Prepared for *Archives of Computational Methods in Engineering*

Kenji Takizawa · Yuri Bazilevs · Tayfun E. Tezduyar · Ming-Chen Hsu ·
Ole Øiseth · Kjell M. Mathisen · Nikolay Kostov · Spenser McIntyre

Engineering Analysis and Design with ALE-VMS and Space–Time Methods

Abstract Flow problems with moving boundaries and interfaces include fluid–structure interaction (FSI) and a number of other classes of problems, have an important place in engineering analysis and design, and offer some formidable computational challenges. Bringing solution and analysis to them motivated the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) method and also the variational multiscale version of the Arbitrary Lagrangian–Eulerian method (ALE-VMS). Since their inception, these two methods and their improved versions have been applied to a diverse set of challenging problems with a common core computational technology need. The classes of problems solved include free-surface and two-fluid flows, fluid–object and fluid–particle interaction, FSI, and flows with solid surfaces in fast, linear or rotational relative motion. Some of the most challenging FSI problems, including parachute FSI, wind-turbine FSI and arterial FSI, are being solved and analyzed with the DSD/SST and ALE-VMS methods as core technologies. Better accuracy and improved turbulence modeling were brought with the recently-introduced VMS version of the DSD/SST method, which is called DSD/SST-VMST (also ST-VMS). In specific classes of problems, such as parachute FSI, arterial FSI, ship hydrodynamics, fluid–object interaction, aerodynamics of flapping wings, and wind-turbine aerodynamics and FSI, the scope and accuracy of the FSI modeling were increased with the special ALE-VMS and ST FSI techniques targeting each of those classes of problems. This article provides an overview of the core ALE-VMS and ST FSI techniques, their recent versions, and the special ALE-VMS and ST FSI techniques. It also provides examples of challenging problems solved and analyzed in parachute FSI, arterial FSI, ship hydrodynamics, aerodynamics of flapping wings, wind-turbine aerodynamics, and bridge-deck aerodynamics and vortex-induced vibrations.

Keywords Parachute · Artery · Flapping wings · MAV · Wind turbine · Ship hydrodynamics · Vortex-induced vibrations · FSI · Space–time · ALE · VMS

K. Takizawa
Department of Modern Mechanical Engineering and
Waseda Institute for Advanced Study, Waseda University
1-6-1 Nishi-Waseda, Shinjuku-ku, Tokyo 169-8050, JAPAN
E-mail: Kenji.Takizawa@tafsm.org

Y. Bazilevs
Department of Structural Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA

T.E. Tezduyar
Mechanical Engineering, Rice University – MS 321
6100 Main Street, Houston, TX 77005, USA

M.-C. Hsu
Department of Mechanical Engineering
Iowa State University
2025 Black Engineering, Ames, IA 50011, USA

O. Øiseth and K.M. Mathisen
Department of Structural Engineering
Norwegian University of Science and Technology
7491 Trondheim, Norway

N. Kostov and S. McIntyre
Mechanical Engineering, Rice University – MS 321
6100 Main Street, Houston, TX 77005, USA

1 Introduction

Flows with moving boundaries and interfaces include fluid–structure interaction (FSI), fluid–object interaction (FOI), fluid–particle interaction (FPI), free-surface and multi-fluid flows, and flows with solid surfaces in fast, linear or rotational relative motion. These problems are frequently encountered in engineering analysis and design, pose some of the most formidable computational challenges, and have a common core computational technology need. That crucial need motivated the development of the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) method [1; 2; 3; 4; 5; 6; 7; 8], which is a general-purpose interface-tracking (moving-mesh) technique, as a core computational technology. The DSD/SST method is an alternative to the Arbitrary Lagrangian–Eulerian (ALE) finite element formulation [9], which is the most widely used moving-mesh technique, with increased emphasis on FSI in recent years (see, for example, [10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 8; 38; 39; 40; 41; 42]). Though less widely used than the ALE formulation, over the past 20 years the DSD/SST method has been applied to some of the most challenging moving-interface problems. The classes of problems solved with the DSD/SST method since its inception include the free-surface and multi-fluid flows [1; 3; 43; 44; 45; 46], FOI [1; 2; 3; 47; 44; 48; 46], aerodynamics of flapping wings [49; 50; 51; 52; 53], flows with solid surfaces in fast, linear or rotational relative motion [44; 45; 54; 55; 28; 56], compressible flows [44], shallow-water flows [45; 57], FPI [44; 58; 59; 60; 45], and FSI [47; 61; 49; 62; 63; 64; 65; 66; 67; 68; 5; 69; 70; 71; 72; 73; 74; 75; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 89; 90; 91; 92; 6; 93; 94; 95; 96; 97; 98; 7; 99; 100; 101; 102]. Very recently, a new version of the DSD/SST method that can address the computational challenges involved in topology changes, such as contact between solid surfaces, was introduced in [103] with the name "ST-TC."

In the DSD/SST formulation, as it was originally envisioned, the ST computations are carried out one ST "slab" at a time, where the "slab" is the slice of the ST domain between the time levels n and $n + 1$. The basis functions are continuous within a ST slab, but discontinuous from one ST slab to another. The original DSD/SST method [1] is based on the SUPG/PSPG stabilization, where "SUPG" and "PSPG" stand for the Streamline-Upwind/Petrov-Galerkin [104] and Pressure-Stabilizing/Petrov-Galerkin [1; 105] methods. Starting in its very early years, the DSD/SST method also included the "LSIC" (least-squares on incompressibility constraint) stabilization. New versions of the DSD/SST method have been introduced since its inception, including those in [5], which have been serving as the core numerical technology in the majority of the ST FSI computations carried out in recent years. The most recent DSD/SST method is the ST version [6; 7] of the residual-based variational multiscale (RBVMS) method [106; 107; 108; 109]. It was named "DSD/SST-VMST" (i.e. the version with the VMS turbulence model) in [6], which was also called "ST-VMS" in [7]. The original DSD/SST method was named "DSD/SST-SUPS" in [6] (i.e. the version with the SUPG/PSPG stabilization), which was also called "ST-SUPS" in [8].

The ALE-VMS formulation [98; 28] is a moving-domain extension of the RBVMS formulation, originally proposed in [108] and successfully applied to simulation of turbulent flows and FSI in [12; 110; 18; 16; 23; 24; 109; 111; 39; 112]. An important additional feature of the ALE-VMS methodology is weak enforcement of essential boundary conditions. Weakly enforced essential boundary conditions were introduced in [113]. Weak boundary conditions produce significantly more accurate solutions than strongly enforced boundary conditions on meshes with insufficient boundary-layer resolution [114; 109; 30], which is almost always the case in practice. The ALE-VMS method with weakly enforced boundary conditions is the main computational technology behind the ALE-VMS computations presented in this article.

The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) [45] was introduced primarily for FOI with multiple fluids (see, for example, [115; 116]). The MITICT was successfully tested in [46], where the interface-tracking technique used was an ST formulation, and the interface-capturing method was the Edge-Tracked Interface Locator Technique (ETILT) [45]. It was also tested in [117] by using a moving Lagrangian interface technique [118] for interface tracking and the ETILT for interface capturing.

In this article, the ALE-VMS formulation is described in the context of the MITICT technique, in which the air-water interface is captured using the level set approach [119]. The level set function, which is convected with the flow, is used for separating the air and water subdomains. The Navier–Stokes equations of incompressible flows are employed in both air and water subdomains. The Navier–Stokes and level set equations are written in an ALE frame [9]. The rigid object is described using balance equations of linear and angular momentum. The ALE technique is employed to track the interface between the moving fluid domain (consisting of air and water subdomains) and the rigid object. Application of the ALE-VMS formulation to free-surface flow and FOI may be found in [120; 25; 29; 32].

Moving-mesh methods require mesh update methods. Mesh update typically consists of moving the mesh for as long as possible and remeshing as needed. With the key objectives being to maintain the element quality near solid surfaces and to minimize frequency of remeshing, a number of advanced mesh update methods [121; 43; 122; 45; 5] were developed to be used with the DSD/SST method, including those that minimize the deformation of the layers of small elements placed near solid surfaces.

An ST method will naturally involve more computational cost per time step than an ALE method, but it gives us the option of using higher-order basis functions in time, including the NURBS basis functions, which have been used very effectively as spatial basis functions (see [123; 12; 16; 124]). This of course increases the order of accuracy in the computations [89; 6; 7], and the desired accuracy can be attained with larger time steps, but there are positive consequences beyond that. The ST context provides us better accuracy and efficiency in temporal representation of the motion and deformation of the moving interfaces and volume meshes, and better efficiency in remeshing. This has been demonstrated in a number of 3D computations, specifically, flapping-wing aerodynamics [50; 51; 52; 53], separation aerodynamics of spacecraft [100], and wind-turbine aerodynamics [56].

There are some advantages in using a discontinuous temporal representation in ST computations. For a given order of temporal representation, we can reach a higher order accuracy than one would reach with a continuous representation of the same order. When we need to change the spatial discretization (i.e. remesh) between two ST slabs, the temporal discontinuity between the slabs provides a natural framework for that change. There are advantages also in continuous temporal representation. We obtain a smooth solution, NURBS-based when needed. We also can deal with the computed data in a more efficient way, because we can represent the data with fewer temporal control points, and that reduces the computer storage cost. These advantages motivated the development of the ST computation techniques with continuous temporal representation (ST-C) [125].

The core and special ALE-VMS and ST FSI methods mentioned above were essentially all motivated by the need for the solution and analysis of specific classes of challenging problems, such as parachute FSI, arterial FSI, aerodynamics of flapping wings, ship hydrodynamics and FOI, and wind-turbine aerodynamics and FSI. This can be seen from the ALE-VMS and ST articles cited in the first paragraph, especially the articles since 2008, and will also be seen from the examples we will present in this paper. In the case of the parachute FSI, the special methods were motivated also by the need for supporting the design process for the NASA spacecraft parachutes.

An overview of the core and special methods will be provided in Sections 2 and 3. Examples of the challenging problems solved will be presented in Section 4, and the concluding remarks will be given in Section 5.

2 Core Methods

2.1 DSD/SST Methods

The DSD/SST method, given in its DSD/SST-SUPS (or ST-SUPS) form by the the equation below,

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q \\
&+ \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q - \int_{(P_n)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P \\
&+ \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{1}{\rho} \left[\tau_\mathrm{SUPG} \rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) \right. \\
&\qquad\qquad\qquad \left. + \tau_\mathrm{PSPG} \boldsymbol{\nabla} q^h \right] \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho r_\mathrm{C}(\mathbf{u}^h) \mathrm{d}Q = 0. \qquad (1)
\end{aligned}$$

has all the ingredients of the semi-discrete SUPG/PSPG finite element formulation. That includes the test functions, domain integrations, stress terms that have been integrated by parts, boundary integrations, and the SUPG, PSPG and LSIC stabilization terms with stabilization parameters τ_SUPG, τ_PSPG and ν_LSIC (see [5] for the definitions). The stabilization is residual based, where the residuals of the momentum equation and incompressibility constraint, $\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h)$ and $r_\mathrm{C}(\mathbf{u}^h)$, appear as factors in the stabilization terms.

In an ST formulation, the domain and boundary integrations are those associated with the ST slab (see Figure 1). The velocity and pressure ($\mathbf{u}^h$ and p^h) and the corresponding test functions ($\mathbf{w}^h$ and q^h) are continuous in space but discontinuous from one ST slab to another. While this increases the number of unknowns per grid point and the computational cost per time step, it also increases the accuracy of the formulation as can be clearly seen in Figure 2. We note that the version of the DSD/SST formulation that uses lower-order functions in time would be comparable to an ALE method in terms of the computational cost and algorithmic nature.

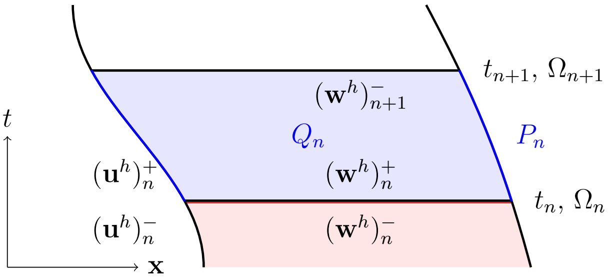


Fig. 1: ST slab.

In addition, an ST method gives us the option of using higher-order functions in time in an ST slab (see Figure 3). Using higher-order functions in temporal representation of the motion and deformation of the interfaces gives us better accuracy. In addition, using higher-order temporal functions in mesh motion and remeshing gives us a more efficient way of managing the mesh update.

Because of the moving-mesh nature of the DSD/SST method, the higher mesh resolution near fluid–solid interfaces can follow the interface, yielding a higher accuracy in resolving the boundary layers near solid surfaces. This concept is illustrated with the simple example in Figure 4. On the other hand, in a typical interface-capturing (nonmoving-mesh) method, there is no higher-resolution mesh near fluid–solid interfaces to begin with. This is illustrated in the hypothetical case shown in Figure 5. Even if one is willing to pay the price of using a very high-resolution mesh everywhere in the domain and that resolution is sufficient for some spheres, it may not be sufficient for others.

