

# Projection-based variational multiscale method for incompressible Navier–Stokes equations in time-dependent domains

Birupaksha Pal and Sashikumaar Ganesan<sup>\*,†</sup>

*Department of Computational and Data Sciences, Indian Institute of Science, Bangalore 560012, India*

## SUMMARY

A variational multiscale method for computations of incompressible Navier–Stokes equations in time-dependent domains is presented. The proposed scheme is a three-scale variational multiscale method with a projection-based scale separation that uses an additional tensor valued space for the large scales. The resolved large and small scales are computed in a coupled way with the effects of unresolved scales confined to the resolved small scales. In particular, the Smagorinsky eddy viscosity model is used to model the effects of unresolved scales. The deforming domain is handled by the arbitrary Lagrangian–Eulerian approach and by using an elastic mesh update technique with a mesh-dependent stiffness. Further, the choice of orthogonal finite element basis function for the resolved large scale leads to a computationally efficient scheme. Simulations of flow around a static beam attached to a square base, around an oscillating beam and around a plunging aerofoil are presented. Copyright © 2016 John Wiley & Sons, Ltd.

Received 10 September 2015; Revised 4 July 2016; Accepted 27 September 2016

KEY WORDS: variational multiscale; incompressible flows; Navier–Stokes equations; arbitrary Lagrangian–Eulerian; time-dependent domains; turbulent flows

## 1. INTRODUCTION

Incompressible flows are modeled by the time-dependent incompressible Navier–Stokes equations (NSE). In general, the Navier–Stokes equations are used for both laminar and turbulent flows, although the flow states are quite different from a physical point of view. The occurrence of one or the other state strongly depends on the Reynolds number associated with the flow. Turbulent flows are highly unsteady flows, where the main velocity field is superimposed by random fluctuations. In typical engineering applications, turbulent flows are prevalent, among other things, because of their positive features, such as more effective transport and mixing ability with respect to a comparable laminar flow. The analytical solution of NSE is deficient, in particular with regard to the turbulent flow regime. There exist no analytical solution for turbulent flows even in simple geometries [1, 2]. It leaves out the numerical approach as the only viable alternative. Besides Reynolds-averaged Navier–Stokes, the most popular conceptual alternatives for the numerical simulation of turbulent flows are the direct numerical simulation (DNS), the large eddy simulation (LES), and the variational multiscale method (VMS), which is a novel approach based on LES. The concepts of DNS and the classical LES in its basic form struggle with different challenges in terms of computational complexity, accuracy, and/or efficiency.

Direct numerical simulation is the most straightforward approach, and it attempts to solve NSE numerically by resolving all scales in the flow, without any additional turbulence model. However, the challenge lies in the broad range of length and time scales associated with the turbulent flow,

<sup>\*</sup>Correspondence to: Sashikumaar Ganesan, Department of Computational and Data Sciences, Indian Institute of Science, Bangalore 560012, India.

<sup>†</sup>E-mail: sashi@cds.iisc.ac.in

which makes DNS computationally infeasible in most of the cases. An alternative is LES, where the strategy is to compute the large flow structures (resolved scales) and model the effect of the small flow structures (unresolved scales) on the large scales. In general, LES relies on a filter to separate the resolved and unresolved scales. The major challenges in LES are firstly, the choice of an appropriate filter function; secondly, it involves commutation error or the error incurred in restricting the theory of LES on unbounded domains to a bounded domain; and thirdly, the effects of unresolved scales are incorporated into the entire resolved scales through a turbulence model, whereas adding its effects only to a part of the resolved scales would have been physically more sound. More details of the classical LES can be found in [3–6].

Variational multiscale method is a novel approach based on the classical LES for simulating turbulent flows. In contrast to the classical LES, one of the main features of VMS is its mathematical consistency that enables transition to DNS when more and more scales are resolved in computations. The theoretical framework of VMS has initially been established in [7], and has further been developed for problems in computational mechanics [8]. VMS allows to separate the entire range of flow scales into a predefined number of scales (mostly two or three), and hence tailored numerical schemes can be used for different flow scales. Two important aspects of VMS are (i) scale separation based on projection into appropriate spaces, rather than using a filter function as in LES thereby doing away with the commutation errors, and (ii) the direct influence of the turbulence model is confined to the resolved small scales, whereas the resolved large scales are devoid of any direct influence of the modeling term. Nevertheless, the resolved large scales are still influenced indirectly by the sub-grid scale model because of the inherent coupling of all scales.

Several variants of VMS implementations have been proposed in the literature (e.g., [9–14]). The residual-based VMS and the projection-based VMS are the most commonly used approaches. The basic idea of the residual-based VMS approach is to seek the solution of the fine-scale equation, which is written in terms of the large scale residual, in an enriched solution space, and then use it in the large-scale approximation. For instance, the small-scale equation is solved using Green's functions in [15]. Alternatively, the small-scale solution is approximated in a space enriched with bubble functions, and the solution is transferred to the large-scale equation in the form of a stability tensor in [16, 17]. Further, this approach maybe considered as a variant of the stabilization methods [16, 18], such as SUPG. Contrarily, in projection-based VMS approaches, the influence of the sub-grid scales on the resolved scales are incorporated by a model, and subsequently, the resolved scales are computed (e.g., [11, 19]).

Apart from other challenges associated with simulations of turbulent flows, moving boundaries/interfaces make computations more challenging. Interface tracing/capturing schemes (e.g., [20–24]) can be classified into two categories: (i) Eulerian approach and (ii) Lagrangian approach. In the Eulerian approach (e.g., level-set, volume of fluids, front-tracking, etc.), the computational grid is fixed and the fluid flows through the grid. Because the moving boundaries/interfaces are not resolved by the mesh, computation of surface forces and suppressing spurious velocities are challenging in the Eulerian approach. On the other hand, moving boundaries/interfaces are resolved by a moving mesh in the Lagrangian approach, for example, arbitrary Lagrangian–Eulerian (ALE). Hence, the solution can be approximated accurately by using the ALE approach. However, handling topological changes is challenging in the ALE approach. Nevertheless the ALE scheme is the appropriate choice when the deformation of the domain is relatively small.

In this paper, the projection-based VMS using the ALE approach is presented for flows in time-dependent domains. Though the residual-based VMS has been extended in [16–18, 25] for the flows in time-dependent domains, to the best of the authors knowledge, the projection-based VMS has not been presented for flows in time-dependent domains. Because the mesh velocity is part of the nonlinear convective term in the ALE form of NSE, the mesh velocity has to be calculated in each nonlinear iteration step of NSE when the domain is *a priori* unknown. Hence, the mesh velocity is needed for both the large-scale and fine-scale equations in the residual-based VMS, whereas this is not the case in the projection-based VMS as we have only the resolved small-scale equations. Moreover, unlike in residual-based VMS, the projection-based VMS does not require an enrichment of the underlying resolved small scales space by higher order polynomial bubble functions. Because

the bubble functions generally vanish on the element boundaries, resolved small scales cannot cross the mesh cell boundaries in residual-based VMS, and therefore this behavior can be avoided in the projection-based VMS.

In the proposed projection-based ALE-VMS scheme, we consider a three-scale decomposition of the flow fields: resolved large scales, resolved small scales, and unresolved (small) scales or the sub-grid scales. Further, the separation of the resolved large and resolved small scales is achieved by a projection, whereas the Smagorinsky turbulence model [19, 26, 27] is used to incorporate the effects of unresolved scales on the resolved small scales. Moreover, the resolved small and large scales are discretized using continuous and discontinuous finite element spaces, respectively. The choice of discontinuous spaces for the resolved large scales allows to reformulate the ALE-VMS system matrix into the standard form of the NSE system matrix. Thus, an existing Navier–Stokes solver needs only a minor modification to implement the projection-based ALE-VMS scheme.

The remainder of the paper is organized as follows: In Section 2, the NSE and its ALE formulation are given. Section 3 describes the projection-based three-scale ALE-VMS scheme. Further, the variational form of the Navier–Stokes equations and its spatial discretization are given in this section. Section 4 describes the temporal discretization and the mesh moving technique for deforming meshes. Section 5 discusses the matrix manipulation and other implementation aspects of the scheme. In Section 6, simulations of flow around a stationary beam attached to a square base, flow around a deforming beam, and flow around a plunging aerofoil are presented. Finally, concluding remarks are summarized in Section 7.

## 2. MATHEMATICAL MODEL AND ITS ARBITRARY LAGRANGIAN–EULERIAN FORMULATION

We consider an incompressible fluid flow around a moving/deforming obstacle in a three-dimensional channel (see Figure 1 for a cross-sectional schematic view). The incompressible fluid flow in the considered geometry is described by the time-dependent incompressible NSE:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} - \frac{2}{\text{Re}} \nabla \cdot \mathbb{D}(\mathbf{u}) + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p &= 0, \text{ in } (0, T] \times \Omega_t, \\ \nabla \cdot \mathbf{u} &= 0, \text{ in } (0, T] \times \Omega_t. \end{aligned} \tag{1}$$

Here,  $\mathbf{u} = (u_1, u_2, u_3)^T$  is the fluid velocity,  $p$  is the pressure in the fluid,  $T$  is a given final time,  $\Omega_t \subset \mathbb{R}^3$ ,  $t \in (0, T]$  is the channel with deforming/moving boundary  $\Gamma_S$  inside the domain,  $\Gamma_{in}$  and  $\Gamma_{out}$  are the inflow and the outflow boundaries, respectively. Further,  $\Gamma_{wall}$  represents the lower, upper and lower boundaries of the channel. The NSE are closed with the initial condition

$$\mathbf{u}(0, \cdot) = \mathbf{u}_0 \text{ in } \Omega_0$$

and boundary conditions

$$\begin{aligned} \mathbf{u}(t, x) &= \mathbf{u}_D \text{ on } (0, T] \times \Gamma_{in}, \\ \mathbf{u}(t, x) &= \mathbf{0} \text{ on } (0, T] \times (\Gamma_{wall} \cup \Gamma_S), \\ \left( \frac{2}{\text{Re}} \mathbb{D}(\mathbf{u}) - p \mathbb{I} \right) \cdot \mathbf{n} &= \mathbf{0} \text{ on } (0, T] \times \Gamma_{out}. \end{aligned}$$

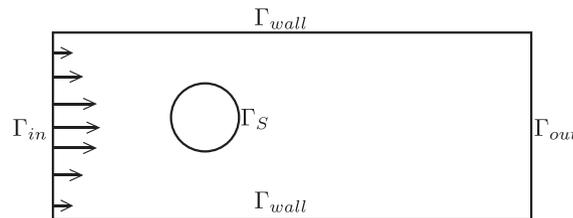


Figure 1. Computational domain for a fluid flow around an obstacle.

Here,  $\mathbf{u}_0$  is a given initial velocity and  $\mathbf{u}_D = (\mathbf{u}_{in}, 0)^T$  a given inlet velocity,  $\mathbb{I}$  is the identity tensor and  $\mathbf{n}$  is the outward normal to the boundary  $\Gamma_{out}$ . Further, the velocity deformation tensor and the Reynolds number are defined by

$$\mathbb{D}(\mathbf{u}) = \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2} \quad \text{and} \quad \text{Re} = \frac{\rho U L}{\mu},$$

where  $\rho$  is the density of the fluid,  $\mu$  is the dynamic viscosity,  $U$  and  $L$  are characteristic velocity and length scales, respectively.

