method for an Arbitrary Finite Volume Discretization, in: 9th Japanese Symposium on CFD, Chuo University, Tokyo, Japan, 123–124, 1995.

[52] M. Soetrisno, S. T. Imlay, D. W. Roberts, A Zonal Implicit Procedure for Hybrid Structured-Unstructured Grids, AIAA Paper 94-0645.

[53] O. C. Zienkiewicz, R. L. Taylor, The Finite Element Method: Volume 1 – The Basis, Butterworth-Heinemann, Oxford, fifth edn., 2000.

[54] G. Karypis, V. Kumar, A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs, SIAM J. Sci. Comput. 20 (1) (1999) 359–392.

[55] D. Sharov, K. Nakahashi, Reordering of Hybrid Unstructured Grids for Lower–Upper Symmetric Gauss–Seidel Computations, AIAA J. 36 (3) (1997) 484–486.

[56] U. Ghia, K. N. Ghia, C. T. Shin, High-Re Solutions for Incompressible Flow using the Navier-Stokes Equations and a Multigrid Method, J. Comput. Phys. 48 (1982) 387–411.

[57] E. Erturk, T. C. Corke, C. Gökçöl, Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers, Int. J. Numer. Meth. Fl. 48 (2005) 747–774.

[58] P. Roache, Verification of codes and calculations, AIAA J. 36 (1998) 696–702.

[59] P. Roache, Quantification of uncertianty in computational fluid mechanics, Annu. Rev. Fluid Mech. 29 (1997) 123–60.

[60] P. R. F. Teixeira, A. M. Awruch, Numerical simulation of fluid–structure interaction using finite element method, Appl. Math. Model. 34 (2005) 249–273.