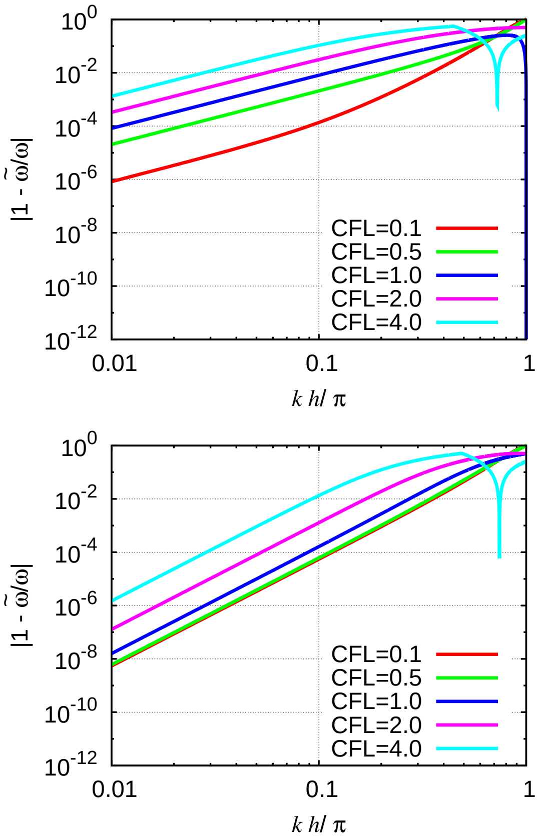


Fig. 2: Phase error, as a function of the nondimensional wave number, for the DSD/SST formulation of the time-dependent advection equation with linear functions in space. Top: Lower-order in time. Bottom: Linear in time.

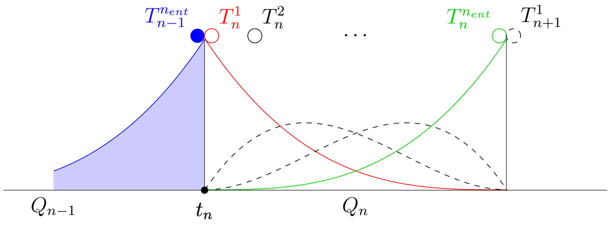


Fig. 3: Higher-order temporal basis functions in Q_n^e.

The conservative form of DSD/SST-VMST (or ST-VMS) method is given as

$$
\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \boldsymbol{\nabla} \cdot \left(\mathbf{u}^h \mathbf{u}^h \right) - \mathbf{f}^h \right) \mathrm{d}Q \\
&+ \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q - \int_{(P_n)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P \\
&+ \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) \right. \\
&\qquad \left. + \boldsymbol{\nabla} q^h \right] \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho r_\mathrm{C}(\mathbf{u}^h) \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \tau_\mathrm{SUPS} \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \left(\boldsymbol{\nabla} \mathbf{w}^h \right) \cdot \mathbf{u}^h \mathrm{d}Q \\
&- \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}^2}{\rho} \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \left(\boldsymbol{\nabla} \mathbf{w}^h \right) \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&= 0, \qquad (2)
\end{aligned}
$$

and the convective form as

$$
\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q
$$

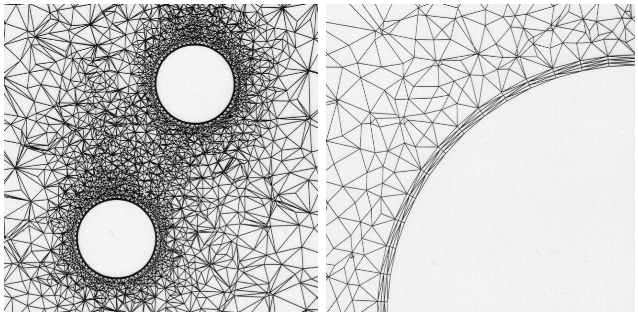


Fig. 4: Moving mesh for two spheres in a flow field. Mesh resolution near the fluid–solid interface.

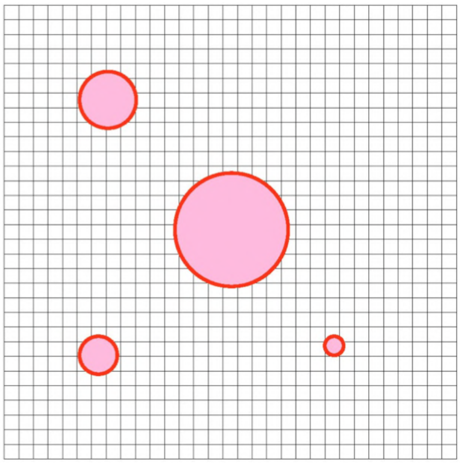


Fig. 5: In a nonmoving-mesh method, there is no higher-resolution mesh near fluid–solid interfaces. Even if the mesh resolution everywhere is sufficient for some spheres, it may not be sufficient for others.

$$
\begin{aligned}
&+ \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h)\mathrm{d}Q - \int_{(P_n)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P \\
&+ \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) \right. \\
&\qquad\qquad \left. + \boldsymbol{\nabla} q^h \right] \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h)\mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho r_\mathrm{C}(\mathbf{u}^h)\mathrm{d}Q \\
&- \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \tau_\mathrm{SUPS} \mathbf{w}^h \cdot \left(\mathbf{r}_\mathrm{M} \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) \mathrm{d}Q \\
&- \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}^2}{\rho} \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \left(\boldsymbol{\nabla} \mathbf{w}^h \right) \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h)\mathrm{d}Q \\
&= 0,
\end{aligned} \tag{3}
$$

where τ_SUPS is defined in essentially the same way as τ_SUPG is defined in [5].

Full discretization of a moving-mesh FSI formulation leads to coupled, nonlinear equation systems that need to be solved at every time step. The FSI coupling technique determines how the coupling between the equation blocks representing the fluid mechanics, structural mechanics and mesh moving equations is handled. It is essential to have a robust coupling method, especially when the structure is light and therefore very sensitive to the changes in the fluid mechanics forces. The coupling method used with the DSD/SST formulation evolved over the years from block-iterative coupling to a more robust version of block-iterative coupling, to quasi-direct coupling and direct coupling (see [5] for the terminology). The quasi-direct and direct coupling methods are applicable to cases with nonmatching fluid and structure meshes at the interface, become equivalent to monolithic methods when the interface meshes are matching, and yield more robust algorithms for FSI computations where the structure is light.

2.2 ALE-VMS Methods for FOI with Free Surface

We describe here the ALE-based MITICT formulation. Let $\mathcal{V}^h$ denote the discrete trial function space for the velocity-pressure-level set triple $\{\mathbf{u}^h, p^h, \phi^h\}$ and let $\mathcal{W}^h$ denote the discrete test function space for the linear momentum, continuity and level set equations $\{\mathbf{w}^h, q^h, \eta^h\}$. Let $\mathbf{u}_\mathrm{CM}$ and $\boldsymbol{\omega}$ denote the rigid object center-of-mass velocity and angular velocity, respectively. The ALE-VMS formulation of FOI with free surface may be stated as follows: find $\{\mathbf{u}^h, p^h, \phi^h\} \in \mathcal{V}^h$, $\mathbf{u}_\mathrm{CM}$, and $\boldsymbol{\omega}$, such that $\forall \{\boldsymbol{w}^h, q^h, \eta^h\} \in \mathcal{W}^h$,

$$
\begin{aligned}
&\int_{\Omega_t} \mathbf{w}^h \cdot \rho \left(\left. \frac{\partial \mathbf{u}^h}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u}^h - \mathbf{v}^h \right) \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega \\
&+ \int_{\Omega_t} \boldsymbol{\varepsilon}\left(\mathbf{w}^h \right) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h \right) \mathrm{d}\Omega \\
&- \int_{(\Gamma_t)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}\Gamma + \int_{\Omega_t} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}\Omega \\
&+ \sum_{e=1}^{n_\mathrm{el}} \int_{\Omega_t^e} \tau_\mathrm{SUPS} \left(\left(\mathbf{u}^h - \mathbf{v}^h \right) \cdot \boldsymbol{\nabla} \mathbf{w}^h + \frac{\boldsymbol{\nabla} q^h}{\rho} \right) \cdot \\
&\qquad\qquad \mathbf{r}_\mathrm{M}\left(\mathbf{u}^h, p^h \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{n_\mathrm{el}} \int_{\Omega_t^e} \rho \nu_\mathrm{LSIC} \boldsymbol{\nabla} \cdot \mathbf{w}^h r_\mathrm{C}(\mathbf{u}^h) \mathrm{d}\Omega \\
&- \sum_{e=1}^{n_\mathrm{el}} \int_{\Omega_t^e} \tau_\mathrm{SUPS} \mathbf{w}^h \cdot \left(\mathbf{r}_\mathrm{M}\left(\mathbf{u}^h, p^h \right) \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) \mathrm{d}\Omega \\
&- \sum_{e=1}^{n_\mathrm{el}} \int_{\Omega_t^e} \frac{\boldsymbol{\nabla} \mathbf{w}^h}{\rho} : \left(\tau_\mathrm{SUPS} \mathbf{r}_\mathrm{M}\left(\mathbf{u}^h, p^h \right) \right) \otimes \\
&\qquad\qquad \left(\tau_\mathrm{SUPS} \mathbf{r}_\mathrm{M}\left(\mathbf{u}^h, p^h \right) \right) \mathrm{d}\Omega \\
&+ \int_{\Omega_t} \eta^h \left(\left. \frac{\partial \phi^h}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u}^h - \mathbf{v}^h \right) \cdot \boldsymbol{\nabla} \phi^h \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{n_\mathrm{el}} \int_{\Omega_t^e} \tau_\mathrm{SUPG} \left(\mathbf{u}^h - \mathbf{v}^h \right) \cdot \boldsymbol{\nabla} \eta^h r_\mathrm{L}\left(\phi^h, \mathbf{u}^h \right) \mathrm{d}\Omega = 0,
\end{aligned} \tag{4}
$$

$$
\frac{\mathrm{d}}{\mathrm{d}t} \left(m \mathbf{u}_\mathrm{CM} \right) - m\mathbf{g} - \int_{\Gamma_\mathrm{I}} \mathbf{h}^h \mathrm{d}\Gamma = \mathbf{0}, \tag{5}
$$

$$
\frac{\mathrm{d}}{\mathrm{d}t} \left(\mathbf{J} \boldsymbol{\omega} \right) - \int_{\Gamma_\mathrm{I}} \left(\mathbf{x} - \mathbf{x}_\mathrm{CM} \right) \times \mathbf{h}^h \mathrm{d}\Gamma = \mathbf{0}. \tag{6}
$$

In the above coupled formulation, Ω_t is the moving domain of the fluid mechanics problem, $\mathbf{v}^h$ is the mesh velocity, $(\Gamma_t)_\mathrm{h}$ is the traction boundary, Ω_t^e is the domain of element e, Γ_I is the interface between the fluid and the rigid object in the current configuration, and $r_\mathrm{L}\left(\phi^h, \mathbf{u}^h\right)$ is the residual of the level set equation. For τ_SUPS, ν_LSIC and τ_SUPG, the definitions given in [29] are used.

From the level set function ϕ^h, the fluid density and viscosity are computed as

$$
\rho = \rho_w H_\varepsilon(\phi^h) + \rho_a (1 - H_\varepsilon(\phi^h)), \tag{7}
$$

$$
\mu = \mu_w H_\varepsilon(\phi^h) + \mu_a (1 - H_\varepsilon(\phi^h)), \tag{8}
$$

where H_ε is the *regularized* Heaviside function given by

$$
H_\varepsilon(\phi) = \begin{cases} 0 & \text{if } \phi \leq -\varepsilon; \\ \frac{1}{2}\left(1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin\left(\frac{\phi \pi}{\varepsilon} \right) \right) & \text{if } |\phi| < \varepsilon; \\ 1 & \text{if } \phi \geq \varepsilon. \end{cases} \tag{9}
$$

Here ρ_w and ρ_a are the densities of water and air, respectively, μ_w and μ_a are the dynamic viscosities of water and air, respectively, and ε is the interface half-width, which scales with the *local* mesh size (see [120; 25; 29; 32] for details).

The fluid no-slip and no-penetration boundary conditions on the surface of the rigid object are enforced weakly [113; 114]. For this, the following terms are added to the left-hand-side of Eq. (4):

$$
\begin{aligned}
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma^b \bigcap \Gamma_{\mathrm{I}}} \mathbf{w}^h \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \mathbf{n} \, \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma^b \bigcap \Gamma_{\mathrm{I}}} \left(2\mu \boldsymbol{\varepsilon}\left(\mathbf{w}^h\right) \mathbf{n} + q^h \mathbf{n}\right) \cdot \left(\mathbf{u}^h - \mathbf{v}^h\right) \, \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma^b \bigcap \Gamma_{\mathrm{I}}^-} \mathbf{w}^h \cdot \rho \left(\left(\mathbf{u}^h - \mathbf{v}^h\right) \cdot \mathbf{n}\right) \left(\mathbf{u}^h - \mathbf{v}^h\right) \, \mathrm{d}\Gamma \\
&+\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma^b \bigcap \Gamma_{\mathrm{I}}} \tau_{\mathrm{TAN}}^{B} \left(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}\right) \mathbf{w}^h \\
&\qquad\qquad \cdot \left(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}\right) \left(\mathbf{u}^h - \mathbf{v}^h\right) \, \mathrm{d}\Gamma \\
&+\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma^b \bigcap \Gamma_{\mathrm{I}}} \tau_{\mathrm{NOR}}^{B} \left(\mathbf{w}^h \cdot \mathbf{n}\right) \left(\left(\mathbf{u}^h - \mathbf{v}^h\right) \cdot \mathbf{n}\right) \, \mathrm{d}\Gamma \\
&-\sum_{b=1}^{n_{\mathrm{eb}}} \int_{\Gamma^b \bigcap \Gamma_{\mathrm{I}}^-} w^h \left(\left(\mathbf{u}^h - \mathbf{v}^h\right) \cdot \mathbf{n}\right) \phi^h \, \mathrm{d}\Gamma,
\end{aligned}
\tag{10}
$$

where the "inflow boundary" Γ_{I}^- is defined as

$$
\Gamma_{\mathrm{I}}^- = \left\{ \mathbf{x} \mid \left(\mathbf{u}^h - \mathbf{v}^h\right) \cdot \mathbf{n} < 0, \ \forall \mathbf{x} \subset \Gamma_{\mathrm{I}} \right\}. \tag{11}
$$

The last term in Eq. (10) enhances the stability of the level set formulation. The details of the boundary stabilization parameters τ_{TAN}^{B} and τ_{NOR}^{B} may be found in [114].