### 2.1. Arbitrary Lagrangian–Eulerian formulation

In order to handle the time-dependent domain, we now derive an ALE form of the NSE (1). Let  $\hat{\Omega}$  be a reference domain, and define a family of bijective ALE mappings

$$\mathcal{A}_t : \hat{\Omega} \rightarrow \Omega_t, \quad \mathcal{A}_t(Y) = x(Y, t), \quad t \in (0, T).$$

The reference domain  $\hat{\Omega}$  can simply be the initial domain  $\Omega_0$  or the previous time-step domain when the deformation in the domain is large. Next, for a function  $v \in C^0(\Omega_t)$  on the Eulerian frame, we define their corresponding function  $\hat{v} \in C^0(\hat{\Omega})$  on the ALE frame as

$$\hat{v} : \hat{\Omega} \times (0, T) \rightarrow \mathbb{R}, \quad \hat{v} := v \circ \mathcal{A}_t, \quad \text{with} \quad \hat{v}(Y, t) = v(\mathcal{A}_t(Y), t).$$

Further, the time derivative on the ALE frame is defined by

$$\left. \frac{\partial v}{\partial t} \right|_Y : \Omega_t \times (0, T) \rightarrow \mathbb{R}, \quad \left. \frac{\partial v}{\partial t} \right|_Y(x, t) = \frac{\partial \hat{v}}{\partial t}(Y, t), \quad \text{with} \quad Y = \mathcal{A}_t^{-1}(x).$$

We now apply the chain rule to the time derivative of  $v \circ \mathcal{A}_t$  on the ALE frame to get

$$\left. \frac{\partial v}{\partial t} \right|_Y = \frac{\partial v}{\partial t}(x, t) + \left. \frac{\partial x}{\partial t} \right|_Y \cdot \nabla_x v = \frac{\partial v}{\partial t} + \frac{\partial \mathcal{A}_t(Y)}{\partial t} \cdot \nabla_x v = \frac{\partial v}{\partial t} + \mathbf{w} \cdot \nabla_x v,$$

where  $\mathbf{w}$  is the domain velocity. Using this relation to the NSE (1) to account for the deformation in the domain, we get the ALE form of the NSE as

$$\begin{aligned} \left. \frac{\partial \mathbf{u}}{\partial t} \right|_Y - \frac{2}{\text{Re}} \nabla \cdot \mathbb{D}(\mathbf{u}) + ((\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u}) + \nabla p = 0, \\ \nabla \cdot \mathbf{u} = 0. \end{aligned} \tag{2}$$

Note that the main difference between (1) and (2) is the additional domain velocity  $\mathbf{w}$  in the ALE form that accounts for the deformation in the domain. The ALE form (2) can be viewed as a generalized form of NSE, because the Lagrangian form of NSE can be obtained by setting  $\mathbf{w} = \mathbf{u}$  and the Eulerian form of NSE can be obtained by setting  $\mathbf{w} = 0$ .

### 2.2. Variational form of the arbitrary Lagrangian–Eulerian equation

To derive the variational form of the ALE-NSE (2), let us define the solution spaces with the ALE mapping

$$\begin{aligned} V_0(\Omega) &:= \{ \mathbf{v} \in (H^1(\Omega))^3 : \mathbf{v} = 0 \text{ on } \Gamma_{in} \cup \Gamma_{wall} \cup \Gamma_S \}, \\ V_\Gamma(\Omega) &:= \{ \mathbf{v} \in (H^1(\Omega))^3 : \mathbf{v} = 0 \text{ on } \Gamma_{wall} \cup \Gamma_S \}, \\ V &:= \left\{ \mathbf{v} \in V_\Gamma(\Omega_t) : \mathbf{v} : (0, T] \times \Omega_t \rightarrow \mathbb{R}^3, \mathbf{v} = \hat{v} \circ A_t^{-1}, \hat{v} \in V_\Gamma(\hat{\Omega}) \right\}, \\ Q &:= \left\{ q \in L^2(\Omega_t) : q : (0, T] \times \Omega_t \rightarrow \mathbb{R}, q = \hat{q} \circ A_t^{-1}, \hat{q} \in L^2(\hat{\Omega}) \right\}. \end{aligned}$$

Now, multiplying Equation (2) with a test function  $\mathbf{v} \in V_0$ , and the continuity equation with a test function  $q \in Q$ , and after applying integration by parts to the higher order derivative term, the variational form of the ALE Equation (2) and the continuity equation read:

For given  $\mathbf{w}$ ,  $\mathbf{u}_0$ , and  $\Omega_0$ , find  $\mathbf{u} \in L^2(0, T; V)$ ,  $p \in L^2(0, T, Q)$  such that

$$A(\mathbf{u}; (\mathbf{u}, p); (\mathbf{v}, q)) - b(\mathbf{w}, \mathbf{u}, \mathbf{v}) = 0 \quad (3)$$

$$(q, \nabla \cdot \mathbf{u}) = 0 \quad (4)$$

for all  $(v, q) \in V_0 \times Q$ , where

$$A(\mathbf{u}; (\mathbf{u}, p); (\mathbf{v}, q)) = \left( \frac{\partial \mathbf{u}}{\partial t} \Big|_Y, \mathbf{v} \right) + b(\mathbf{u}, \mathbf{u}, \mathbf{v}) + \frac{2}{\text{Re}} (\mathbb{D}(\mathbf{u}), \mathbb{D}(\mathbf{v})) - (p, \nabla \cdot \mathbf{v}),$$

$$b(\mathbf{w}, \mathbf{u}, \mathbf{v}) = (\mathbf{w} \cdot \nabla \mathbf{u}, \mathbf{v}).$$

Here,  $(\cdot, \cdot)$  denotes the  $L^2$ -inner product in  $\Omega_t$ . Further,  $\mathbf{u} \in L^2(0, T; V)$  implies that the mapping  $t \mapsto \mathbf{u}(t)$  is continuous.

### 3. VARIATIONAL MULTISCALE METHOD FOR ARBITRARY LAGRANGIAN–EULERIAN NAVIER–STOKES EQUATIONS

#### 3.1. Variational multiscale method in time-dependent domains

The solution of the variational form (3) can be obtained numerically by a finite dimensional discretization. However, the discretization does not capture the fluid flow scales that are smaller than the mesh size (discretization parameter). Because the turbulent kinetic energy dissipates at the Kolmogorov length scale, the mesh has to be fine enough (at least up to the Kolmogorov length scale) in order to capture the dynamics of the turbulent flows. This approach is referred to as DNS. Nevertheless, the choice of the mesh relies on the model (1), and in general, a finer mesh requires enormous memory and computing power. To overcome this huge demand, a reasonable mesh with a model to correct the discretization error is used, and this model can be referred to as a turbulence model. One of the simple and attractive approaches for turbulence modeling is the recently proposed two-scale VMS [28]. The basic idea in VMS is to decompose the flow fields into resolved (large) and unresolved (small) scales, and incorporate the effects of small scales into the solution of the large-scales by a model. The resolved scale is finite dimensional, and it can be represented by a standard finite element space. The modeled unresolved scale, also known as the sub-grid scale, is the remnant of the solution and is infinite dimensional.

A three-scale VMS having the same model equations as the two-scale VMS but with different modeling assumptions has been proposed in [9]. In this paper, we apply the three-scale decomposition, wherein the resolved solution space is further partitioned into resolved large and resolved small scales, that is,

$$V = \bar{V} \oplus \tilde{V} \oplus \hat{V} \quad \text{and} \quad Q = \bar{Q} \oplus \tilde{Q} \oplus \hat{Q},$$

and the velocity test space as

$$V_0 = \bar{V}_0 \oplus \tilde{V}_0 \oplus \hat{V}_0.$$

In the aforementioned equations, the bar, the tilde, and the hat over the spaces represent the resolved large, the resolved small, and the unresolved small scales, respectively. Consequentially, the functions  $\mathbf{u} \in V$  and  $q \in Q$  can be written as

$$\mathbf{u} = \bar{\mathbf{u}} + \tilde{\mathbf{u}} + \hat{\mathbf{u}} \quad \text{and} \quad q = \bar{q} + \tilde{q} + \hat{q}. \quad (5)$$

Using the decomposition (5), the momentum balance Equation (3) becomes

$$\begin{aligned} A(\mathbf{u}; (\bar{\mathbf{u}}, \bar{p}), (\bar{\mathbf{v}}, \bar{q})) + A(\mathbf{u}; (\tilde{\mathbf{u}}, \tilde{p}), (\bar{\mathbf{v}}, \bar{q})) + A(\mathbf{u}; (\hat{\mathbf{u}}, \hat{p}), (\bar{\mathbf{v}}, \bar{q})) - b(\mathbf{w}, \bar{\mathbf{u}} + \tilde{\mathbf{u}} + \hat{\mathbf{u}}, \bar{\mathbf{v}}) &= 0 \\ A(\mathbf{u}; (\bar{\mathbf{u}}, \bar{p}), (\tilde{\mathbf{v}}, \tilde{q})) + A(\mathbf{u}; (\tilde{\mathbf{u}}, \tilde{p}), (\tilde{\mathbf{v}}, \tilde{q})) + A(\mathbf{u}; (\hat{\mathbf{u}}, \hat{p}), (\tilde{\mathbf{v}}, \tilde{q})) - b(\mathbf{w}, \bar{\mathbf{u}} + \tilde{\mathbf{u}} + \hat{\mathbf{u}}, \tilde{\mathbf{v}}) &= 0 \quad (6) \\ A(\mathbf{u}; (\bar{\mathbf{u}}, \bar{p}), (\hat{\mathbf{v}}, \hat{q})) + A(\mathbf{u}; (\tilde{\mathbf{u}}, \tilde{p}), (\hat{\mathbf{v}}, \hat{q})) + A(\mathbf{u}; (\hat{\mathbf{u}}, \hat{p}), (\hat{\mathbf{v}}, \hat{q})) - b(\mathbf{w}, \bar{\mathbf{u}} + \tilde{\mathbf{u}} + \hat{\mathbf{u}}, \hat{\mathbf{v}}) &= 0. \end{aligned}$$

Note that the DNS solution will also contain the approximation of the last equation in (6).

### 3.2. Modeling assumptions

To avoid DNS of the variational form (6), the following assumptions are made.

- Contrary to DNS, the main purpose of VMS is to avoid explicit computation of the unresolved scales, and hence, the last equation (6) is ignored as the test functions are from the unresolved scales.
- The first equation in (6), that is, the equation with the test function from the resolved large scales, the term  $b(\mathbf{w}, \bar{\mathbf{u}} + \tilde{\mathbf{u}} + \hat{\mathbf{u}}, \bar{\mathbf{v}})$  incorporates to the mesh movement into the model. Using linearity in the second component, this term can be expanded, and further  $b(\mathbf{w}, \bar{\mathbf{u}}, \bar{\mathbf{v}})$ ,  $b(\mathbf{w}, \tilde{\mathbf{u}}, \bar{\mathbf{v}})$  can be combined with  $A(\mathbf{u}; (\bar{\mathbf{u}}, \bar{p}), (\bar{\mathbf{v}}, \bar{q}))$  and  $A(\mathbf{u}; (\tilde{\mathbf{u}}, \tilde{p}), (\bar{\mathbf{v}}, \bar{q}))$ , respectively, to obtain  $A(\mathbf{u} - \mathbf{w}; (\bar{\mathbf{u}}, \bar{p}), (\bar{\mathbf{v}}, \bar{q}))$  and  $A(\mathbf{u} - \mathbf{w}; (\tilde{\mathbf{u}}, \tilde{p}), (\bar{\mathbf{v}}, \bar{q}))$ . Similar modifications can be done to the second equation in (6).
- The resolved scales are further split into resolved large and resolved small scales, and it is assumed that the direct influence of the unresolved small scales is only on the resolved small scales and has negligible direct influence on the resolved large scales. Also, the Reynolds stress and cross stress terms

$$A(\mathbf{u} - \mathbf{w}; (\hat{\mathbf{u}}, \hat{p}), (\bar{\mathbf{v}}, \bar{q})), b(\hat{\mathbf{u}}, \bar{\mathbf{u}}, \bar{\mathbf{v}}), b(\hat{\mathbf{u}}, \tilde{\mathbf{u}}, \bar{\mathbf{v}}), b(\hat{\mathbf{u}}, \hat{\mathbf{u}}, \bar{\mathbf{v}})$$

that contain the unresolved and resolved large scales are assumed to be zero.