The following definition is employed for the computation of the fluid traction vector $\mathbf{h}^h$ acting on the rigid object:

$$
\begin{aligned}
\mathbf{h}^h = -p^h \mathbf{n} + 2\mu \boldsymbol{\varepsilon}\left(\mathbf{u}^h\right) \mathbf{n} + \tau_{\mathrm{TAN}}^{B} \left(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}\right) \left(\mathbf{u}^h - \mathbf{v}^h\right) \\
+ \tau_{\mathrm{NOR}}^{B} \left(\left(\mathbf{u}^h - \mathbf{v}^h\right) \cdot \mathbf{n}\right) \mathbf{n}.
\end{aligned}
\tag{12}
$$

The expression given in Eq. (12) is motivated by the global linear-momentum conservation of the underlying discrete formulation (see [109] for more details). Note that the expression for the traction vector contains the pressure and viscous traction terms as well as terms coming from weak enforcement of no-slip and no-penetration boundary conditions. The latter terms enhance the accuracy of the traction vector on coarser meshes, and vanish with mesh refinement near the fluid–object interface.

The governing equations for the rigid object, Eqs. (5) and (6), represent the balance of global linear and angular momentum, respectively. Here m is the mass of the object, $\mathbf{J}$ is its moment of inertia in the current configuration, $\mathbf{x}_{\mathrm{CM}}$ is the center-of-mass position vector, and $\mathbf{g}$ is the gravitational acceleration. The reference-configuration rigid-object surface geometry $(\Gamma_{\mathrm{I}})_0$, m, and reference-configuration inertia tensor $\mathbf{J}_0$ completely specify a given rigid object. The inertia tensor in the current configuration may be computed as

$$
\mathbf{J} = \mathbf{R} \mathbf{J}_0 \mathbf{R}^T, \tag{13}
$$

where $\mathbf{R}$ is the rotation matrix for the rigid object. The rotation matrix is governed by an additional ordinary differential equation:

$$
\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{R} = \boldsymbol{\omega} \times \mathbf{R}, \tag{14}
$$

where the cross product is interpreted column-wise (for derivation, see [126]). Integrating Eq. (14) in time using a midpoint rule ensures that $\mathbf{R}$ remains a proper rotation matrix at every time step (that is, $\mathbf{R}\mathbf{R}^T = \mathbf{R}\mathbf{R}^T = \mathbf{I}$; this result is due to [127]).

The fluid mesh at the interface moves with the rigid object. In the interior of the fluid domain, the mesh update method with Jacobian-based stiffening [121; 43; 122; 8], which protects the good quality of the boundary-layer elements near the solid surface, is used.

To further stabilize the free-surface formulation, YZβ discontinuity capturing, introduced in [128] and tested in [128; 129; 130; 131; 132; 133], is employed in the Navier–Stokes and level set equations. In the case of the Navier–Stokes equations, $\beta = 2$ version is used, which corresponds to the least intrusive definition. In the case of the level set equation, $\beta = 1$ version is used. The discontinuity-capturing parameters are residual-based and isotropic. Alternative definitions, which are not residual based, can be found in [134] in the context of incompressible flows and in [135] in the context of incompressible flows with thermal coupling.

The generalized-α method (see [136; 137; 16]) is employed to integrate the free-surface equations is time. The rigid-body equations are integrated with a midpoint method to preserve the orthonormal property of the column vectors of $\mathbf{R}$. At the level of the Newton–Raphson iteration, the Navier–Stokes and level set equations are solved simultaneously as a coupled 5×5-block system. The rigid-body and mesh moving equations are handled in a block-iterative fashion (see [138; 65; 5] for the terminology). The level set redistancing and mass correction are performed at the end of each time step. The benefits of monolithic coupling between the Navier–Stokes and level set equations are demonstrated in [29].

Remark 1 *Linear and angular springs and dampers are often used in combination with the rigid objects for the modeling of mechanical systems. In this case the rigid-object motion Eqs.(5) and (6) may be augmented as follows:*

$$
\frac{\mathrm{d}}{\mathrm{d}t} \left(m \mathbf{u}_{\mathrm{CM}}\right) + \mathbf{C}_{\mathrm{L}} \mathbf{u}_{\mathrm{CM}} + \mathbf{K}_{\mathrm{L}} \mathbf{y}_{\mathrm{CM}} - m\mathbf{g} - \int_{\Gamma_{\mathrm{I}}} \mathbf{h}^h \, \mathrm{d}\Gamma = \mathbf{0}, \tag{15}
$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{J}\boldsymbol{\omega}\right)+\mathbf{C}_{\mathrm{A}}\boldsymbol{\omega}+\mathbf{K}_{\mathrm{A}}\boldsymbol{\theta}-\int_{\Gamma_{\mathrm{I}}}\left(\mathbf{x}-\mathbf{x}_{\mathrm{CM}}\right)\times\mathbf{h}^{h}\ \mathrm{d}\Gamma=\mathbf{0}, \tag{16}$$

where $\mathbf{C}_{\mathrm{L}}$ *and* $\mathbf{K}_{\mathrm{L}}$ *are the linear damping and stiffness matrices, respectively,* $\mathbf{C}_{\mathrm{A}}$ *and* $\mathbf{K}_{\mathrm{A}}$ *are their angular counterparts,* $\mathbf{y}_{\mathrm{CM}}$ *is the center-of-mass displacement obtained from the kinematics relation*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}_{\mathrm{CM}}-\mathbf{u}_{\mathrm{CM}}=\mathbf{0}, \tag{17}$$

and $\boldsymbol{\theta}$ *is the angular displacement, which is a nonlinear function of the entries of the rotation matrix* $\mathbf{R}$.

Remark 2 *The linear damping and stiffness terms may be replaced by their nonlinear counterparts, and coupling between the linear and angular damping and stiffness may also be introduced.*

Remark 3 *A single fluid formulation may be obtained by setting the fluid density and dynamic viscosity to the desired values, and omitting the solution of the level-set equations all together.*

3 Special Methods

A certain class of FSI problems might involve some specific computational challenges beyond those encountered in a typical FSI problem. That requires development of special FSI methods targeting those challenges. A good number of special methods were developed in conjunction with the core ST FSI method to address the specific computational challenges involved in parachute FSI [97], patient-specific arterial FSI [92], aerodynamics of flapping wings [51; 52], and wind-turbine aerodynamics [56]. The details on these special methods can be found in the references cited above. Here we give three examples.

3.1 Homogenization Model for Ringsail Parachutes

Parachute FSI involves all the computational challenges of a typical FSI problem. Spacecraft parachutes are most of the time very large ringsail parachutes that are made of a large number of gores, where a gore is the slice of the canopy between two radial reinforcement cables running from the parachute vent to the skirt (see Figure 6). Ringsail parachute gores are constructed from rings and sails, resulting in a parachute canopy with hundreds of ring gaps and sail slits (see Figure 7). The complexity created by this geometric porosity makes FSI modeling inherently challenging.

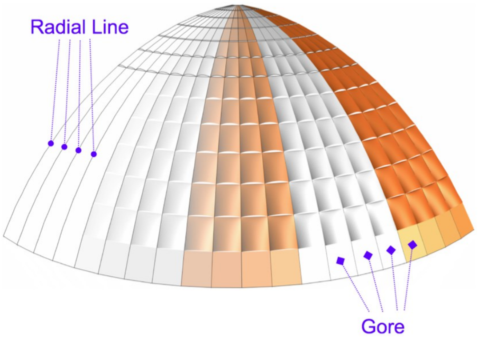


Fig. 6: Parachute radial lines and gores.

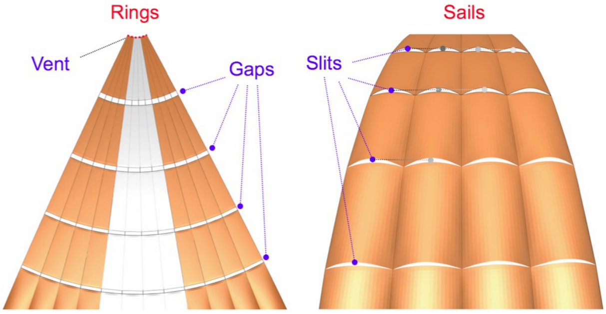


Fig. 7: Rings, sails, ring gaps, and sail slits.

The Homogenized Modeling of Geometric Porosity (HMGP) [5] and its new version, "HMGP-FG" [97], were introduced to help us bypass the intractable complexities of the geometric porosity by approximating it with an equivalent, locally varying homogenized porosity. In HMGP-FG, the normal velocity crossing the parachute canopy under a pressure differential $\varDelta p$ is modeled as

$$u_n=-\left(k_{\mathrm{F}}\right)_J\frac{A_{\mathrm{F}}}{A_1}\varDelta p-\left(k_{\mathrm{G}}\right)_J\frac{A_{\mathrm{G}}}{A_1}\mathrm{sgn}(\varDelta p)\sqrt{\frac{|\varDelta p|}{\rho}}, \tag{18}$$

where A_1, A_{F} and A_{G} are defined in Figure 8, and $\left(k_{\mathrm{F}}\right)_J$ and $\left(k_{\mathrm{G}}\right)_J$ are the homogenized porosity coefficients for each patch J, calculated in a one-time fluid mechanics computation with an n-gore slice of the parachute canopy (see Figure 9). Even in a fully open configuration, the parachute canopy goes through a periodic breathing motion where the diameter varies between its minimum and maximum values. The shapes and areas of the gaps and slits vary significantly during this breathing motion (see Figure 10). The porosity coefficients have very good in-

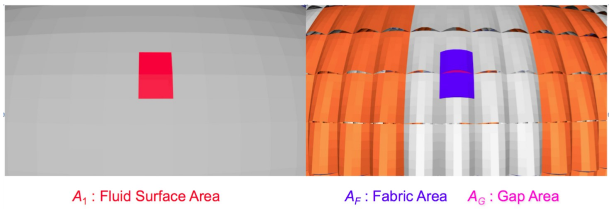


Fig. 8: Areas used in HMGP-FG.

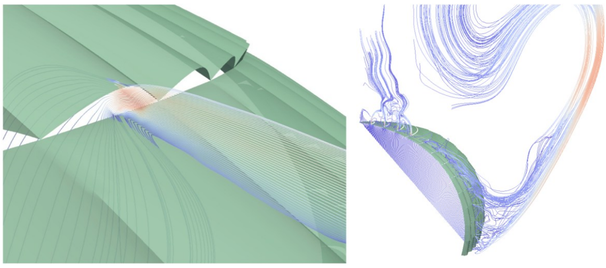


Fig. 9: The two porosity coefficients for each patch are calculated in a one-time fluid mechanics computation with an n-gore slice of the parachute canopy, where the flow through all the gaps and slits is resolved. Expect for the first and last patches, each patch contains a gap or a slit. See [5; 97] for details.

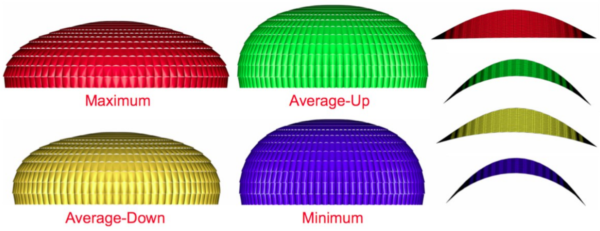


Fig. 10: The shapes and the areas of the slits vary significantly during the canopy breathing motion.

variance properties with respect to these shape and area changes, and this can be seen in Figure 11.

3.2 Flapping-Wing Motion Representation with Higher-Order Temporal Functions

Computer modeling of the aerodynamics of flapping wings requires an accurate temporal representation of the motion and deformation of the wings. It also requires robust and efficient ways of moving the mesh and remeshing as needed. Special techniques to be used in conjunction with the DSD/SST method have been developed (see [50; 51; 52]) based on using higher-order functions (specifically NURBS basis functions) in time in representing the wing motion and deformation, mesh motion, and remeshing. Using cubic NURBS basis functions in temporal representation of the wing position gives us a continuous representation of the acceleration, which in turn gives us a continuous representation of the aerodynamic forces. Using NURBS basis functions in temporal representation of the mesh motion (see Figure 12) gives us a very effective way of dealing with moving meshes. This allows us to do mesh computations with longer time in between, but get the mesh-related information, such as the coordinates and their time derivatives, from the temporal representation whenever we need it. Figure 13 illustrates how remeshing is handled in this approach. We perform multiple knot insertions where we want to remesh, and that point in time becomes a patch boundary. More details on how temporal NURBS basis functions are used in mesh motion and remeshing can be found in [50; 51].

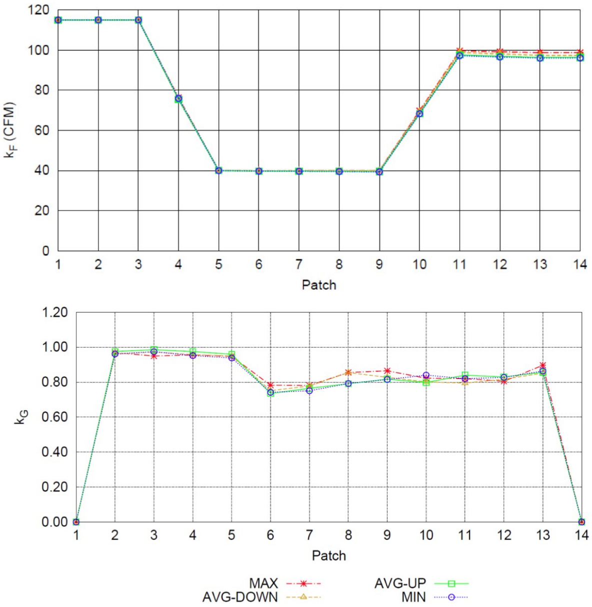


Fig. 11: The porosity coefficients $(k_{\mathrm{F}})_J$ and $(k_{\mathrm{G}})_J$ for each patch J, at different canopy shapes during the breathing motion. The plots show good invariance for these coefficients with respect to the shape changes.

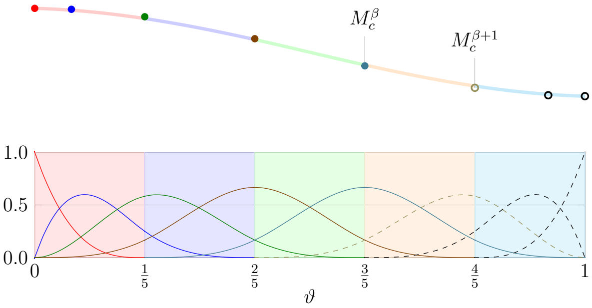


Fig. 12: Mesh motion is represented by using NURBS basis functions in time. The temporal-control meshes are the coefficients of the NURBS basis functions.

3.3 Redistancing and Mass Conservation for the Level-Set Formulation

Additional computational technology is employed to enhance the accuracy and robustness of the free-surface flow formulation. The use of a regularized Heaviside function in the definition of the fluid density and viscosity necessitates the level set to satisfy the signed-distance function property near the air–water interface. To main-

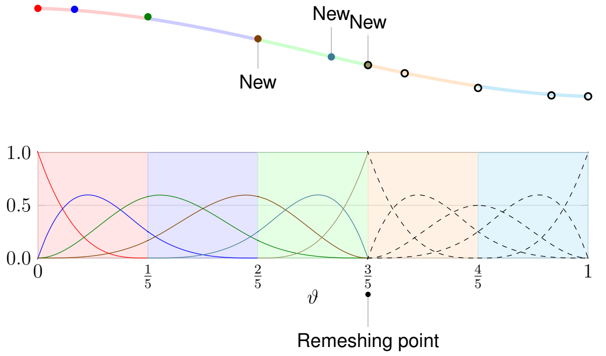


Fig. 13: Remeshing is handled by multiple knot insertion where we want to remesh. That point in time becomes a patch boundary.

tain the signed-distance property of the level set function, a *redistancing* procedure based on the Eikonal partial differential equation is employed. The details of the numerical formulation may be found in [120; 25; 29; 32].

Furthermore, both convection and redistancing of the level set do not inherently conserve mass. Convergence to a mass-conserving solution occurs only with mesh refinement. Coarse (and not-so-coarse) mesh simulations may suffer from significant water mass loss. (This depends on the problem setup and boundary conditions. In the case of liquids sloshing in closed containers, mass loss may be significant. In problems with inflow and outflow boundaries the effect may not be as pronounced.) This effect is amplified when the equations are integrated for a long time period, when seemingly small mass errors for a given time step compound into a large mass error toward the end of the computation. As a result, an explicit mass correction procedure is necessary. To ensure mass balance at every time step, after redistancing of the level set, we modify the level set function by a *global constant*, such that the following equations holds:

$$\begin{aligned}&\int_{\Omega_{n+1}} \rho_{n+1}\, d\Omega - \int_{\Omega_n} \rho_n\, \mathrm{d}\Omega \\ &+ \Delta t_{n+1} \int_{\Gamma_{n+1/2}} \rho_{n+1/2} \left(\mathbf{u}^h_{n+1/2} - \mathbf{v}^h_{n+1/2}\right) \cdot \boldsymbol{n}_{n+1/2}\, \mathrm{d}\Gamma \\ &= 0.\end{aligned} \tag{19}$$

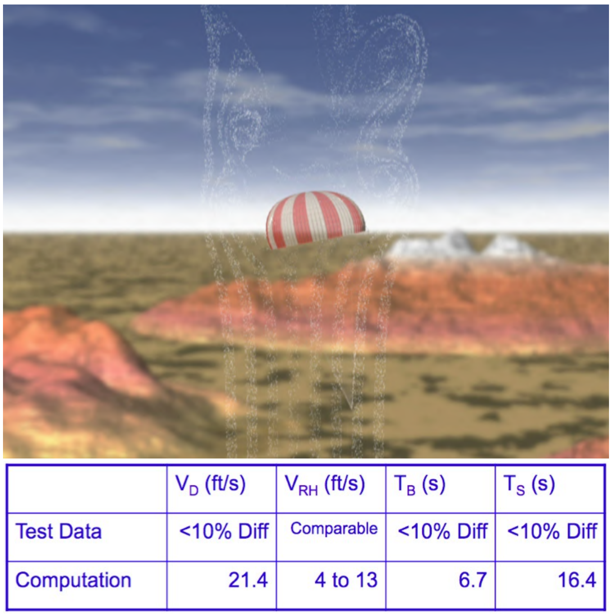
In Eq. (19), the quantities are subscripted with a temporal index and Δt_{n+1} is the time step size. This is the simplest technique that restores mass balance in the simulations. Other versions of mass correction are also possible: in [118; 139; 140] the authors proposed a total-domain based mass conservation technique, validated it experimentally in [141], and developed it in the context of MITICT (with mass conservation for fluid-solid interfaces) in [117]. A "chunk"-based (subdomain-based) version of mass conservation was developed in [45; 46].

4 Examples

Several examples of computational analysis with the methods described in the earlier sections are presented. Examples in Sections 4.1–4.4 were computed with the DSD/SST methods, and the examples in Sections 4.5–4.8 with the ALE-VMS methods.

4.1 FSI Analysis of Spacecraft Parachutes

The first example, a parachute computation, serves the purpose of comparing our computed results to data from drop tests with a base parachute design and gaining confidence in our parachute FSI model. Figure 14 shows the parachute shape and flow field at an instant during the computation and the comparison with the test data. With that confidence, we can do simulation-based design studies [97], such as evaluating the aerodynamic performance of the parachute as a function of the suspension line length (see Figure 15).



	V_D (ft/s)	V_{RH} (ft/s)	T_B (s)	T_S (s)
Test Data	<10% Diff	Comparable	<10% Diff	<10% Diff
Computation	21.4	4 to 13	6.7	16.4

Fig. 14: Parachute shape and flow field at an instant during the computation and comparison with the test data. Here $\mathrm{V_D}$, $\mathrm{V_{RH}}$, $\mathrm{T_B}$, and $\mathrm{T_S}$ are the descent speed, horizontal speed, breathing period, and swinging period.

Spacecraft parachutes are typically used in clusters of two or three parachutes. The contact between the canopies of the parachute cluster is a computational challenge that we have addressed recently (see [97]). Figure 16 shows a cluster of three parachutes at three different instants during the FSI computation, with contact between two of the parachutes.

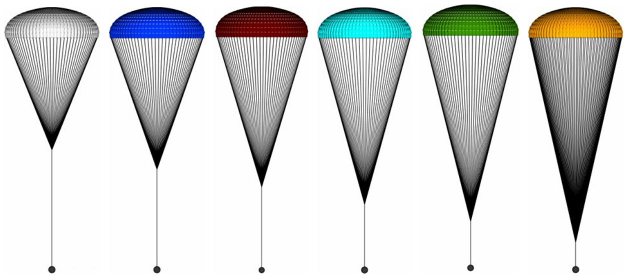


Fig. 15: A simulation-based parachute design study, where the objective is to evaluate the aerodynamic performance of the parachute as a function of the suspension line length. See [97] for details of the study.

Spacecraft parachutes are also typically used in multiple stages, starting with a “reefed” stage where a cable along the parachute skirt constrains the diameter to be less than the diameter in the subsequent stage. After a certain period of time during the descent, the cable is cut and the parachute “disreefs” (i.e. expands) to the next stage. Computing the parachute shape at the reefed stage and FSI modeling during the disreefing involve additional computational challenges created by the increased geometric complexities and by the rapid changes in the parachute geometry. Figure 17 shows such a disreefing (see [99]).

As an additional computational challenge, the ringsail parachute canopy might, by design, have some of its panels and sails removed. The purpose is to increase the aerodynamic performance of the parachute. In FSI computation of parachutes with such “modified geometric porosity,” the flow through the “windows” created by the removal of the panels and the wider gaps created by the removal of the sails cannot be accurately modeled with the HMGP and needs to be actually resolved. This challenge was successfully addressed in the computations reported in [102]. Figure 18 shows a cluster of three parachutes with modified geometric porosity, at an instant during the FSI computation.

4.2 Aerodynamic Analysis of Wind Turbines

Computer modeling of wind-turbine aerodynamics is challenging because correct aerodynamic torque calculation requires correct separation-point calculation, which requires an accurate flow field, which in turn requires good mesh resolution and turbulence model. We describe from [54] how we computed the aerodynamics of an actual wind-turbine rotor by using the DSD/SST-SUPS and DSD/SST-VMST methods. Figure 19 shows, from [23], the airfoil cross-sections of the wind-turbine blade superposed on the blade. Figure 20 shows the full wind-turbine rotor. Figure 21 shows the vorticity magnitude, computed with the DSD/SST-VMST method. In that figure, the blue and yellow correspond to low and high vorticity values, and lighter and darker shades of those two colors correspond to lower and higher values. Figure 22 shows time history of the aerodynamic torque generated by a single blade, as computed with the DST/SST-SUPS, DSD/SST-VMST, and ALE methods.

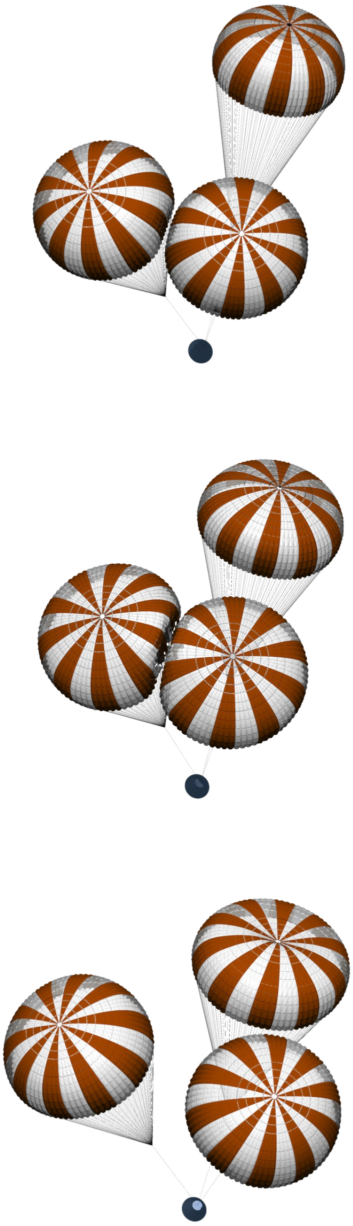


Fig. 16: A cluster of three parachutes at three instants during the FSI computation, with contact between two of the parachutes.

Including the tower in the model (see Figure 23) increases the computational challenge because of the fast, rotational relative motion between the rotor and tower.

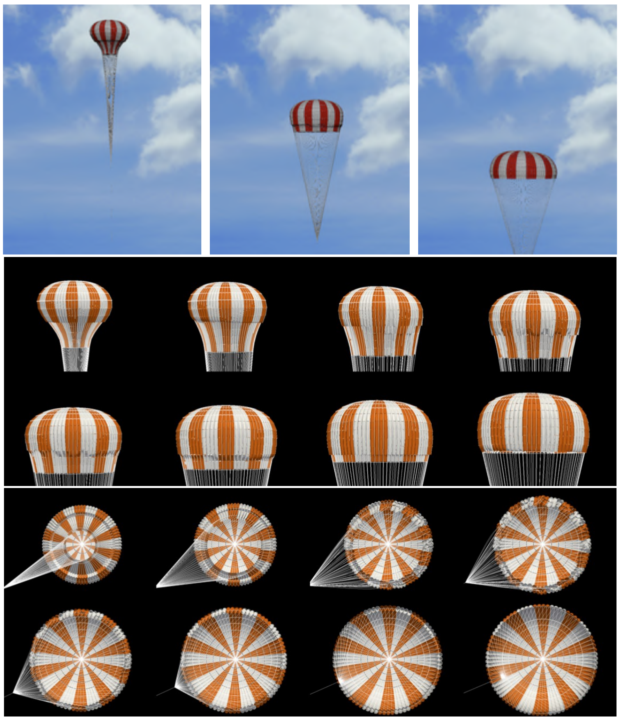


Fig. 17: Parachute disreefing from [99]: side view (top and middle) and bottom view (bottom).

We address this additional challenge in [56] by using NURBS basis functions for the temporal representation of the rotor motion, mesh motion and also in remeshing. This is essentially the same computational technology described in Section 3.2 for modeling the aerodynamics of flapping wings. We named this "ST/NURBS Mesh Update Method (STNMUM)" in [56]. Figure 24 shows, from [56], the vorticity magnitude, computed with the DST/SST-VMST method and the STNMUM. In that figure, the color range from blue to red corresponds to a vorticity range from low to high, and lighter and darker shades of a color correspond to lower and higher values.

4.3 Patient-Specific FSI Analysis of Cerebral Arteries with Aneurysm

Patient-specific computer modeling of arterial FSI has many challenges. They include calculating an estimated zero-pressure arterial geometry, specifying the velocity profile at an inflow boundary with non-circular shape, using variable arterial wall thickness, building layers of refined fluid mesh near the arterial walls, proper calculation of the wall shear stress (WSS) and oscillatory shear index (OSI), and properly scaling the flow rate at the inflow boundary. Special techniques developed to address these challenges can be found in [92]. Here we present some computations from [92] for cerebral arteries with aneurysm.