- The unresolved scales dissipate the turbulent kinetic energy from the small scales, and the unresolved scales are sufficiently small. Thus, the influence of unresolved scales on resolved small scales is modeled by an appropriate turbulence model, that is,

$$A(\mathbf{u} - \mathbf{w}; (\hat{\mathbf{u}}, \hat{p}), (\tilde{\mathbf{v}}, \tilde{q})) + b(\hat{\mathbf{u}}, \bar{\mathbf{u}}, \tilde{\mathbf{v}}) + b(\hat{\mathbf{u}}, \tilde{\mathbf{u}}, \tilde{\mathbf{v}}) + b(\hat{\mathbf{u}}, \hat{\mathbf{u}}, \tilde{\mathbf{v}}) \approx B(\mathbf{u} - \mathbf{w}; (\bar{\mathbf{u}}, \bar{p}), (\tilde{\mathbf{u}}, \tilde{p}), (\tilde{\mathbf{v}}, \tilde{q})).$$

This modeling is essential to incorporate the effect of the unresolved scales into the resolved scales of the flow. Imposing these assumptions, the VMS form of ALE-NSE (6) reduces to

$$\begin{aligned} A(\bar{\mathbf{u}} + \tilde{\mathbf{u}} - \mathbf{w}; (\bar{\mathbf{u}}, \bar{p}), (\bar{\mathbf{v}}, \bar{q})) + A(\bar{\mathbf{u}} + \tilde{\mathbf{u}} - \mathbf{w}; (\tilde{\mathbf{u}}, \tilde{p}), (\bar{\mathbf{v}}, \bar{q})) &= 0 \\ A(\bar{\mathbf{u}} + \tilde{\mathbf{u}} - \mathbf{w}; (\bar{\mathbf{u}}, \bar{p}), (\tilde{\mathbf{v}}, \tilde{q})) + A(\bar{\mathbf{u}} + \tilde{\mathbf{u}} - \mathbf{w}; (\tilde{\mathbf{u}}, \tilde{p}), (\tilde{\mathbf{v}}, \tilde{q})) + B(\mathbf{u} - \mathbf{w}; (\bar{\mathbf{u}}, \bar{p}), (\tilde{\mathbf{u}}, \tilde{p}), (\tilde{\mathbf{v}}, \tilde{q})) &= 0 \end{aligned}$$

Note that the turbulent model in the three-scale VMS acts only on the resolved small scales and not on the resolved large scales directly. Nevertheless, the three-scale model indirectly incorporates the effects of unresolved scales in the resolved large scales because of the inherent coupling of the resolved small scales with the resolved large scales. This is in contrast to the classical LES, where the turbulent model acts directly on all resolved scales.

### 3.3. Finite element discretization

A number of approaches have been proposed in the literature for the scale separation (e.g., [19, 29–35]). In this work, we consider the projection-based approach presented in [19], and extend it for ALE-NSE in time-dependent domain. Let  $\Omega_{t,h}$  be the triangulation of the domain  $\Omega_t$  into simplex cells,  $V_h(\Omega_{t,h}) \subset V(\Omega_t)$  and  $Q_h(\Omega_{t,h}) \subset Q(\Omega_t)$  be the conforming finite element spaces satisfying the discrete inf-sup condition

$$\inf_{q_h \in Q_h} \sup_{\mathbf{v}_h \in V_h} \frac{(q_h, \nabla \cdot \mathbf{v}_h)}{\|q_h\|_{L^2} \|\nabla \mathbf{v}_h\|_{L^2}} \geq \beta \geq 0 \quad (7)$$

for some positive constant  $\beta$ , which is independent of the mesh size  $h$ . Note that the continuous spaces  $V$  and  $Q$  contain all scales, whereas the finite dimensional spaces  $V_h$  and  $Q_h$  contain only the resolved large and small scales. Further, let

$$L = \{\mathbb{L} : \mathbb{L} \in (L^2(\Omega))^{3 \times 3}, \mathbb{L} = \mathbb{L}^T\}$$

be a space of  $3 \times 3$  symmetric tensors (deformation tensor in the context of NSE), and  $L_H \subset L$  be a coarse discrete space defined as

$$L_H = \{\mathbb{D}(\mathbf{v}) : \mathbb{D}(\mathbf{v}) \in L, \forall \mathbf{v} \in V_H \subset (H^1(\Omega_t))^3\}.$$

Here, the subscript  $H$  in  $V_H$  denotes that the polynomial order of  $V_H$  is smaller than the polynomial order of  $V_h$ . Next, define  $P_{L_H} : L \rightarrow L_H$ ,  $\mathbb{D}(\mathbf{v}) \mapsto P_{L_H} \mathbb{D}(\mathbf{v})$  with the  $L^2$ -projection

$$((\mathbb{I} - P_{L_H})\mathbb{D}(\mathbf{u}), \mathbb{L}_H) = 0, \quad \forall \mathbb{D}(\mathbf{u}) \in L, \mathbb{L}_H \in L_H.$$

Using these discrete finite dimensional spaces and the projection operator, the semi-discrete (continuous in time) VMS form of the ALE-NSE reads:

For given  $\mathbf{w}$ ,  $\mathbf{u}_0$  and  $\Omega_{0,h}$ , find  $\mathbf{u}_h \in L^2(0, T; V_h)$ ,  $p_h \in L^2(0, T, Q_h)$ , and  $\mathbb{G}_H = P_{L_H} \mathbb{D}(\mathbf{u}_h) \in L^2(0, T, L_H)$  such that

$$\begin{aligned} A(\mathbf{u}_h; (\mathbf{u}_h, p_h); (\mathbf{v}_h, q_h)) + \nu_T (\mathbb{D}(\mathbf{u}_h) - \mathbb{G}_H, \mathbb{D}(\mathbf{v}_h)) - b(\mathbf{w}, \mathbf{u}_h, \mathbf{v}_h) &= 0 \\ (q_h, \nabla \cdot \mathbf{u}_h) &= 0 \\ (\mathbb{D}(\mathbf{u}_h) - \mathbb{G}_H, \mathbb{L}_H) &= 0 \end{aligned} \quad (8)$$

for all  $(\mathbf{v}_h, q_h) \in V_h \times Q_h$  and  $\mathbb{L}_H \in L_H$ . Here,  $\nu_T \geq 0$  is a turbulent viscosity model and  $\mathbf{w}$  is the mesh velocity.

#### Remark 1

The space  $L_H$  and the turbulence model  $\nu_T$  need to be chosen suitably, and these two choices characterize the turbulence model of the VMS scheme. A popular choice for the turbulent viscosity model is the Smagorinsky eddy viscosity model. Further, the first equation in (8) essentially means that the turbulent viscosity is added to all resolved scales and then the turbulent viscosity is subtracted from the large scales, thereby limiting the effect of turbulent model only on the resolved small scales. This is the main idea of VMS.

#### Remark 2

The third equation in (8) represents an  $L^2$ -projection of  $\mathbb{D}(\mathbf{u}_h)$  from  $L$  onto  $L_H$ . In a three-way partition of the flow field, the space  $L$  can be interpreted as a space representing the deformation tensor of the resolved scales, containing both the large and the small scales, whereas the space  $L_H$  represents the deformation tensor of the resolved large scales.

## 4. TEMPORAL DISCRETIZATION AND LINEARIZATION

### 4.1. Temporal discretization

Let  $0 = t^0 < t^1 < \dots < t^N = T$  be a uniform decomposition of the considered time interval  $[0, T]$ , and  $\delta_t = t^n - t^{n-1}$ ,  $1 \leq n \leq N$ , be the time-step size. For brevity, we denote the discrete functions  $\mathbf{u}_h(t^n, x)$  and  $p_h(t^n, x)$  as  $\mathbf{u}_h^n$  and  $p_h^n$ , respectively. Further, the triangulated domain  $\Omega_{t,h}$  and the moving solid structure boundary  $\Gamma_S$  at time  $t^n$  are denoted by  $\Omega_{t^n,h}$  and  $\Gamma_S^n$ , respectively. Moreover,  $V_h^n$ ,  $Q_h^n$ , and  $L_H^n$  denote the discrete spaces defined on  $\Omega_{t^n,h}$ . The mesh velocity is often part of

the solution and needs to be approximated, for instance in fluid–structure interaction problems. In order to solve the VMS form of the ALE-NSE (9), the mesh velocity is needed. The discrete ALE mapping in  $[t^{n-1}, t^n]$  is defined by

$$\mathcal{A}_{h,t}(Y) = \frac{t - t^{n-1}}{\delta_t} \mathcal{A}_{h,t^n}(Y) + \frac{t^n - t}{\delta_t} \mathcal{A}_{h,t^{n-1}}(Y),$$

for  $t \in [t^{n-1}, t^n]$  and the discrete mesh velocity is then defined by

$$\hat{\mathbf{w}}_h^n(Y) = \frac{\mathcal{A}_{h,t^n}(Y) - \mathcal{A}_{h,t^{n-1}}(Y)}{\delta_t}$$

as a piecewise constant function in time. Further, the functions in the Eulerian frame are defined by applying the discrete ALE mapping. For example, the unknown discrete functions in the Eulerian frame are defined by

$$\mathbf{w}_h^n = \hat{\mathbf{w}}_h^n \circ \mathcal{A}_{h,t^n}^{-1}(x), \quad p_h^n = \hat{p}_h^n \circ \mathcal{A}_{h,t^n}^{-1}(x), \quad \mathbb{G}_H^n = \hat{\mathbb{G}}_H^n \circ \mathcal{A}_{h,t^n}^{-1}(x), \quad \mathbf{w}_h^n = \hat{\mathbf{w}}_h^n \circ \mathcal{A}_{h,t^n}^{-1}(x).$$

Applying now the general  $\theta$ -scheme, the discrete form of (8) in the interval  $(t^{n-1}, t^n)$  reads:

For given  $\mathbf{w}_h^n, \mathbf{u}^{n-1}$  and  $\Omega_{t^{n-1},h}$  with  $\mathbf{u}^0 = \mathbf{u}_0$ , find  $\mathbf{u}_h^n \in V_h^n, p_h \in Q_h^n$  and  $\mathbb{G}_H^n \in L_H^n$  such that

$$\begin{aligned} \left( \frac{\mathbf{u}_h^n - \mathbf{u}_h^{n-1}}{\delta_t}, \mathbf{v} \right) + \frac{2\theta}{\text{Re}} (\mathbb{D}(\mathbf{u}_h^n), \mathbb{D}(\mathbf{v})) + \theta b(\mathbf{u}_h^n - \mathbf{w}_h^n, \mathbf{u}_h^n, \mathbf{v}) + \theta \nu_T (\mathbb{D}(\mathbf{u}_h^n) - \mathbb{G}_H^n, \mathbb{D}(\mathbf{v}_h)) \\ - (p_h^n, \nabla \cdot \mathbf{v}) = (1 - \theta) \left[ \frac{2}{\text{Re}} (\mathbb{D}(\mathbf{u}_h^{n-1}), \mathbb{D}(\mathbf{v})) + b(\mathbf{u}_h^{n-1} - \mathbf{w}_h^{n-1}, \mathbf{u}_h^{n-1}, \mathbf{v}) \right] \\ + (1 - \theta) \nu_T (\mathbb{D}(\mathbf{u}_h^{n-1}) - \mathbb{G}_H^{n-1}, \mathbb{D}(\mathbf{v}_h)) \\ (q_h, \nabla \cdot \mathbf{u}_h^n) = 0 \\ \theta (\mathbb{D}(\mathbf{u}_h^n) - \mathbb{G}_H, \mathbb{L}_H) = (1 - \theta) (\mathbb{D}(\mathbf{u}_h^{n-1}) - \mathbb{G}_H^{n-1}, \mathbb{L}_H) \end{aligned} \tag{9}$$

for all  $(\mathbf{v}_h, q_h) \in V_h^n \times Q_h^n$  and  $\mathbb{L}_H \in L_H^n$ . Note that a nonlinear solver has to be used to solve the system (9) because of the nonlinear convective term. Alternatively, an iteration of fixed point type can also be applied to handle the nonlinear convective term.

*Remark 3*

The main difference between the NSE defined in a fixed and in a time-dependent domain is the mesh velocity term,  $b(\mathbf{u}_h^n - \mathbf{w}_h^n, \mathbf{u}_h^n, \mathbf{v})$  in the first equation in (9). Suppose the displacement of the domain is prescribed, then the mesh velocity is known a priori and it is enough to compute  $\mathbf{w}_h^n$  only once at each time step. However, the computational domain is part of the solution in fluid–structure interaction and multiphase flows, and therefore  $\mathbf{w}_h^n$  is unknown, and eventually it has to be calculated at each nonlinear iteration step.

*4.2. Computation of the mesh velocity and mesh movement*

In this section, we describe the calculation of the mesh velocity and the used mesh movement technique. Let  $\gamma_S^n$  be the displacement of the solid structure boundary at  $t^n$  from  $\Gamma_S^{n-1}$  to  $\Gamma_S^n$ . Then, the displacement  $\Psi^n$  of the inner points (mesh vertices) in  $\Omega_{t^{n-1},h}$  is obtained by solving the linear elasticity equation

$$\begin{aligned} \nabla \cdot \mathbb{T}(\Psi^n) &= 0 && \text{in } \Omega_{t^{n-1},h} \\ \Psi^n &= \gamma_S^n && \text{on } \Gamma_S^{n-1} \\ \Psi^n &= 0 && \text{on } \partial\Omega_{t^{n-1},h} \setminus \Gamma_S^{n-1} \end{aligned} \tag{10}$$

where  $\mathbb{T}(\phi) = \lambda_1(\nabla \cdot \phi)\mathbb{I} + 2\lambda_2\mathbb{D}(\phi)$ . In computations, the Lamé constants  $\lambda_1$  and  $\lambda_2$  are chosen as one. Further, continuous piecewise linear polynomials are used to approximate each component of the displacement vector. Moreover, the solution of Equation (10) is approximated by using mapped finite elements, that is, all integrals in the discrete form of Equation (10) are transformed over the reference cell by affine mapping, and evaluated on the reference cell. Further, the determinant of the Jacobian is dropped in the transformed reference integral that contains the stiffness term in order to preserve the predefined mesh adaptivity during the deformation. Because the determinant of the Jacobian will be small for smaller cells, ignoring it in the reference integral of the stiffness term induces more stiffness to the smaller cells. Consequently, the deformation of the smaller cells will be less in comparison to the larger cells, and it helps to maintain the predefined mesh adaptivity.

Once the displacement vector is calculated for all points in  $\Omega_{t^{n-1},h}$ , then the domain  $\Omega_{t^n,h}$  is obtained by adding the displacement to the coordinates of vertices in  $\Omega_{t^{n-1},h}$ . Further, the discrete mesh velocity at  $\mathbf{w}_h^n$  is obtained by

$$\mathbf{w}_h^n = \frac{\Psi^n}{\delta_t},$$

which is piecewise constant in time. The mesh movement with the linear elastic approach avoids the remeshing in the considered examples.

## 5. IMPLEMENTATIONS OF ARBITRARY LAGRANGIAN–EULERIAN VARIATIONAL MULTISCALE METHOD IN TIME-DEPENDENT DOMAINS

In this section, we present the matrices that arise from the discrete form (9), and then the necessary modifications to obtain the standard system matrix format of the NSE. In general, the small and large-scale finite element spaces can be defined in two-ways, either using the same order polynomials for both  $L_H$  and  $V_h$  but defining it on a coarse and a fine mesh, respectively, or defining  $L_H$  and  $V_h$  on the same mesh with lower and higher order polynomial functions, respectively. We prefer the second choice.

Let  $\phi_i^I, \phi_i^{II}, \phi_i^{III}$ ,  $i = 1, 2, \dots, N_U$  be the basis functions of the velocity components ( $u_{1,h}, u_{2,h}, u_{3,h}$ ), respectively. Here,  $N_U$  is the number of degrees of freedom (DOF) of each velocity component. Then, the finite element space for the resolved small scale is defined by

$$V_h = \text{span} \left\{ \begin{pmatrix} \phi_i^I \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \phi_i^{II} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \phi_i^{III} \end{pmatrix} : \text{for } i = 1, 2, \dots, N_U \right\}.$$

The superscripts  $I, II, III$  are used to indicate the corresponding basis functions of the velocity components. However, the same basis functions are used for each component of the velocity. Let  $\psi_k, k = 1, 2, \dots, N_L$  be the basis functions of each component of the tensor. Here,  $N_L$  is the number of DOF of each tensor component. Then, the components of the tensor are defined by

$$\mathbb{G}_H = \begin{bmatrix} g_{11}^h & g_{12}^h & g_{13}^h \\ g_{12}^h & g_{22}^h & g_{23}^h \\ g_{13}^h & g_{23}^h & g_{33}^h \end{bmatrix},$$

Furthermore, the space of symmetric tensors,  $L_H$  is defined by

$$L_H = \text{span} \left\{ \begin{pmatrix} \psi_k & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \frac{1}{2} \begin{pmatrix} 0 & \psi_k & 0 \\ \psi_k & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \frac{1}{2} \begin{pmatrix} 0 & 0 & \psi_k \\ 0 & 0 & 0 \\ \psi_k & 0 & 0 \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 0 & 0 & 0 \\ 0 & \psi_k & 0 \\ 0 & 0 & 0 \end{pmatrix}, \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \psi_k \\ 0 & \psi_k & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \psi_k \end{pmatrix} \right\}$$

for  $k = 1, 2, \dots, N_L$ . Using the definition of finite dimensional spaces, the discrete form of ALE-NSE (9) with  $\theta = 1$  results in the system of algebraic equations

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & B_1^T & \tilde{G}_{11} & \tilde{G}_{12} & \tilde{G}_{13} & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & B_2^T & 0 & \tilde{G}_{22} & 0 & \tilde{G}_{24} & \tilde{G}_{25} & 0 \\ A_{31} & A_{32} & A_{33} & B_3^T & 0 & 0 & \tilde{G}_{33} & 0 & \tilde{G}_{35} & \tilde{G}_{36} \\ B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G_{11} & 0 & 0 & 0 & M & 0 & 0 & 0 & 0 & 0 \\ G_{21} & G_{22} & 0 & 0 & 0 & M/2 & 0 & 0 & 0 & 0 \\ G_{31} & 0 & G_{33} & 0 & 0 & 0 & M/2 & 0 & 0 & 0 \\ 0 & G_{42} & 0 & 0 & 0 & 0 & 0 & M & 0 & 0 \\ 0 & G_{52} & G_{53} & 0 & 0 & 0 & 0 & 0 & M/2 & 0 \\ 0 & 0 & G_{63} & 0 & 0 & 0 & 0 & 0 & 0 & M \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ P \\ g_{11}^h \\ g_{12}^h \\ g_{13}^h \\ g_{22}^h \\ g_{23}^h \\ g_{33}^h \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (11)$$

Here, the matrices  $A_{rs}, B_r^T$  and  $B_r, 1 \leq r, s \leq 3$  are the standard matrices in Navier–Stokes problem. Note that  $A_{rr}$  contains the mesh velocity. Further,

$$\begin{aligned} (\tilde{G}_{11})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 2\phi_{i,x}^I & \phi_{i,y}^I & \phi_{i,z}^I \\ \phi_{i,y}^I & 0 & 0 \\ \phi_{i,z}^I & 0 & 0 \end{pmatrix} : \begin{pmatrix} 2\psi_j & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{12})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 2\phi_{i,x}^I & \phi_{i,y}^I & \phi_{i,z}^I \\ \phi_{i,y}^I & 0 & 0 \\ \phi_{i,z}^I & 0 & 0 \end{pmatrix} : \begin{pmatrix} 0 & \psi_j & 0 \\ \psi_j & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{13})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 2\phi_{i,x}^I & \phi_{i,y}^I & \phi_{i,z}^I \\ \phi_{i,y}^I & 0 & 0 \\ \phi_{i,z}^I & 0 & 0 \end{pmatrix} : \begin{pmatrix} 0 & 0 & \psi_j \\ 0 & 0 & 0 \\ \psi_j & 0 & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{22})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 0 & \phi_{i,x}^{II} & 0 \\ \phi_{i,x}^{II} & 2\phi_{i,y}^{II} & \phi_{i,z}^{II} \\ 0 & \phi_{i,z}^{II} & 0 \end{pmatrix} : \begin{pmatrix} 0 & \psi_j & 0 \\ \psi_j & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{24})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 0 & \phi_{i,x}^{II} & 0 \\ \phi_{i,x}^{II} & 2\phi_{i,y}^{II} & \phi_{i,z}^{II} \\ 0 & \phi_{i,z}^{II} & 0 \end{pmatrix} : \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2\psi_j & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{25})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 0 & \phi_{i,x}^{II} & 0 \\ \phi_{i,x}^{II} & 2\phi_{i,y}^{II} & \phi_{i,z}^{II} \\ 0 & \phi_{i,z}^{II} & 0 \end{pmatrix} : \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \psi_j \\ 0 & \psi_j & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{33})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 0 & 0 & \phi_{i,x}^{III} \\ 0 & 0 & \phi_{i,y}^{III} \\ \phi_{i,x}^{III} & \phi_{i,y}^{III} & 2\phi_{i,z}^{III} \end{pmatrix} : \begin{pmatrix} 0 & 0 & \psi_j \\ 0 & 0 & 0 \\ \psi_j & 0 & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{35})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 0 & 0 & \phi_{i,x}^{III} \\ 0 & 0 & \phi_{i,y}^{III} \\ \phi_{i,x}^{III} & \phi_{i,y}^{III} & 2\phi_{i,z}^{III} \end{pmatrix} : \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \psi_j \\ 0 & \psi_j & 0 \end{pmatrix} \right\} \\ (\tilde{G}_{36})_{ij} &= \frac{\nu_T}{4} \left\{ \begin{pmatrix} 0 & 0 & \phi_{i,x}^{III} \\ 0 & 0 & \phi_{i,y}^{III} \\ \phi_{i,x}^{III} & \phi_{i,y}^{III} & 2\phi_{i,z}^{III} \end{pmatrix} : \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2\psi_j \end{pmatrix} \right\} \\ (G_{11})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 2\psi_i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} : \begin{pmatrix} 2\phi_{i,x}^I & \phi_{i,y}^I & \phi_{i,z}^I \\ \phi_{i,y}^I & 0 & 0 \\ \phi_{i,z}^I & 0 & 0 \end{pmatrix} \right\} \end{aligned}$$

$$\begin{aligned}
 (G_{21})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & \psi_i & 0 \\ \psi_i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} : \begin{pmatrix} 2\phi_{i,x}^I & \phi_{i,y}^I & \phi_{i,z}^I \\ \phi_{i,y}^I & 0 & 0 \\ \phi_{i,z}^I & 0 & 0 \end{pmatrix} \right\} \\
 (G_{22})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & \psi_i & 0 \\ \psi_i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} : \begin{pmatrix} 0 & \phi_{i,x}^{II} & 0 \\ \phi_{i,x}^{II} & 2\phi_{i,y}^{II} & \phi_{i,z}^{II} \\ 0 & \phi_{i,z}^{II} & 0 \end{pmatrix} \right\} \\
 (G_{31})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & 0 & \psi_i \\ 0 & 0 & 0 \\ \psi_i & 0 & 0 \end{pmatrix} : \begin{pmatrix} 2\phi_{i,x}^I & \phi_{i,y}^I & \phi_{i,z}^I \\ \phi_{i,y}^I & 0 & 0 \\ \phi_{i,z}^I & 0 & 0 \end{pmatrix} \right\} \\
 (G_{33})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & 0 & \psi_i \\ 0 & 0 & 0 \\ \psi_i & 0 & 0 \end{pmatrix} : \begin{pmatrix} 0 & 0 & \phi_{i,x}^{III} \\ 0 & 0 & \phi_{i,y}^{III} \\ \phi_{i,x}^{III} & \phi_{i,y}^{III} & 2\phi_{i,z}^{III} \end{pmatrix} \right\} \\
 (G_{42})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2\psi_i & 0 \\ 0 & 0 & 0 \end{pmatrix} : \begin{pmatrix} 0 & \phi_{i,x}^{II} & 0 \\ \phi_{i,x}^{II} & 2\phi_{i,y}^{II} & \phi_{i,z}^{II} \\ 0 & \phi_{i,z}^{II} & 0 \end{pmatrix} \right\} \\
 (G_{52})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \psi_i \\ 0 & \psi_i & 0 \end{pmatrix} : \begin{pmatrix} 0 & \phi_{i,x}^{II} & 0 \\ \phi_{i,x}^{II} & 2\phi_{i,y}^{II} & \phi_{i,z}^{II} \\ 0 & \phi_{i,z}^{II} & 0 \end{pmatrix} \right\} \\
 (G_{53})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2\psi_i & 0 \\ 0 & 0 & 0 \end{pmatrix} : \begin{pmatrix} 0 & 0 & \phi_{i,x}^{III} \\ 0 & 0 & \phi_{i,y}^{III} \\ \phi_{i,x}^{III} & \phi_{i,y}^{III} & 2\phi_{i,z}^{III} \end{pmatrix} \right\} \\
 (G_{63})_{ij} &= \frac{1}{4} \left\{ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2\psi_i \end{pmatrix} : \begin{pmatrix} 0 & 0 & \phi_{i,x}^{III} \\ 0 & 0 & \phi_{i,y}^{III} \\ \phi_{i,x}^{III} & \phi_{i,y}^{III} & 2\phi_{i,z}^{III} \end{pmatrix} \right\}.
 \end{aligned}$$