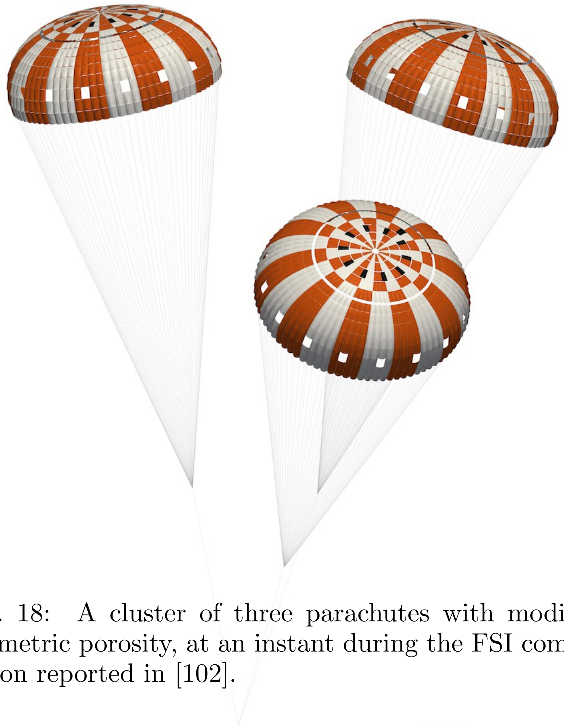


Fig. 18: A cluster of three parachutes with modified geometric porosity, at an instant during the FSI computation reported in [102].

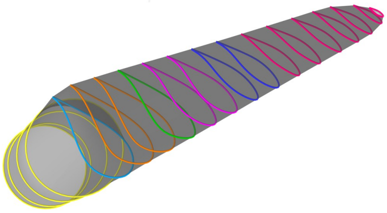


Fig. 19: Airfoil cross-sections of the wind-turbine blade superposed on the blade (from [23]).

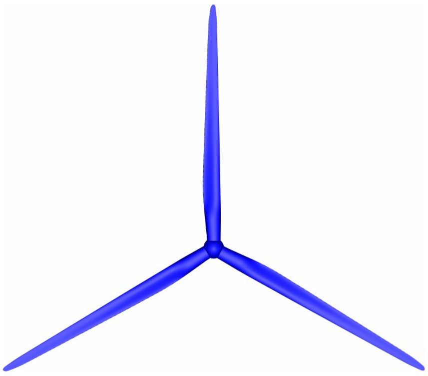


Fig. 20: Wind-turbine rotor.

Figure 25 shows the arterial lumen geometry obtained from voxel data for three arterial models we consider: Model 1, Model 2, and Model 3. Figure 26 shows, as an example, the fluid mechanics mesh for Model 3, including the layers of refined mesh near the arterial wall. Figure 27 shows, for the three models, the streamlines

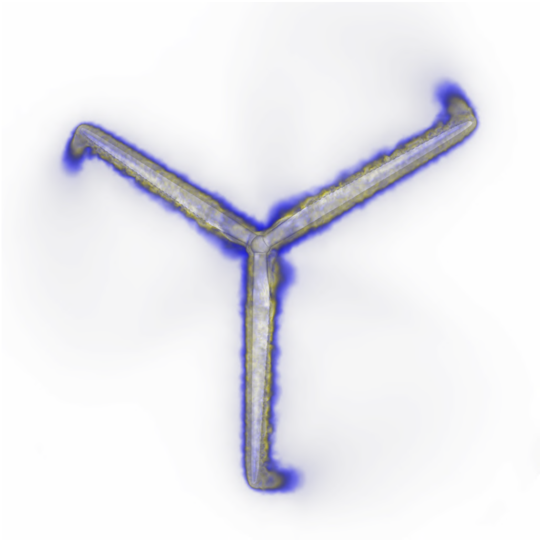


Fig. 21: Vorticity magnitude, computed with the DSD/SST-VMST method.

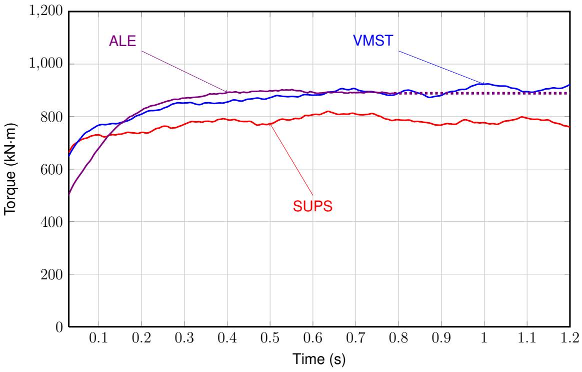


Fig. 22: Time history of the aerodynamic torque generated by a single blade. Computed with the DST/SST-SUPS ("SUPS"), DST/SST-VMST ("VMST"), and ALE methods.

when the flow rate is maximum. Figures 28 and 29 show the WSS and OSI for Model M6ACom from [92].

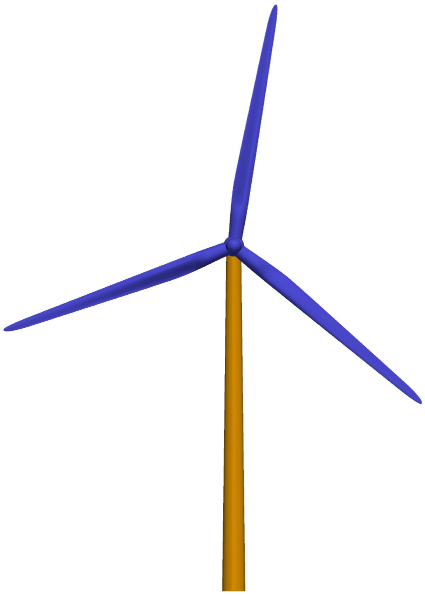


Fig. 23: Wind-turbine rotor and tower from [56].

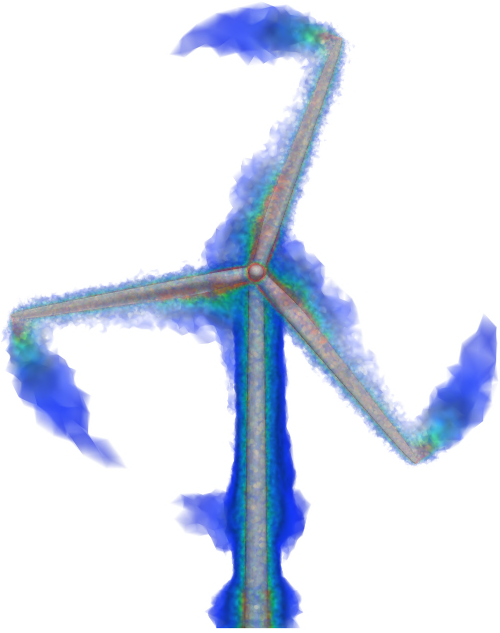


Fig. 24: Vorticity magnitude, computed with the DST/SST-VMST method and the STNMUM (see [56]).

4.4 Aerodynamic Analysis of Flapping Wings of an Actual Locust and an MAV

As a last set of examples from analyses with the ST methods, we present from [51; 52] computational aerodynamics modeling of flapping wings of an actual locust and an MAV. The motion and deformation data for the wings is extracted from the high-speed, multi-camera video recordings of a locust in a wind tunnel at Baylor College of Medicine (BCM), Houston, Texas. The video recording is accomplished by using a set of tracking points marked on the forewings (FW) and hindwings (HW) of the locust. The tracking points can be seen in Figure 30. How the wing motion and deformation data is extracted from the video data and represented using NURBS basis functions in space and time is described in detail in [51]. Figures 31 and 32 show the wind tunnel photographs and the computational model at eight points in time. Figure 33 show how the body and wings compare for the locust and MAV models, and Figure 34 shows the length scales involved in the computations with those models.

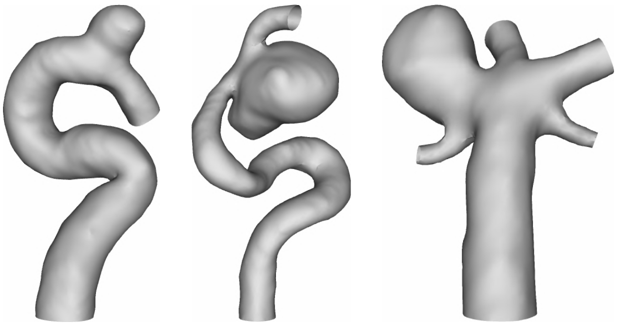


Fig. 25: Arterial lumen geometry obtained from voxel data for Model 1, Model 2, and Model 3.

Figure 35 shows the streamlines from the locust computation. Figures 36 and 37 show for the locust the vorticity magnitude during the second flapping cycle. Figures 38 and 39 show for the MAV the vorticity magnitude during the third flapping cycle. In Figures 36–39, the color range from blue to red corresponds to a vorticity range from low to high, and lighter and darker shades of a color correspond to lower and higher values. Figures 40–42 show the lift and thrust generated by the locust and MAV.

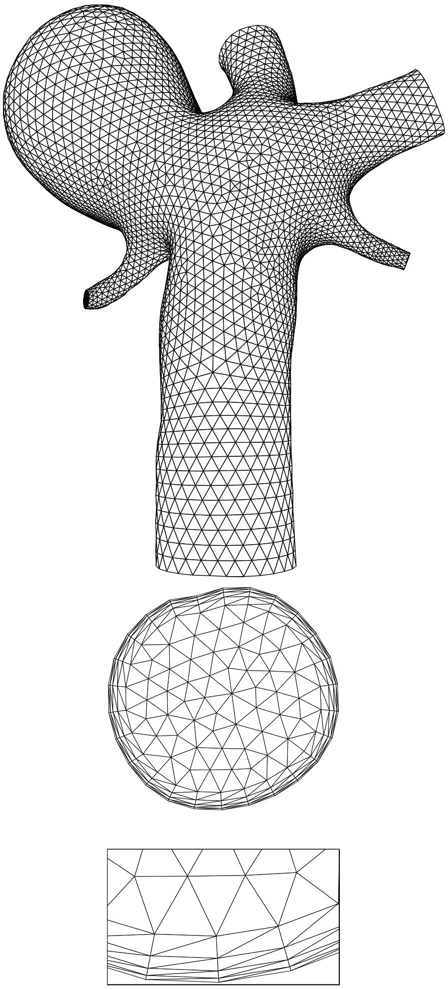


Fig. 26: Fluid mechanics mesh for Model 3. Mesh at the fluid–structure interface (top) and inflow plane (middle and bottom).

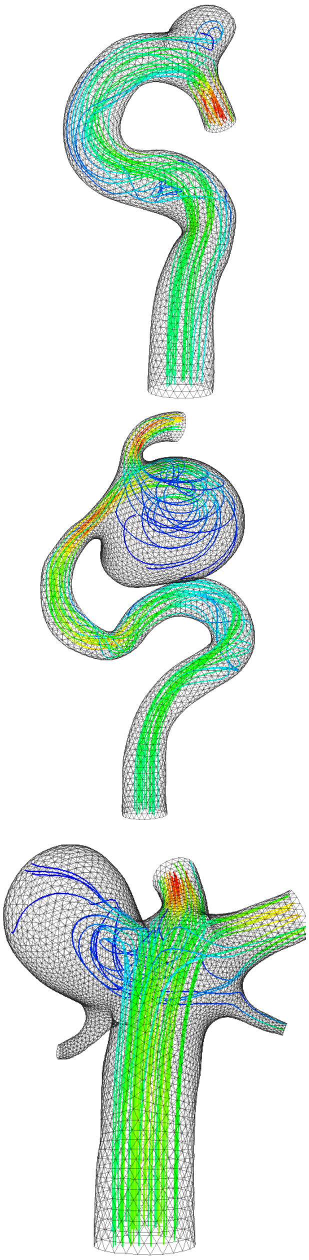


Fig. 27: Streamlines for the three models when the volumetric flow rate is maximum.

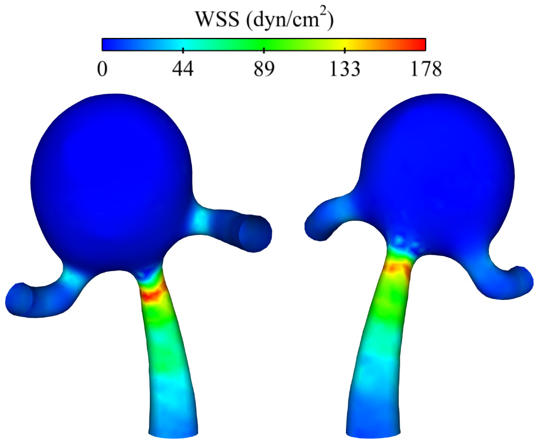


Fig. 28: WSS for Model-M6Acom from [92] when the volumetric flow rate is maximum.

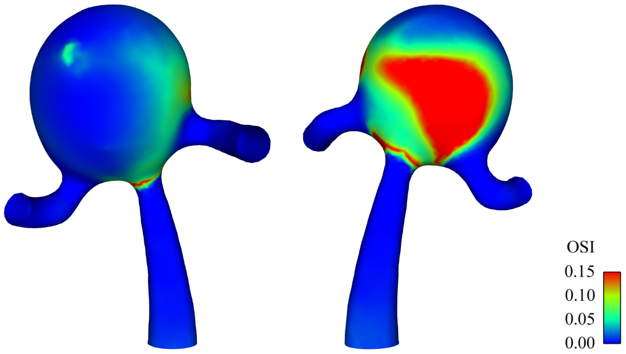


Fig. 29: OSI for Model-M6Acom from [92].