In the preceding expression, the subscripts  $x$ ,  $y$ ,  $z$  in the basis functions denote the derivative of the basis functions with respect to  $x$ ,  $y$ ,  $z$ . Further, the matrix  $M$  is the mass matrix of the large scale space  $L_H$  and is given by  $M_{i,j} = (\psi_i, \psi_j)$ . Here,  $( : )$  represents the tensor product in the respective spaces. The preceding system can be solved for  $g_{11}$ ,  $g_{12}$ ,  $g_{13}$ ,  $g_{22}$ ,  $g_{23}$ ,  $g_{33}$ , and then substituting it in (11) results in a saddle point system of the form

$$\begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} & \hat{A}_{13} & B_1^T \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{23} & B_2^T \\ \hat{A}_{31} & \hat{A}_{32} & \hat{A}_{33} & B_3^T \\ B_1 & B_2 & B_3 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ 0 \end{bmatrix}.$$

For the explicit form of the modified block matrices, we refer to [19]. There are two main challenges associated with the computation of these modified matrices: first, the inverse of the large-scale space mass matrix has to be simple to compute, and next, the sparsity of  $\hat{A}_{rs}$ ,  $1 \leq r, s \leq 3$  should be similar to that of  $A_{rs}$ . The mass matrix becomes a diagonal matrix when orthogonal basis functions are used for the large-scale space, and hence the inverse can be calculated efficiently. Further, the choice of discontinuous basis functions for the large-scale space leads  $\hat{A}_{rs}$ ,  $1 \leq r, s \leq 3$ , to the same sparsity as of  $A_{rs}$ .

The preceding saddle point system is solved by an iteration of fixed point type because of the implicit treatment of the nonlinear convective term in the block matrices  $\hat{A}_{11}$ ,  $\hat{A}_{22}$ , and  $\hat{A}_{33}$ . Note that the convective term also contains the mesh velocity. Contrary to the NSE in stationary domains, all matrices in the preceding system change in time. Hence, in addition to the assembling of  $\hat{A}_{11}$ ,  $\hat{A}_{22}$ , and  $\hat{A}_{33}$  matrices at each fixed point iteration, all other block matrices in the system need to be assembled at every time step.

### 6. NUMERICAL EXAMPLES

In this section, we present an array of numerical simulations for flow around a stationary beam, flow around an oscillating beam, and flow around a plunging aerofoil. The first two test examples are two-dimensional, whereas the third example is three-dimensional. In all computations, the Smagorinsky model,  $\nu_T = c_t \delta^2 \|\mathbb{D}(u_h)\|_F$  with  $c_t = 0.01$  and  $\delta = 2h$  is employed. Here,  $h$  is the mesh size and  $\|\cdot\|_F$  denotes the Frobenius norm.

The two-dimensional computational domain is triangulated into triangles using the mesh generator ‘Triangle’ [36, 37], whereas the mesh generator ‘Gmesh’ [38, 39] is used to tetrahedralize the three-dimensional domain. On the triangulated mesh, an inf-sup stable finite element pair,  $P_2^{bubble}/P_1^{disc}$ , that is, continuous piecewise quadratic polynomials enriched with a cell bubble (cubic polynomial in 2D and quadratic polynomial in 3D) and four face bubbles (cubic polynomials) in 3D for the velocity, and discontinuous piecewise linear polynomial for the pressure, is used for the resolved small scales. Moreover, a piecewise constant polynomial,  $P_0$ , is used as a projection space for the resolved large scales. Further, an unconditionally stable second-order Crank-Nicolson method is used for the temporal discretization, by choosing  $\theta = 0.5$  in (9). In each time step, an iteration of fixed point type is applied to handle the non-linear convective term in the NSE. Note that the mesh velocity, which is part of the convective term, needs to be calculated in the fixed point iteration step, if the mesh velocity is unknown *a priori*. Finally, the resulting system matrix is solved using the direct solvers UMFPACK [40] in 2D, whereas MUMPS solver [41, 42] is used for 3D system. All computations are performed using our in-house finite element package, ParMooN.

#### 6.1. Flow around a stationary beam attached at a square base

We consider a two-dimensional channel that contains a rectangular beam attached to a square base. The channel is rectangular in shape of dimension 46 by 20 units, whereas the beam is of dimension 5 by 0.2 units. The base to which the beam is attached, is a unit square centered at the origin. An outline of the fluid domain that contains the structure is shown in Figure 2. We impose a free-stream boundary condition  $\mathbf{u} = (u_1, u_2)^T = (1, 0)^T$  on the top and bottom walls of the channel and a unit inlet velocity at the inflow boundary, whereas a zero Neumann condition is imposed on the outflow boundary. Further, the no-slip boundary condition is imposed on the square base and on the surface of the beam. Moreover, a zero initial condition,  $\mathbf{u}_0 = 0$ , is used. Computations are carried out with 59,860 DOF for each  $u_1$  and  $u_2$ , respectively, and 15,082 DOF for the pressure. Two sets of computations are performed with time step  $\delta_t$  as 0.025 and 0.05 for the considered Reynolds number,  $Re=100$ . Typically at this range of Reynolds number, flow around the square base without a beam attached to it would produce Karman vortex-like structures in the wake region.

Figure 3 shows the streamwise velocity profile of the fluid around a unit square obstacle and of a beam attached to a unit square base at time  $t = 35$ . The development of asymmetric flow can be seen in Figure 3(a), and it could further lead to Karman vortex-like structures at a later stage. However, it can be seen in Figure 3(b) that the presence of the beam has suppressed the formation of asymmetry flow.

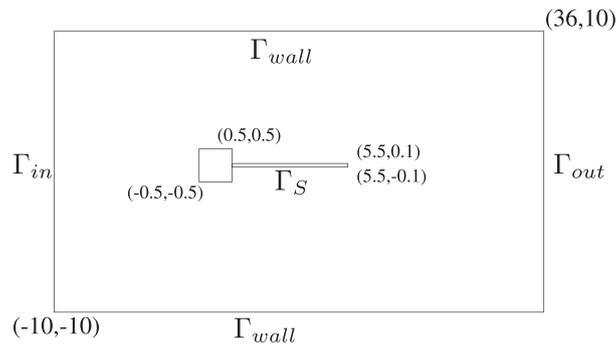


Figure 2. Schematic view of a beam attached to a square base in the fluid domain.

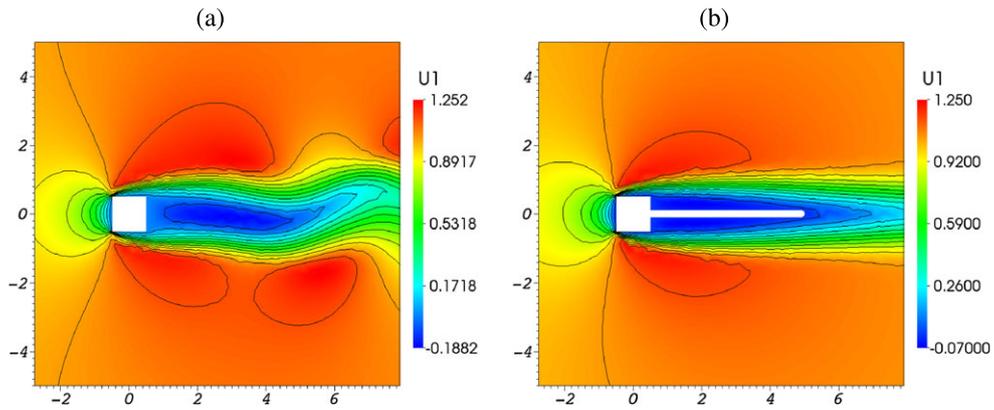


Figure 3. Streamwise velocity profile of the fluid flow around (a) unit square obstacle (b) stationary beam attached to a unit square base at  $t = 35$ . The used time-step  $\delta_t = 0.025$ .

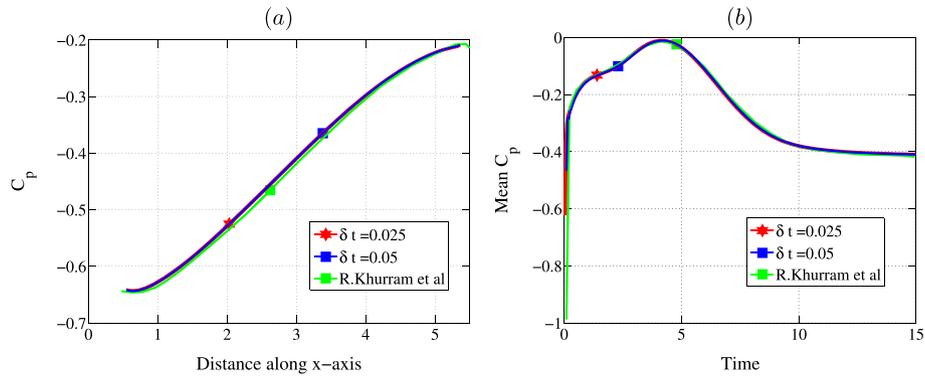


Figure 4. (a): The variation of  $C_p$  along the entire upper surface of the stationary beam at time  $t = 15$ . Time-step used  $\delta_t = 0.025$ . (b): The mean pressure coefficient,  $\overline{C_p}(t)$  over a part of the upper surface of the beam in a flow over a stationary beam.

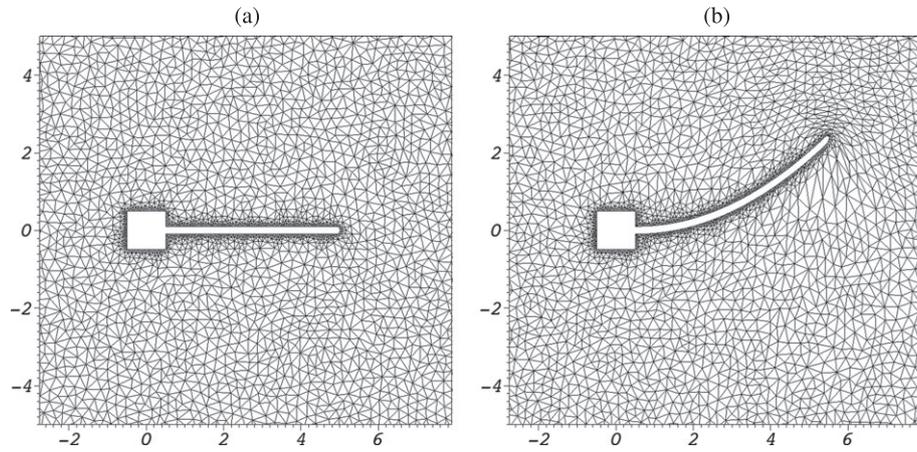


Figure 5. A close-up view of the mesh used in the oscillating beam example. (a) Beam at the initial position, and (b) beam at its maximum amplitude.

Next, to quantify the pressure variation around the beam, the mean pressure coefficient  $\overline{C_p}(t)$  is computed by

$$\text{Mean } C_p = \overline{C_p}(t) = \frac{1}{|\Gamma_C|} \int_{\Gamma_C} C_p(t, x) ds,$$

where the pressure coefficient  $C_p(t, x)$  is defined by

$$C_p(t, x) = \frac{p(t, x) - p_\alpha(t)}{(1/2)(v_\alpha(t))^2}.$$

Here,  $p_\alpha$  and  $v_\alpha$  are the free-stream pressure and velocity that are chosen to be the pressure and the streamwise velocity at the point  $(-10, 0)$ . The integral in the definition of  $\overline{C_p}(t)$  is evaluated using trapezoidal rule. In Figure 4(a), the distribution of pressure coefficient,  $C_p$  over the entire upper

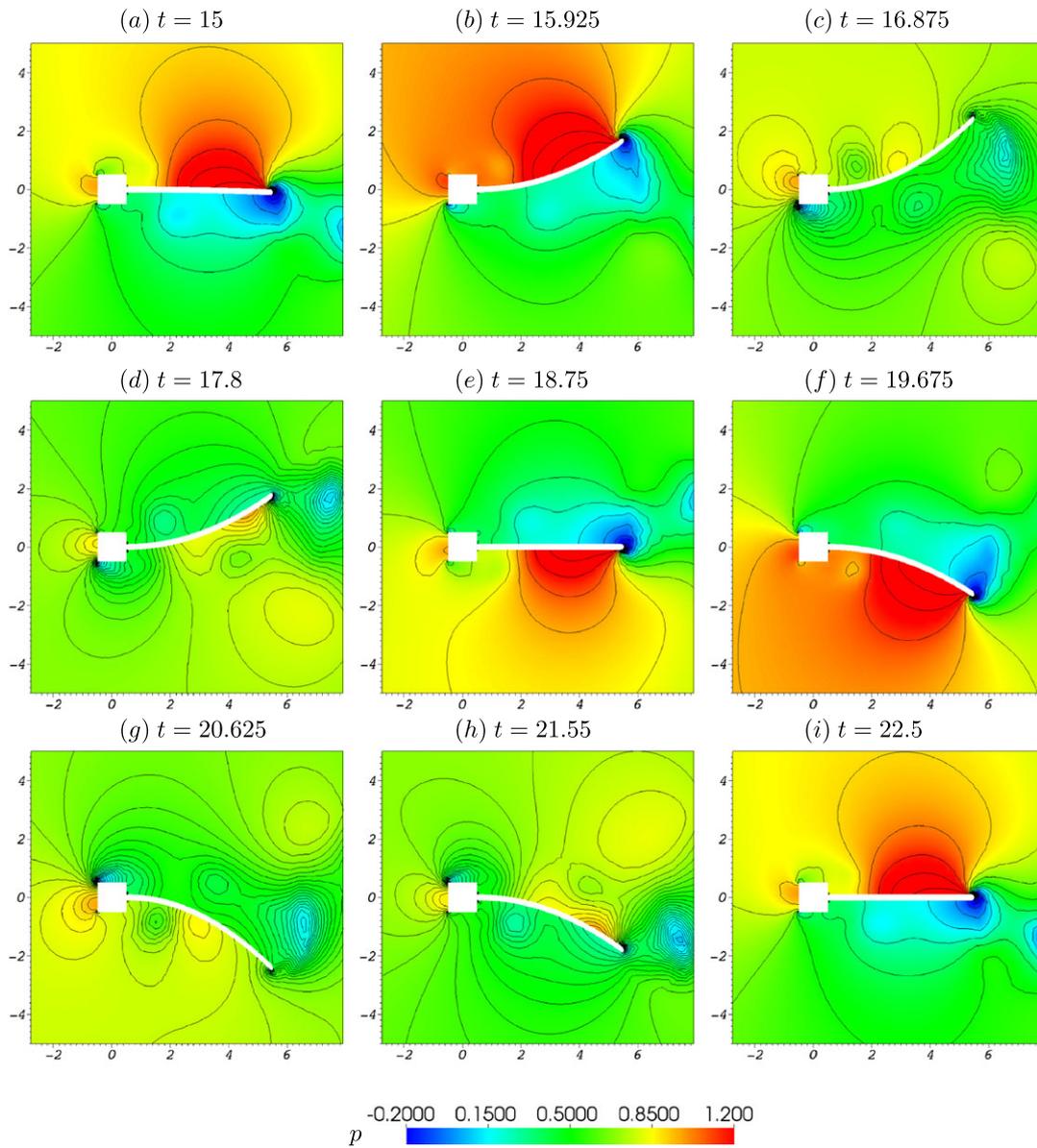


Figure 6. Pressure profile in the fluid flow around an oscillating beam at different instances of time  $t = 15, 15.925, 16.875, 17.8, 18.75, 19.675, 20.625, 21.55, 22.5$ . The used time-step  $\delta_t = 0.025$ .

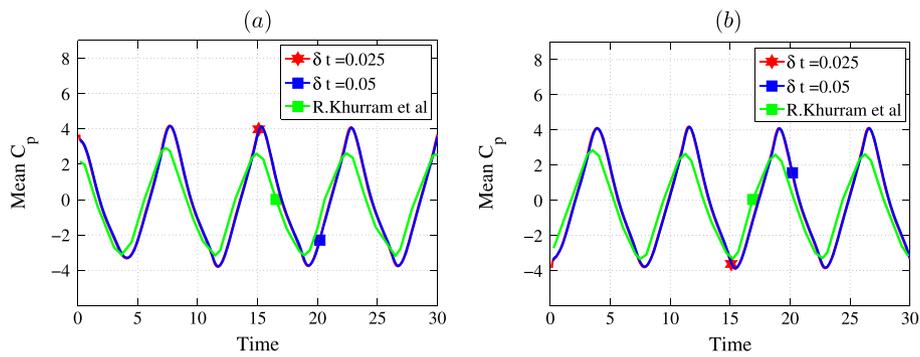


Figure 7. Mean pressure ( $\overline{C_p}$ ) profiles obtained for a section of the upper surface, (a):  $\Gamma_C = [(2.97, 3.06) \times 0.1]$ , and the lower surface, (b):  $\Gamma_C = [(2.97, 3.06) \times -0.1]$ , of the oscillating beam.

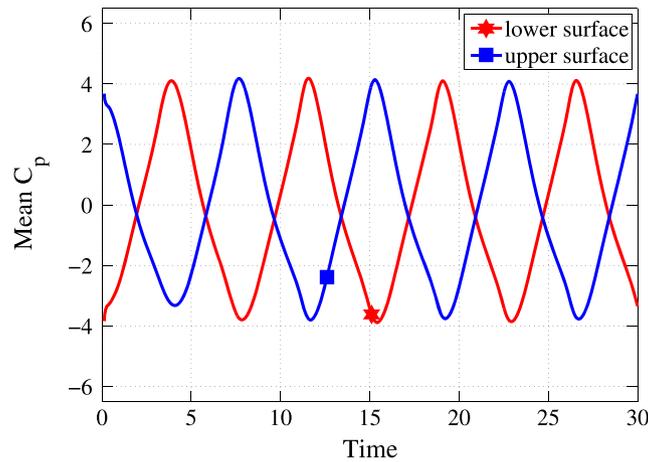


Figure 8. Superimposed mean pressure ( $\overline{C_p}$ ) profiles obtained for the upper and the lower surface of the oscillating beam. Time-step used  $\delta_t = 0.025$

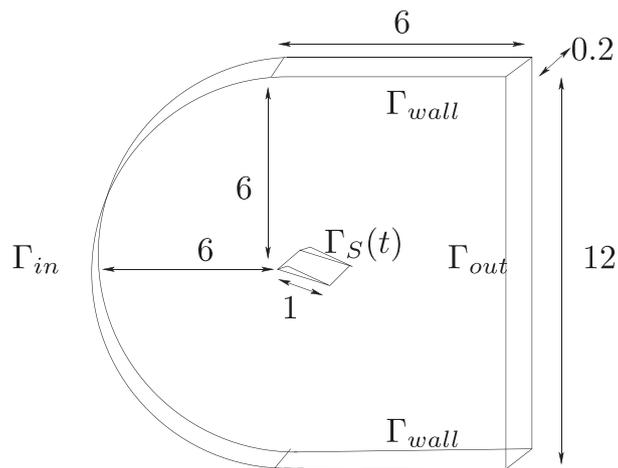


Figure 9. Computational domain for a fluid flow around an aerofoil, with characteristic length 1.

surface of the beam at  $t = 15$  is plotted. The variation of  $\overline{C_p}$  along the upper part of the beam,  $\Gamma_C = [(2.97, 3.06) \times 0.1]$ , that is, along a part of the line  $y = 0.1$ , over time is shown in Figure 4(b). After increasing initially, the mean pressure coefficient reaches a steady state. Because a similar behavior is observed for the lower surface of the beam, only the plots for the upper surface is shown. Further, the computed results,  $C_p$  and  $\overline{C_p}(t)$ , are compared with the numerical results reported in [16] and are in very good agreement.

### 6.2. Flow around an oscillating beam

We next consider a typical fluid–structure interaction problem, that is, a flow around an oscillating beam that directs the flow around it. In this two-dimensional example, the dimension of the channel and of the structure are same as in the previous example (Figure 2). Also, the initial condition, the boundary conditions, and the Reynolds number are same as in the previous example. Computations are performed with  $\delta_t = 0.025$  and  $\delta_t = 0.05$  by using 59,860 DOF for each  $u_1$  and  $u_2$ , respectively, and 15,082 DOF for the pressure. The simulations are performed till the time  $T = 50$ .

In the example, the flow is being directed by a prescribed sinusoidal movement of the beam in  $y$ -direction, and the coordinates of the beam,  $(x, y) \in \Gamma_S(t)$ , at any given time  $t$  is prescribed by

$$\begin{aligned} x(t) &= x_0 - 0.25d \tan \theta - y_0 \sin \theta; & y(t) &= y_0 + d, \\ d(t) &= A(x_0 - \alpha)^2 \sin(2\pi\omega t); & \tan \theta &= \frac{y_0}{x_0 - \alpha}. \end{aligned}$$

Here,  $(x_0, y_0) \in \Gamma_S^0$ ,  $A = 0.1$  is the amplitude,  $\alpha = 0.5$  is the abscissa of the joint of the beam with the square, and  $\omega$  is the angular velocity of the oscillation with time period for one full oscillation

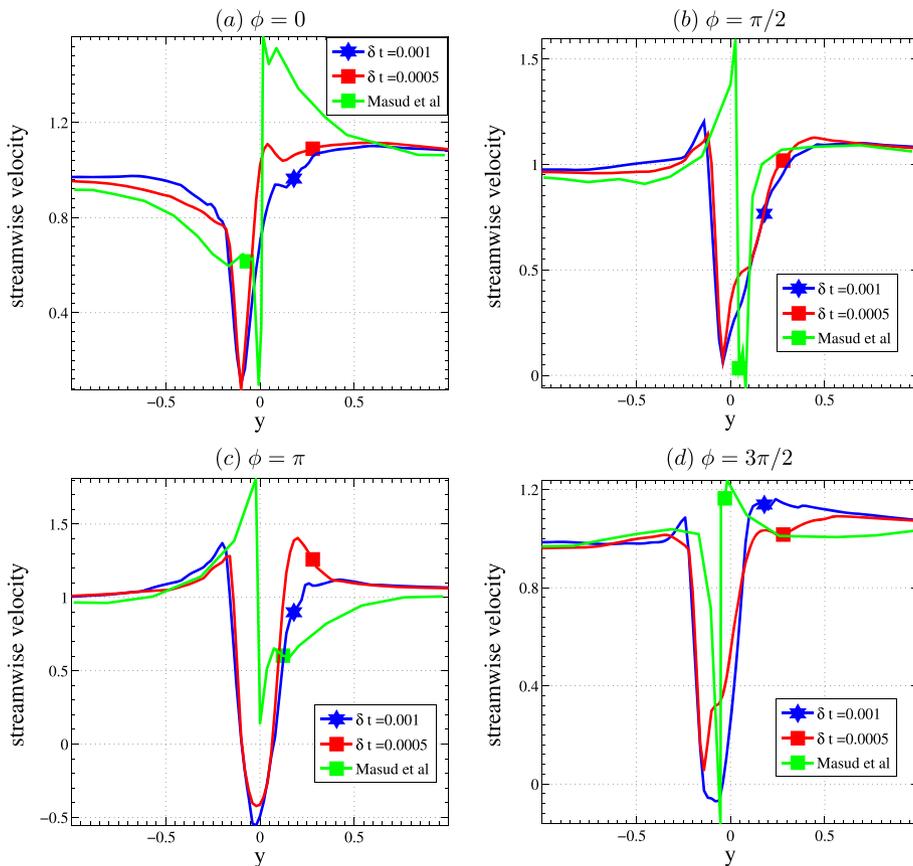


Figure 10. Mean streamwise velocity profile obtained along the line  $x = 1$  at (a)  $\phi = 0$  (b)  $\phi = \pi/2$  (c)  $\phi = \pi$  (d)  $\phi = 3\pi/2$  in plunging aerofoil computations with  $Re = 40,000$ .

being 7.5 time units. Moreover, the movement of the mesh is handled using the ALE approach as described in Sections 2.1 and 4.2. Because the Jacobian is dropped in the mesh matrix assembling, the smaller triangles near the structure get much less deformation compared with the larger triangles away from it. Consequentially, the effect of the domain deformation is absorbed substantially by the larger cells compared with the smaller cells (see Section 4.2 for more details). Further, a pre-defined adaptive mesh with a high resolution near the deforming structure and coarser away from the structure is preserved during the mesh movement. Further, the tip of the beam is considered as a semi-circle to do away with the singularities that might occur due to the sharp corners. A zoomed view of the mesh with the beam at the initial state and at its highest position are shown in Figure 5. Note that no remeshing is used.