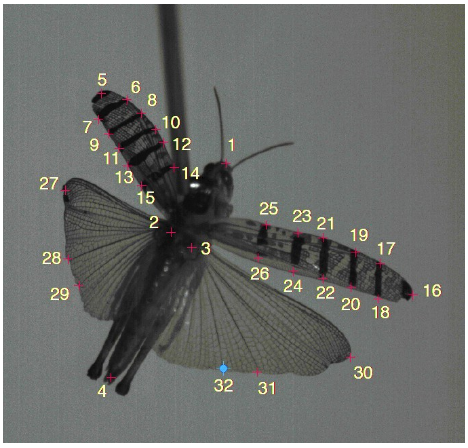


Fig. 30: Tracking points in the data set from the BCM wind tunnel.

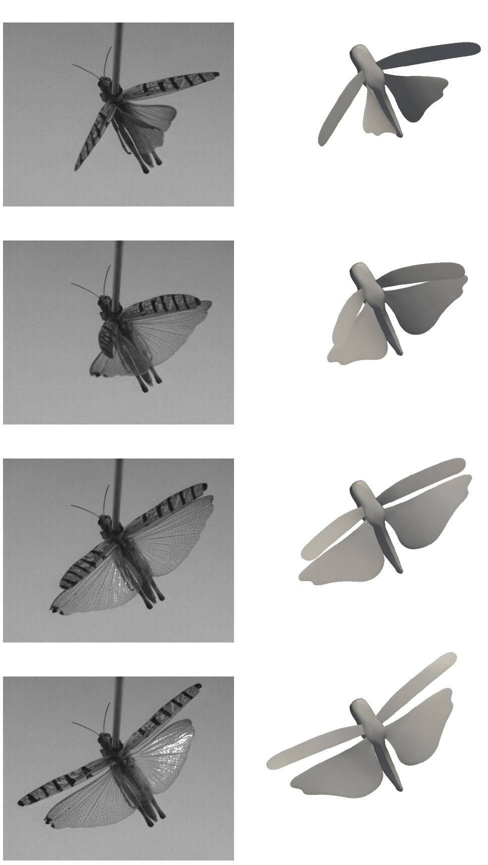


Fig. 31: Comparison of computational model and wind tunnel photographs at first four points in time. Viewing angles are matched approximately. Wind tunnel photographs provided by BCM collaborators.

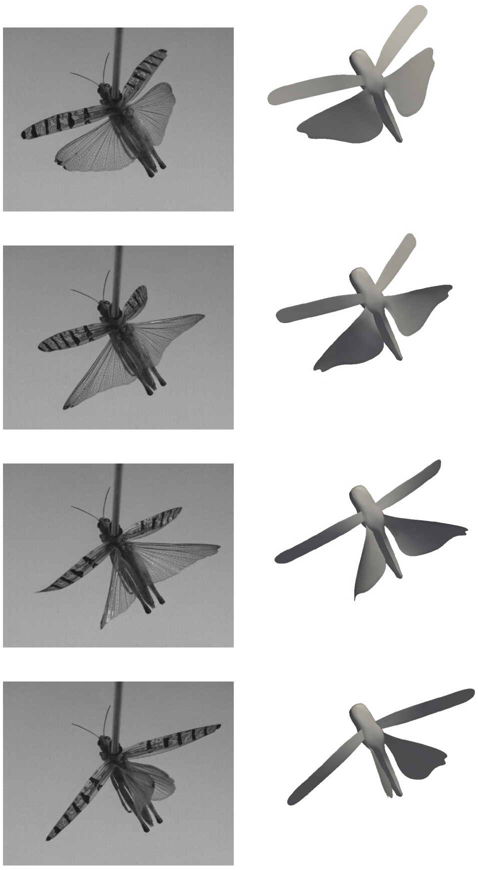


Fig. 32: Comparison of computational model and wind tunnel photographs at last four points in time. Viewing angles are matched approximately. Wind tunnel photographs provided by BCM collaborators.

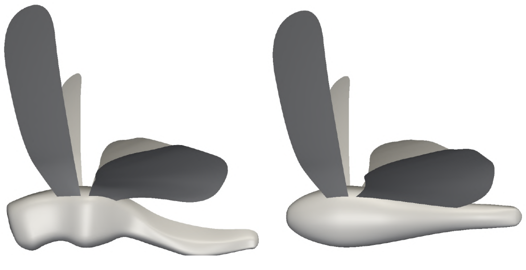


Fig. 33: Locust body and wings (left) and MAV body and wings (right).

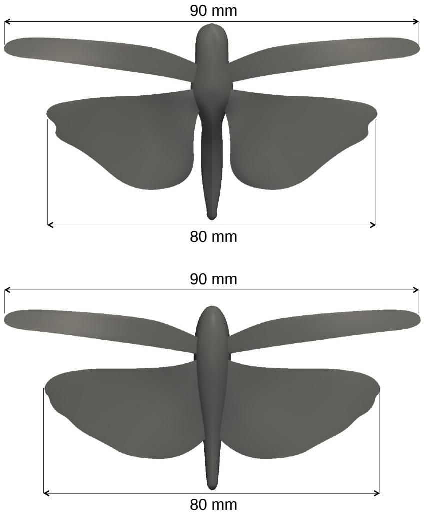


Fig. 34: Length scales involved in the computations with the locust (top) and MAV (bottom) models.

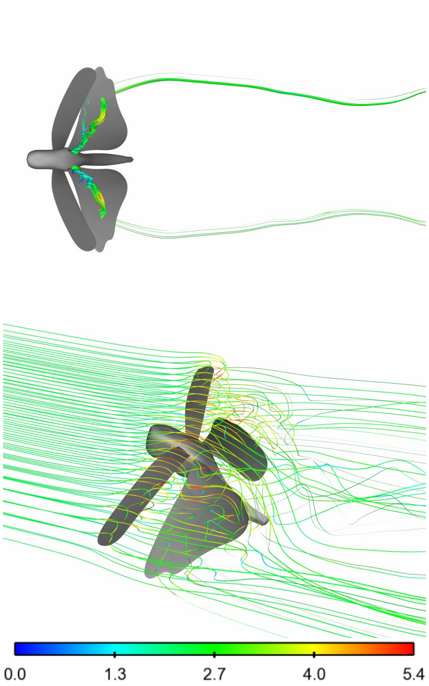


Fig. 35: Locust. Streamlines colored by velocity magnitude in m/s at approximately 25% (top) and 50% (bottom) of the second flapping cycle.

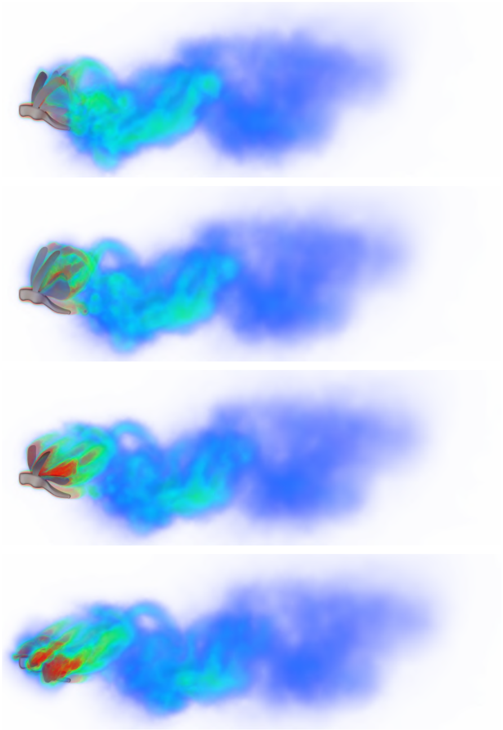


Fig. 36: Locust. Vorticity magnitude for the first four of eight equally-spaced points during the second flapping cycle.

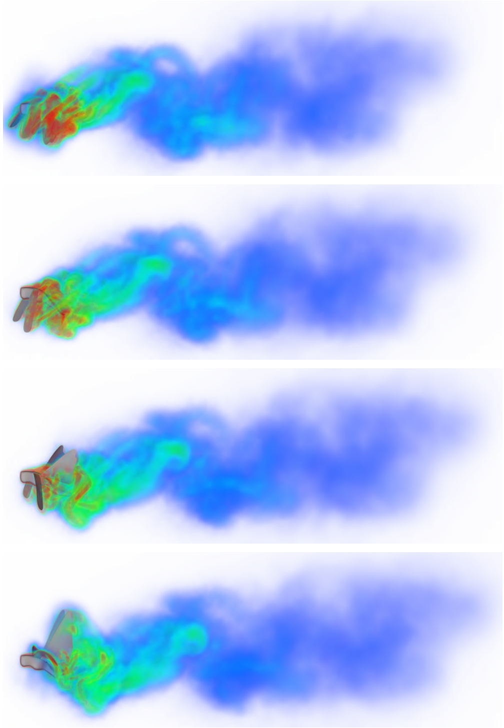


Fig. 37: Locust. Vorticity magnitude for the last four of the eight equally-spaced points during the second flapping cycle.

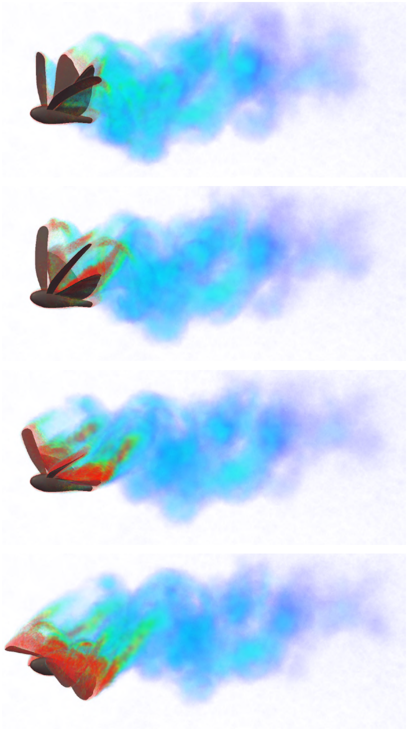


Fig. 38: MAV. Vorticity magnitude for the first four of eight equally-spaced points during the third flapping cycle.

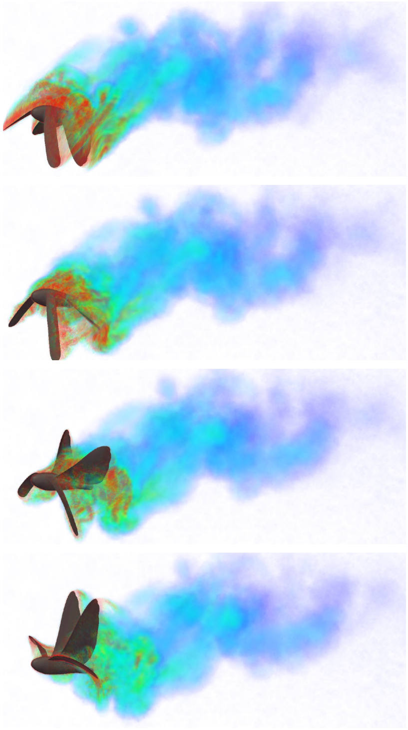


Fig. 39: MAV. Vorticity magnitude for the last four of eight equally-spaced points during the third flapping cycle.

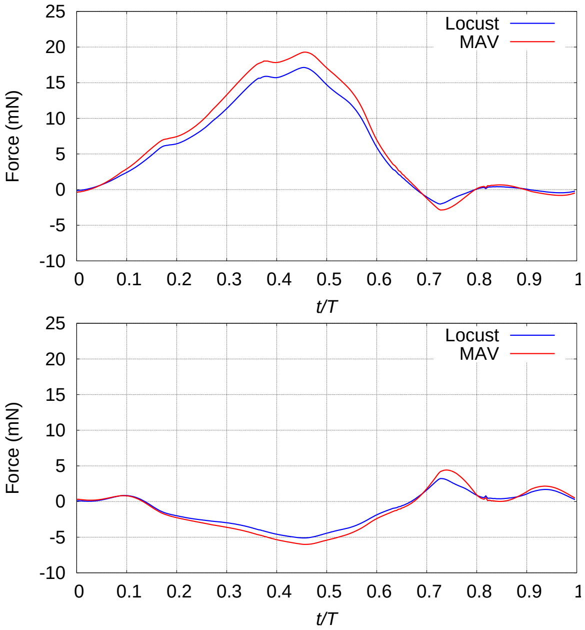


Fig. 40: Total lift (top) and thrust (bottom) generated over one cycle.

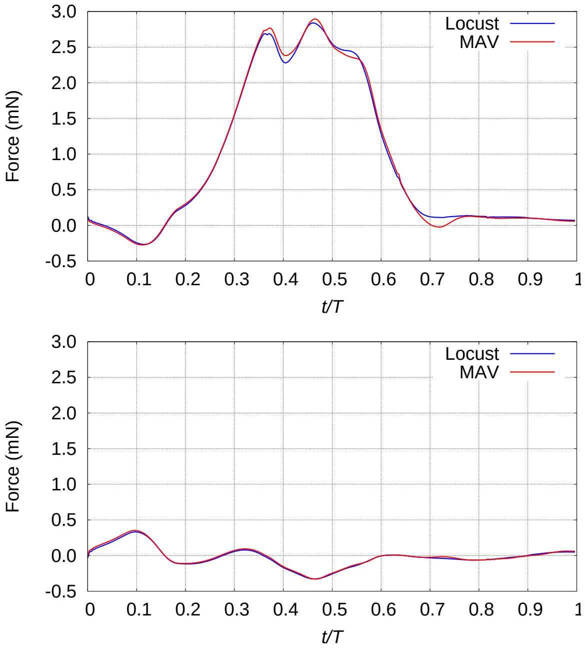


Fig. 41: Lift (top) and thrust (bottom) generated on the right FW.

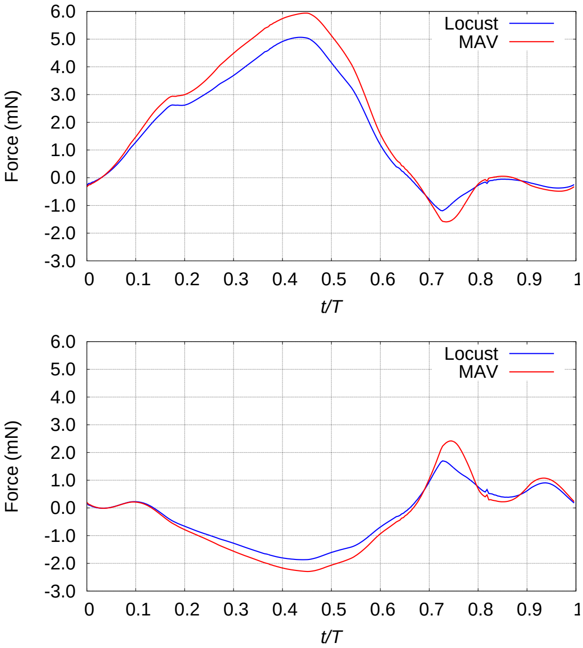


Fig. 42: Lift (top) and thrust (bottom) generated on the right HW.

4.5 The MARIN Dam Break Problem

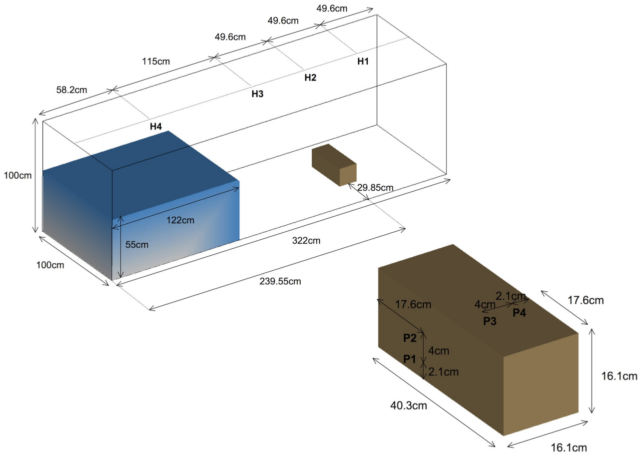


Fig. 43: The MARIN dam break problem. Geometry definition. The computational domain is a rectangular box with dimensions 3.22 m × 1 m × 1 m. The object has dimensions 0.2 m × 0.2 m × 0.4 m and is placed at the back end of the tank. The water column, initially at rest, has dimensions 1 m × 1 m × 0.55 m. The locations where pressure and water height are sampled are also depicted.

The setup of the dam break problem, initially proposed by the Maritime Research Institute Netherlands (MARIN) [142], is depicted in Figure 43, and is taken from [25]. The problem consists of a column of water, initially at rest, that collapses under the action of gravity and impacts a fixed rectangular container. We compute the problem using two types of the spatial discretization: linear tetrahedral finite elements and NURBS. The quadratic NURBS mesh is significantly more coarse than the linear tetrahedral mesh. Free-slip and no-penetration boundary conditions are applied on all surfaces, including the top of the tank. The problem is run until $T = 6$ s.

Snapshots comparing the solutions coming from tetrahedral FEM and NURBS computations are given in Figure 44. Large-scale features of the solution are very similar in the two simulations, however the details of the small-scale features are better represented on a much finer tetrahedral grid, as expected.

Time series of the pressure at different locations on the obstacle are shown in Figure 45. The first wave hits the block at approximately $t = 0.5$ s, and the second, much smaller wave arrives at the block at about $t = 5$ s. The wave impact times and pressure peaks are predicted very well with both linear elements and quadratic NURBS. Given that the NURBS mesh has about half of the degrees-of-freedom of the linear FEM mesh in each Cartesian direction, the accuracy of NURBS results is remarkable; linear FEM is not capable of attaining such accuracy at this level of resolution (see [25]), and requires a finer mesh for comparable accuracy.

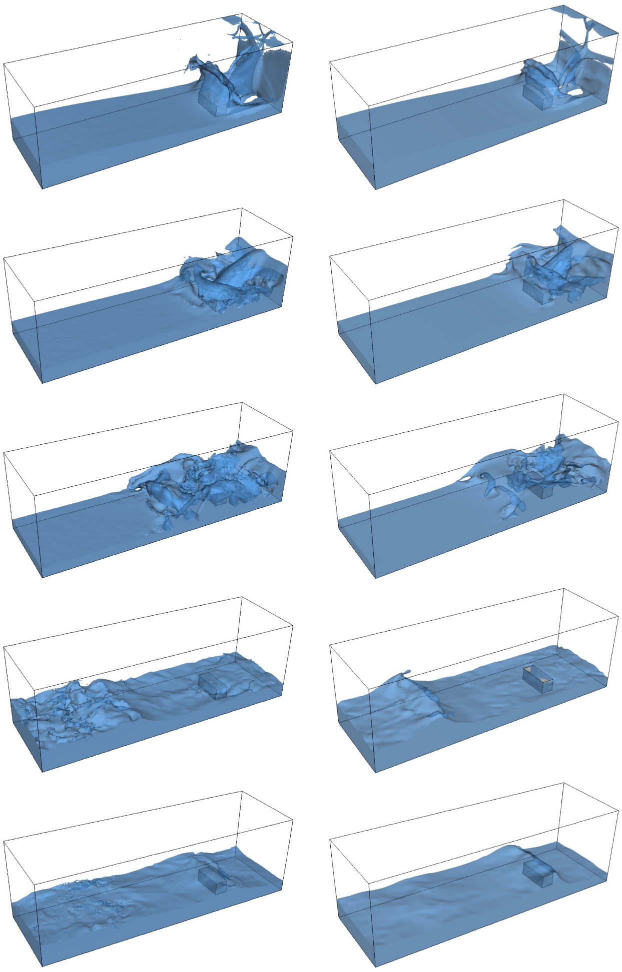


Fig. 44: The MARIN dam break problem. Snapshots of the free surface solution on the tetrahedral (left) and NURBS (right) meshes at $t = 1.0, 1.5, 2.0, 4.0,$ and 5.0 s.

4.6 Fridsma Planing Hull

In this section we present our simulation results for the Fridsma planing hull [143]. We give a detailed definition of the hull geometry, present a mesh refinement study, and assess the effect of hull speed on the drag force and trim angle. Only flat-water (i.e., no waves), constant hull speed cases are considered in this paper. The computational results presented are from [32].

The Fridsma hull geometry definition is given in Figure 46. The hull is comprised of idealized shapes: a bow consisting of four ruled surfaces followed by a wedge-shaped straight section with an constant deadrise angle of 20°. Analytical expressions for the bounding curves for the ruled surfaces are provided in the figure. The rel-

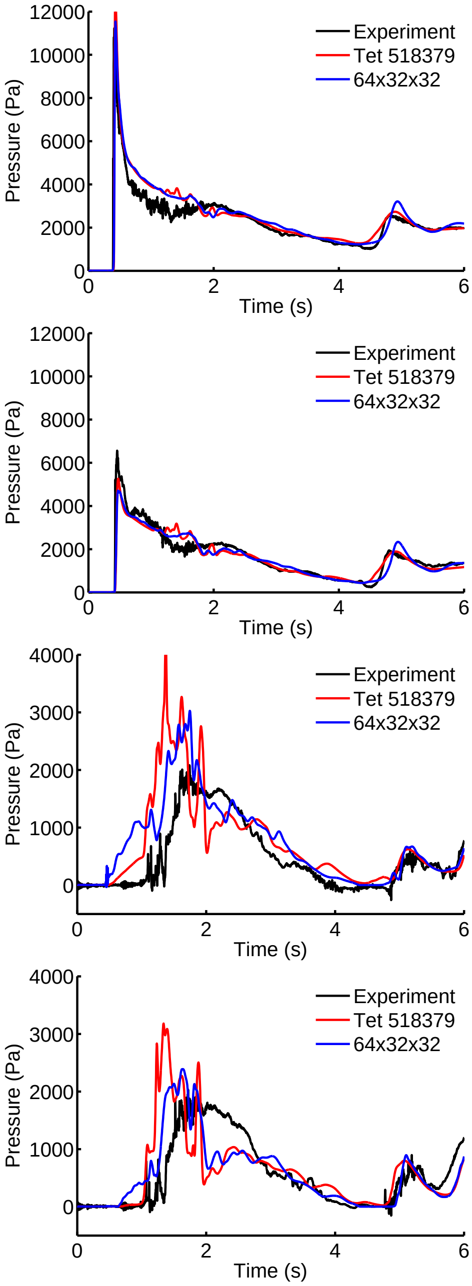


Fig. 45: The MARIN dam break problem. Time history of the pressure at four locations on the obstacle. Experimental data is from [142].

evant global geometry parameters are given in Table 1. The hull mass, center of gravity, and moment of inertia are summarized in Table 2. This data pertains to the center of gravity located at 70% of the hull length, measured from the aft of the hull [143].

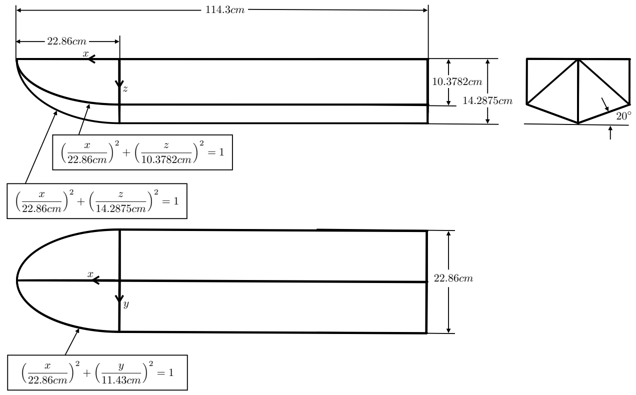


Fig. 46: Fridsma hull. Geometry definition. Top left: side view. Top right: aft view. Bottom: top view.

Length (L)	114.3 cm
Beam (b)	22.86 cm
Height	14.2875 cm
Deadrise	20°

Table 1: Fridsma hull. Global geometric parameters.

Mass (m)	7.257 kg
x_{cg}	80.01 cm
z_{cg}	6.721 cm
Gyradius (r)	25% L
$I_{yy} = mr^2$	0.6165 kg m^2

Table 2: Fridsma hull. Mass, center of gravity, and moment of inertia.

We perform a mesh convergence study at Froude number Fr = 0.8950[1]. The Froude number is defined as

$$\mathrm{Fr} = \frac{u}{\sqrt{gL}}, \tag{20}$$

where u is the hull speed, g is the magnitude of gravitational acceleration, L is the hull length. The Froude number represents the significance of inertial forces relative to gravity. At this chosen Froude number, according to [143], the trim angle reaches its maximum. A convergence study was performed on a sequence of four meshes. The coarsest mesh is shown in Figure 47. The figure also shows the water and air subdomains in the undisturbed configuration. The mesh is dense near the hull surface and in the wake. The hull is fixed in the direction of travel, and the corresponding velocity is set at the inflow of the computational domain together with the level set

---

[1] In [143] the results are reported in terms of the Speed-Length Ratio (SLR), $u/\sqrt{L}$, which is a dimensional quantity. Here we chose to report the results in terms of the Froude number, which is non-dimensional.

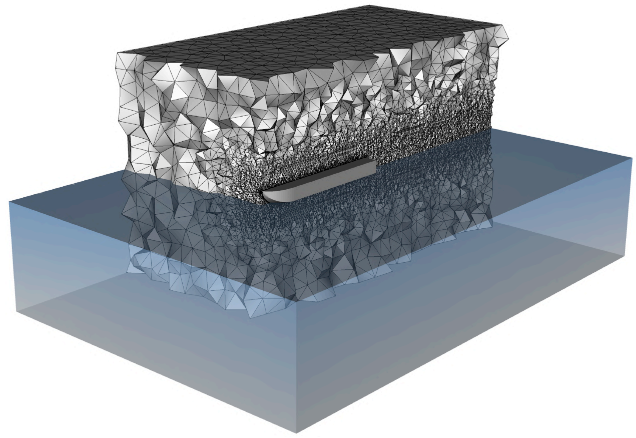


Fig. 47: Fridsma hull. Coarsest mesh with water and air domains shown.