Figure 6 shows the pressure profile in the fluid at different instances,  $t = 15, 15.925, 16.875, 17.8, 18.75, 19.675, 20.625, 21.55, 22.5$ , during one complete oscillation of the beam. Figure 6(a), (b), and (c) show the upward sweep of the beam in the positive  $y$ -direction from its initial mean position. During this stage, the velocity of the beam that is maximum at the mean position gradually decreases as the beam reaches its highest position. Because the displacement of the beam is against the fluid flow, a high pressure is observed on the upper surface of the beam and the pressure is more near the tip of the beam. Nevertheless the pressure on the upper surface of the beam decreases when the beam reaches its highest position as the velocity of the beam, which pushes the beam against the fluid, is minimum at this position. Moreover, a relatively low pressure near the square base on the upper surface can also be observed, and it can be attributed to the fact that the flow is hindered in this region by the square base. Figure 6(d) and (e) show the pressure profile in the fluid during the downward sweep of the beam from its highest position back to its mean position. During this stage, the pressure on the upper surface of the beam gradually reduces while at the same time it

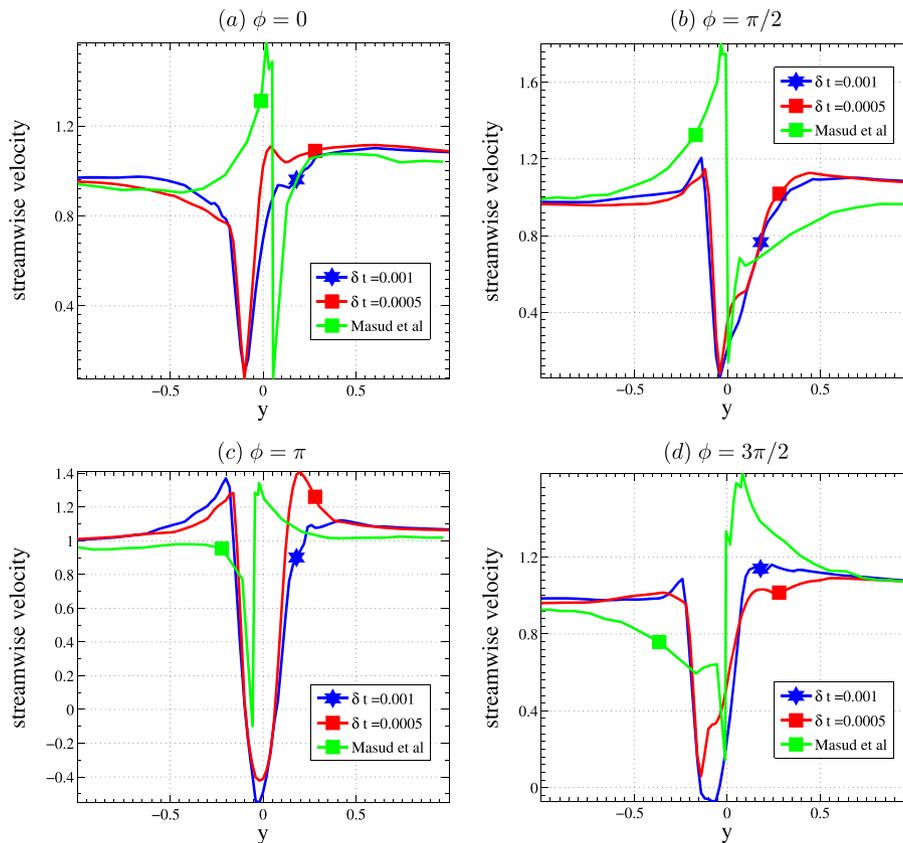


Figure 11. Mean streamwise velocity profile obtained along the line  $x = 1$  at (a)  $\phi = 0$  (b)  $\phi = \pi/2$  (c)  $\phi = \pi$  (d)  $\phi = 3\pi/2$  in plunging aerofoil computations with  $Re = 60,000$ .

increases significantly on the lower surface. Figure 6(e), (f), and (g) show the downward motion of the beam in the negative  $y$ -direction from its mean position. Because the prescribed motion of the beam being is a periodic sinusoidal one, as expected, the pressure profile in the fluid during this stage is symmetric to the previous instances when the beam was moving upward. Finally, Figure 6(h) and (i) show the upward motion of beam returning from its lowest position to its mean position. Note that the pressure profile at the initial stage (Figure 6(a)) and after one complete oscillation of the beam (Figure 6(i)) are identical, and it shows the accuracy of the numerical scheme.

Next, the plots in Figure 7(a) and (b) show the mean pressure coefficient  $\overline{C_p}(t)$  over a section on the upper surface,  $\Gamma_C = [(2.97, 3.06) \times 0.1]$ , and the lower surface,  $\Gamma_C = [(2.97, 3.06) \times -0.1]$ , of the beam, respectively. Because the oscillation of the beam is periodic and the definition of  $\overline{C_p}$  is a linear function of pressure, the oscillations of  $\overline{C_p}$  are also periodic. Both these plots are compared with the numerical results presented in [16], and are in good agreement. Next, to show the phase difference in  $\overline{C_p}(t)$  for the upper and lower surfaces of the beam, the mean pressure,  $\overline{C_p}(t)$  for both the surfaces are plotted together in Figure 8.

### 6.3. Flow around a plunging aerofoil

In this section, we present a set of 3D simulations of flow around a plunging aerofoil. A standard SD7003 aerofoil is placed within the channel whose schematic representation is given in Figure 9. The aerofoil, whose chord-line is of unit length, is placed with its tip at the origin, and is at  $4^\circ$  angle of attack. The inflow boundary,  $\Gamma_{in}$ , is semicircular with radius 6 units, with its geometric center at the origin. The outflow boundary,  $\Gamma_{out}$ , is 6 units further away from the origin. The width of the channel is 12 units and the thickness is 0.2 units. We impose the free-slip boundary condition at the top and bottom as well as at the lateral walls of the channel. On the surface of the aerofoil, the no-slip boundary condition is imposed. The Dirichlet condition with a unit inflow is imposed on the

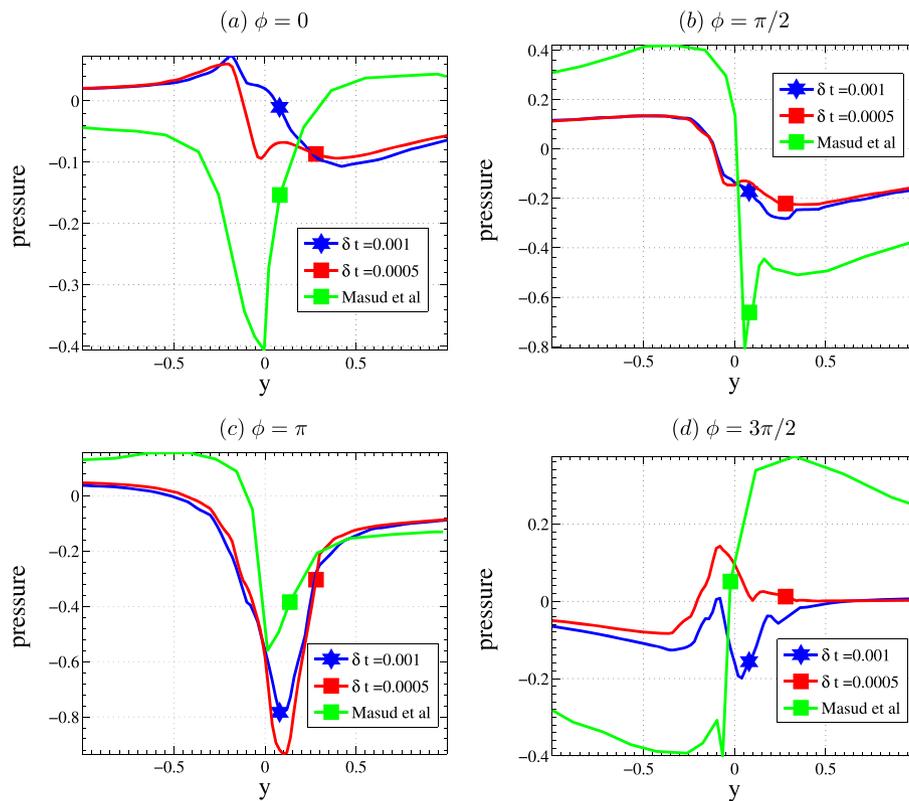


Figure 12. Mean pressure profile obtained along the line  $x = 1$  at (a)  $\phi = 0$  (b)  $\phi = \pi/2$  (c)  $\phi = \pi$  (d)  $\phi = 3\pi/2$  in plunging aerofoil computations with  $Re = 40,000$ .

inflow boundary, and the zero Neumann boundary condition is imposed on the outflow boundary. The plunging motion of the aerofoil is prescribed by a sinusoidal displacement in  $y$ -direction, the  $y$ -ordinate of the aerofoil at any given time  $t$  is prescribed by

$$y(t) = y_0 + d(t), \quad d(t) = 0.05 + 0.05 \sin(3.93(t - 6.395)) \quad \text{when } t > 6.395,$$

with time period 1.6, and  $(x_0, y_0) \in \Gamma_S^0$ . Further, a coarse mesh with  $\delta_t = 0.001$  and a finer mesh with  $\delta_t = 0.0005$  are using in this example. The choice of  $P_2^{bubble} / P_1^{disc}$  finite elements on the coarser mesh results in 30,682 DOF for each  $u_1, u_2$ , and  $u_3$ , respectively, and 23,204 DOF for the pressure, and on the finer mesh, we have 46,440 DOF for each  $u_1, u_2$ , and  $u_3$ , respectively, and 35,844 DOF for the pressure. Moreover, computations are performed with Reynolds number 40,000 and 60,000 on both meshes.

In all variants of these computations, the inlet velocity is increased linearly from 0 to 1 during the time interval  $[0, 6.395]$ , and during this same time interval the position of the aerofoil is also moved linearly in the positive  $y$ -direction from  $y = 0$  to  $y = 0.05$ . After this initial phase, the previously prescribed sinusoidal motion of the aerofoil is allowed to ensue.

In this study, four different positions of the aerofoil during its plunging motion are considered: (i) the initial stationary position as phase angle  $\phi = 0$ ; (ii) the highest displaced position along the positive  $y$ -axis as  $\phi = \pi/2$ ; (iii) the stationary position during its downward sweep as  $\phi = \pi$ ; and finally (iv) the maximum displacement along the negative  $y$ -axis as  $\phi = 3\pi/2$ .

The flow features at the wake of the plunging aerofoil at these four positions are studied by calculating the mean value of the streamwise velocity and pressure along the lines,  $x = 1$  with  $y$  ranging from  $-1$  to  $1$  along 11 equidistant points in the spanwise ( $z$ ) direction. Further, for averaging over the time, computed values are sampled over three consecutive oscillations. These measurements are

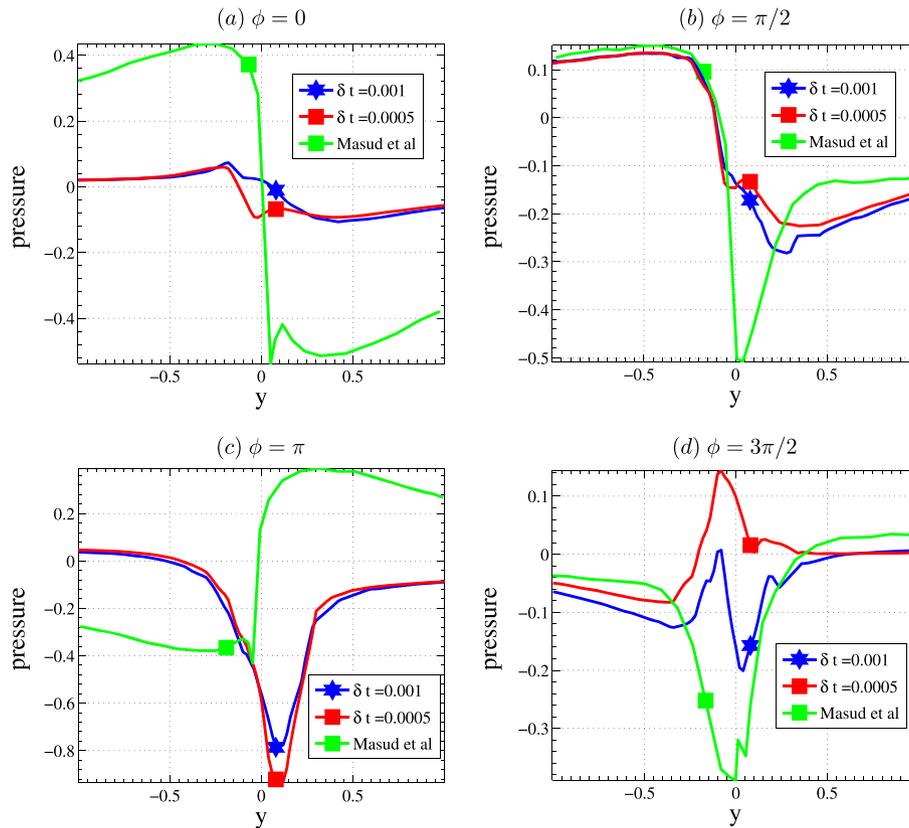


Figure 13. Mean pressure profile obtained along the line  $x = 1$  at (a)  $\phi = 0$  (b)  $\phi = \pi/2$  (c)  $\phi = \pi$  (d)  $\phi = 3\pi/2$  in plunging aerofoil computations with  $Re = 60,000$ .

taken only after the flow around the aerofoil evolve to a statistical periodicity, which is after five complete oscillations.

Figures 10 and 11 show the mean streamwise velocity profiles corresponding to  $Re = 40,000$  and  $Re = 60,000$ , respectively. The streamwise velocity profiles agree with the results of [17]. Next, the mean pressure profiles corresponding to  $Re = 40,000$  and  $Re = 60,000$  are presented in Figures 12 and 13, respectively. Although the behavior of the pressure profiles are comparable with the results in [17], there are variations in the range of some of the pressure values. Because of the sequential implementation of our code, even in the finer mesh we used 2,774 nodes and 8,961 cells, whereas 1,676,592 nodes and 1,600,225 cells were used in [17] for the coarse mesh itself. Moreover, the generated mesh is unstructured, whereas the structured mesh was used in [17]. Even though the number of nodes are very less in our computations, choosing a fine mesh near the aerofoil leads to 175,164 DOFs in total, which provides a comparable result. Nevertheless, we strongly believe that this variation in the mesh resolution is the reason for the differences in some of the obtained values. However, the proposed scheme is still able to capture the velocity and pressure profiles even with a comparatively coarser mesh.

## 7. CONCLUSION

In this paper, a finite element variational multiscale method for the incompressible NSE in time-dependent domains is presented. This work is an extension of the multiscale method proposed in [19] for turbulence flows in fixed domains. In the considered three-scale VMS scheme, the flow fields are decomposed into resolved large and small scales by a projection together with a model for the unresolved subgrid scales. In particular, a turbulent viscosity is added to all resolved scales to incorporate the effects of unresolved scales, and then the added turbulent viscosity is subtracted from the large scales, thereby confining its effect only on the resolved small scales. Moreover, the velocity space in the proposed VMS scheme does not need to be enriched with higher order polynomial bubble functions, which vanish on element boundaries, to capture the resolved small scales. Further, the second-order Crank–Nicolson method is used for the temporal discretization. The moving boundaries at the fluid–solid interfaces and the time-dependent domain are handled by the ALE approach, whereas an adaptivity preserving mesh movement is obtained by solving the linear elastic equation with mesh-dependent stiffness. Furthermore, this method can be implemented efficiently on an existing NSE solver with some assumptions on the resolved large-scale space.

Three numerical examples are considered in this study, to show the robustness and applicability of this scheme: (i) flow around a static beam attached to a square base; (ii) flow around an oscillating beam; and (iii) flow around plunging aerofoil, with different Reynolds numbers and meshes. The obtained results are in agreement with the existing results in the literature. The extension of the projection-based three-scale VMS scheme for fully coupled fluid–structure interaction problems will be pursued in the future.

## ACKNOWLEDGEMENTS

The authors would like to thank Prof. Volker John for sharing his insightful knowledge on turbulent models with us during the course of this research. Also, we would like to thank our group member Mr. Raviteja Meesala for his help and support on computations.

## REFERENCES

1. Davidson PA. *Turbulence. An Introduction for Scientists and Engineers*. Oxford University Press: Oxford, 2004.
2. Pope SB. *Turbulent Flows*. Cambridge University Press: Cambridge, 2000.
3. Fischer P, Iliescu T. Large eddy simulation of turbulent channel flows by the rational LES model. *Physics of Fluids* 2003; **15**:3036–3047.
4. Guermond JL, Oden JT, Prudhomme S. Mathematical perspectives on large eddy simulation models for turbulent flows. *Journal of Mathematical Fluid Mechanics* 2004; **6**:194–248.
5. Rogallo RS, Moin P. Numerical simulation of turbulent flows. *Annual Review of Fluid Mechanics* 1984; **16**:99–137.
6. Sagaut P. *Large Eddy Simulation for Incompressible Flows* (2nd ed). Springer: Berlin, 2002.

7. Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet to Neumann formulation, subgrid scale models, bubbles and origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**: 387–401.
8. Hughes TJR, Feijoo GR, Mazzei L, Quincy JB. The variational multiscale method—a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering* 1998; **166**:3–24.
9. Collis SS. Monitoring unresolved scales in multiscale turbulence modelling. *Physics of Fluids* 2001; **13**(6): 1800–1806.
10. Gamnitzer P, Gravemeier V, Wall WA. Advances in variational multiscale methods for turbulent flows. *Multiscale Methods in Computational Mechanics Lecture Notes in Applied and Computational Mechanics* 2011; **55**:39–52.
11. Gravemeier V. The variational multiscale methods for laminar and turbulent incompressible flows. *PhD thesis*, Institute of structural mechanics. University of Stuttgart, 2003.
12. John V. On large eddy simulation and variational multiscale methods in the numerical simulation of turbulent incompressible flows. *Applied Mathematics* 2006; **51**(4):321–353.
13. John V, Kaya S, Layton WJ. A two-level variational multiscale method for convection-dominated convection diffusion equations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:4594–4603.
14. Kaya S, Layton WJ. Subgrid scale eddy viscosity models and variational multiscale methods. *Technical Report TR-MATH 03-05*, University of Pittsburgh, Pittsburgh, 2003.
15. Bazilevs Y, Calo VM, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2007; **197**:173–201.
16. Khurram RA, Masud A. A multiscale stabilized formulation of the incompressible Navier-Stokes equations for moving boundary flows and fluid-structure interaction. *Computational Mechanics* 2006; **38**:403–416.
17. Masud A, Calderer R. Residual based turbulence models for moving boundary flows: hierarchical application of vms and three level scale separation. *International Journal for Numerical Methods in Fluids* 2013; **73**:284–305.
18. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics* 2008; **43**:3–37.
19. John V, Kaya S. A finite element variational multiscale method for the Navier-Stokes equation. *SIAM Journal on Scientific Computing* 2005; **26**(5):1485–1503.
20. Ganesan S., Tobiska L. An accurate finite element scheme with moving meshes for computing 3D-axisymmetric interface flows. *International Journal for Numerical Methods in Fluids* 2008; **57**(2):119–138.
21. Ganesan S, Tobiska L. Arbitrary Lagrangian–Eulerian finite-element method for computation of two-phase flows with soluble surfactants. *Journal of Computational Physics* 2012; **231**(9):3685–3702.
22. Osher S, Sethian JA. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
23. Pilliod JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics* 2001; **169**(2):708–759.
24. Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, Jan YJ. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics* 2001; **169**(2):708–759.
25. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu MC, Kostov N, McIntyre S. Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. *Archives of Computational Methods in Engineering* 2014; **21**: 359–398.
26. John V, Kindl A. Variants of projection based finite element variational multiscale methods for simulation of turbulent flows. *International Journal for Numerical Methods in Fluids* 2008; **56**:1321–1328.
27. John V, Roland M. Simulations of the turbulent channel flow at  $Re=180$ , with projection based variational multiscale methods. *International Journal for Numerical Methods in Fluids* 2007; **55**:407–429.
28. Hughes TJR, Mazzei L, Jensen KE. Large eddy simulation and variational multiscale methods. *Comput. Visual. Science* 2000; **3**:47–59.
29. Gravemeier V. Scale-separating operators for variational multiscale large eddy simulation of turbulent flows. *Journal of Computational Physics* 2006; **212**:400–435.
30. Gravemeier V, Wall WA, Ramm E. A three-level finite element method for the stationary incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(15-16):1323–1366.
31. Gravemeier V, Wall WA, Ramm E. Large eddy simulation of turbulent incompressible flows by a three-level finite element method. *International Journal for Numerical Methods in Fluids* 2005; **48**(10):1067–1099.
32. Holmen Jens, Hughes TJR, Oberai AA, Wells GN. Sensitivity of the scale partition for variational multiscale large eddy simulation of channel flow. *Physics of Fluids* 2004; **16**(3):824–827.
33. Hughes TJR, Oberai AA, Mazzei L. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Physics of Fluids* 2001; **13**(6):1784–1799.
34. Ramakrishnan S, Collis SS. Turbulence control simulation using the variational multiscale method. *AIAA Journal* 2004; **42**(4):745–753.
35. Ramakrishnan Ss, Collis SS. Partition selection in multiscale turbulence modeling. *Physics of Fluids* 2006; **18**(7):075105: 1–16.
36. Shewchuk JR. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. *Applied Computational Geometry Towards Geometric Engineering* 1996; **1148**:203–222.
37. Shewchuk JR. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry* 2002; **22**:21–74.

38. Geuzaine C, Remacle JF. Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering* 2009; **79**:1309–1331.
39. Remacle JF, Geuzaine C, Compère G, Marchandise E. High-quality surface remeshing using harmonic maps. *International Journal for Numerical Methods in Engineering* 2010; **83**:403–425.
40. Davis TA. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software* 2004; **30**(2):196–199.
41. Amestoy PR, Duff IS, Koster J, L'Excellent JY. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications* 2001; **23**(1):15–41.
42. Amestoy PR, Guermouche A, L'Excellent JY, Pralet S. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing* 2006; **32**(2):136–156.