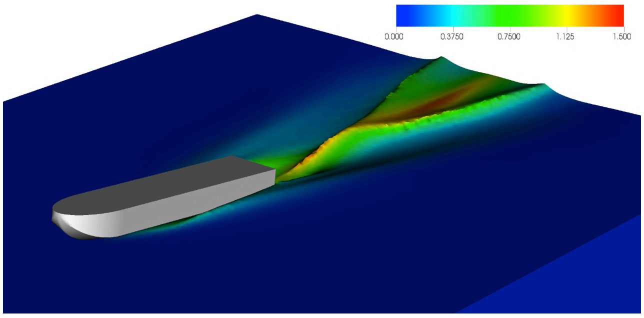
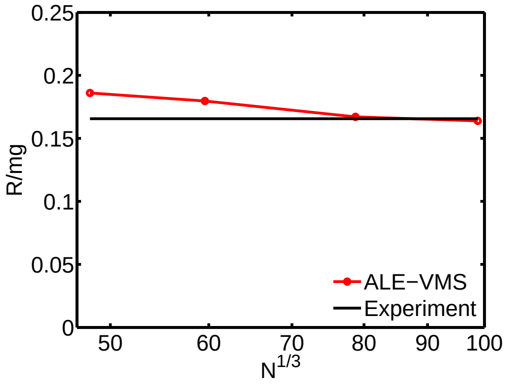
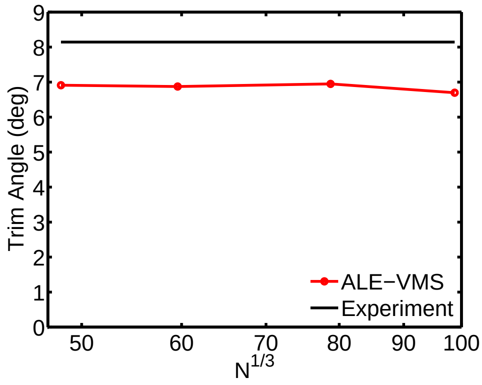


Fig. 48: Fridsma hull. Free surface colored by the flow speed relative to the hull speed in m/s.

function. The hull is allowed to pitch, and displace in the vertical direction. At the outflow a hydrostatic pressure profile is imposed as a traction boundary condition. On the side, bottom, and top boundaries of the computational domain free-slip boundary conditions are imposed. Figure 48 shows the deformed free surface colored by the flow speed relative to the hull speed. The hull rises up and develops a trim angle such that the bow is higher than the aft. Note the presence of the "rooster tail" feature, which is typical for planing hulls. Also note that the rooster tail feature goes all the way to the outflow boundary, which suggests that a longer-domain simulation may be needed in the future. Figure 49 shows convergence of the drag force and trim angle. The drag force is nondimensionalized by the gravitational force. From the results we see that the drag force converges quickly to the experimental value. On the other hand, the trim angle is underestimated by 12% with respect to the experimental data, and does not improve with mesh refinement. Possible causes for this may be the choice downstream, lateral, and bottom boundary locations. Errors in the experimental data are also possible.



(a) Drag Force



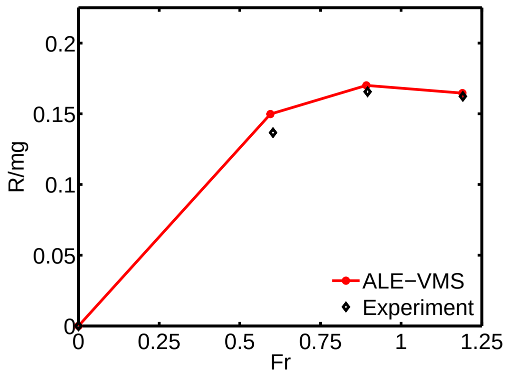
(b) Trim Angle

Fig. 49: Fridsma hull. Convergence of the drag force and trim angle with mesh refinement and comparison with experimental results.

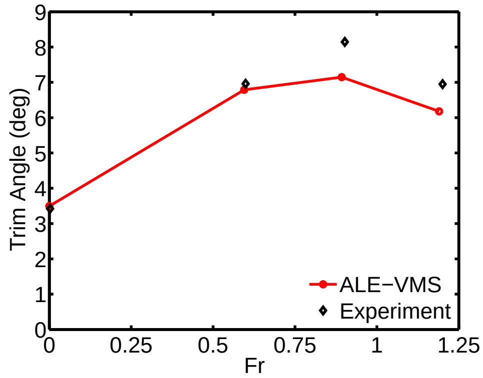
We also examine the effect of the hull speed on the drag force and trim angle is studied. In addition to the $\mathrm{Fr} = 0.8950$ case, we consider $\mathrm{Fr} = 0$, $\mathrm{Fr} = 0.5925$, and $\mathrm{Fr} = 1.190$ cases. The simulations are started impulsively in the configuration depicted in Figure 47. In the case of $\mathrm{Fr} = 0$, although the hull speed is zero, a non-zero trim angle develops such that the hull is in equilibrium with the hydrostatic forces. In all other cases, there is a rapid transient followed by a largely steady-state response. The steady-state drag force and trim angle are plotted as a function of Froude number, and compared to the experimental results in Figure 50. Accurate prediction of the drag force is attained in all cases. The trim angle is predicted very well for the first two Froude number cases, and a deviation from the experiment by 10%–12% is seen in the remaining two cases.

4.7 DTMB 5415 Navy Combatant in Head Waves

Here we present the simulation of the DTMB 5415 Navy combatant at lab scale from [29]. This ship has been investigated by other researchers, both experimentally and computationally (see, e.g., [144; 145; 146]).



(a) Drag Force



(b) Trim Angle

Fig. 50: Fridsma hull. Steady-state drag force and trim angle as a function of Froude number. Comparison with experimental results.

The length of the ship hull is 5.72 m. The ship mass, center of gravity and inertia tensor are computed by meshing the ship interior and performing a direct computation. The total ship volume is 1,366 m^3. The ship mass is equal to 532.3 kg. It is obtained by multiplying the volume of the ship below the water line by the constant water density. The center of gravity and the inertia tensor are computed assuming the ship's effective density (i.e., the ship mass divided by its total volume), which results in

$$\mathbf{X}_0 = \begin{bmatrix} 2.761 \\ 0 \\ 0.280 \end{bmatrix} \text{ m} \tag{21}$$

and

$$\mathbf{J}_0 = \begin{bmatrix} 7.256\text{E-}2 & 2.69\text{E-}7 & 5.35\text{E-}2 \\ 2.69\text{E-}7 & 2.89 & \text{-}2.44\text{E-}8 \\ 5.35\text{E-}2 & \text{-}2.44\text{E-}8 & 2.91 \end{bmatrix} \text{ kg m}^2, \tag{22}$$

respectively.

We compute the ship in head waves, meaning the waves that travel in the direction opposite to that of the ship. We assume that the ship speed is $U_{in} = 1.873$ m/s, which gives the Froude number of 0.25 based on the ship length. The ship was allowed to move vertically, to pitch and to roll, while the rest of the rigid body degrees-of-freedom were constrained.

We make use of the linear Airy waves [147] to prescribe inlet boundary conditions. The Airy waves may be derived using potential theory, and are specified as follows: Given, the wave amplitude, wave length and water depth, $A_w = 0.2$ m, $L_w = 5.72$ m and $h = 3.49$ m, respectively, we compute $k = 2\pi/L_w$, the angular wavenumber, $\omega = \sqrt{gk\tanh(kh)}$, the wave phase speed, and $A_v = \frac{\omega A_w}{\sinh(kh)}$, the velocity amplitude. With these definitions, the Airy waves are given by

$$u = A_v \cosh(kz)\cos(kx - \omega t) + U_{in} \tag{23}$$

$$v = 0 \tag{24}$$

$$w = A_v \sinh(kz)\sin(kx - \omega t) \tag{25}$$

$$\phi = A_w \cos(kx - \omega t) + h - z, \tag{26}$$

where $(u, v, w)^T$ is the fluid velocity vector and the air-water interface in the hydrostatic configuration is assumed to be located at $z = 0$.

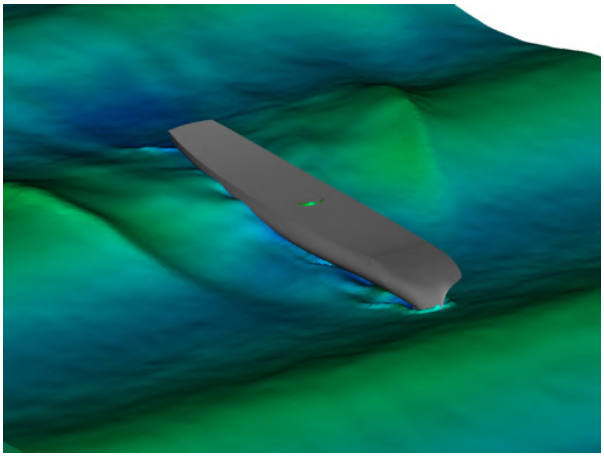
Figure 51 shows the snapshots of the ship negotiating high-amplitude waves. The bottom part of Figure 51 shows the ship partially submerged in water, which is a result of the oncoming wave hitting the bow of the ship. In this case, near the bow, the free surface experiences topological changes, which necessitates the use of an interface-capturing method to handle the air-water interface for this class of problems.

4.8 Vortex-Induced Vibrations of a Bridge Deck

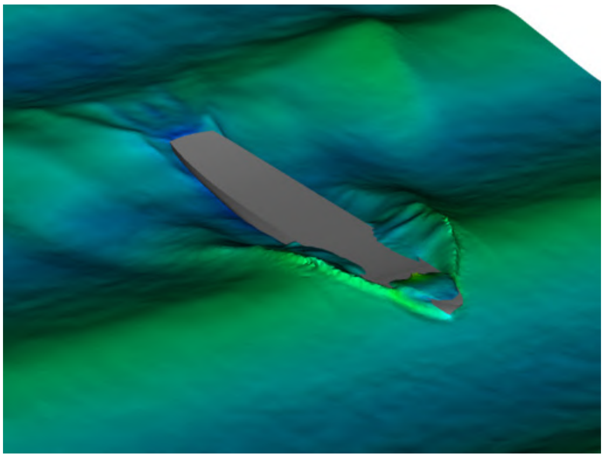
Here we present an example of a fluid-object interaction simulation using a scaled model of the Hardanger bridge deck section [148]. The bridge deck geometric model is shown in Figures 52 and 53. This study was initiated to examine the effect of the guide-vane-like vortex mitigation devices (VMDs) installed on the underside of the bridge deck (see Figure 53 for a zoom on the guide vanes) on the resulting wind aerodynamics and structural response of the bridge. The height h, width b, and length l of the deck scaled model are 0.0666 m, 0.366 m and 1.7 m, respectively.

The bridge deck section computational domain is shown in Figure 54. The locations of the top, bottom, and lateral walls are coincident with those of the wind tunnel where the experiments took place. Note that there is only a 2.5 cm gap between the tunnel wall and the side of the bridge deck section.

All the geometric details of the model-scale bridge deck are modeled in the computations, including the hand and bicycle rails on the top of the deck, and the maintenance rails in the front and rear of the deck. Computations are performed for 2.6 m/s and 6.0 m/s wind speed, with and without the VMDs. Figures 55 and 56



(a) t=9.00 s



b) t=9.50 s

Fig. 51: DTMB 5415 in head waves. Snapshots of the ship negotiating high-amplitude waves. The water surface is colored by the fluid speed.

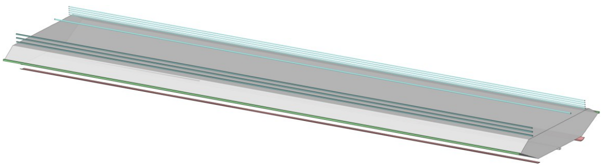


Fig. 52: Geometric model of the scaled Hardanger bridge deck section.

show the mesh resolution used in this study. Boundary-layer prismatic elements are used near all solid surfaces, and tetrahedral elements are used elsewhere in the computational domain. The mesh is refined near the deck and downstream of it to better capture the wake turbulence. The uniform wind speed is prescribed at the inflow boundary, the traction vector is set to zero at the outflow boundary, and the slip condition is set on the top, bot-

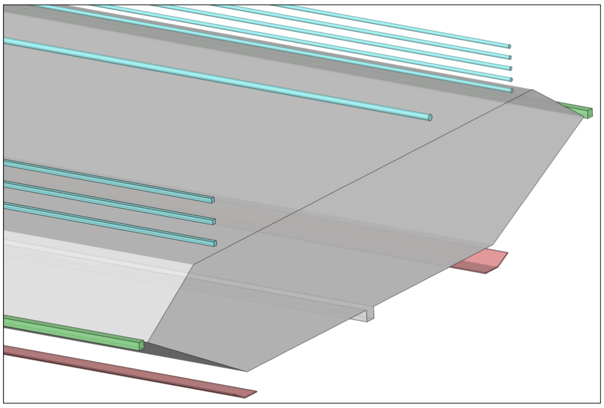


Fig. 53: Zoom on the geometric details of the bridge deck section model. The guide-vane-like vortex mitigation devices are located on the underside of the deck and are shown in light red color.

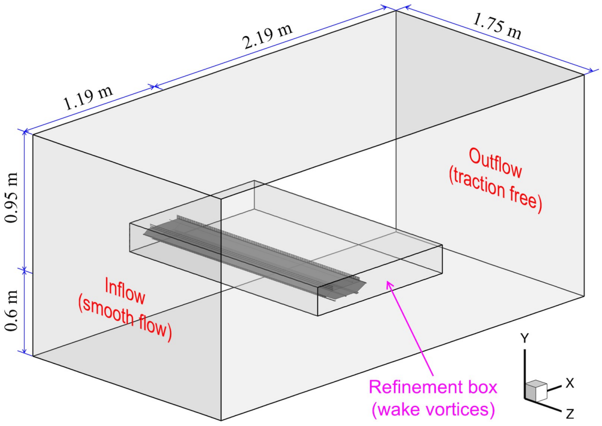


Fig. 54: Bridge deck section computational domain and boundary conditons.

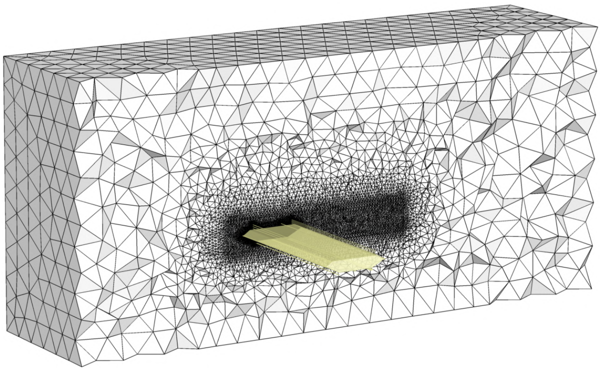


Fig. 55: Aerodynamics mesh of the bridge deck section.

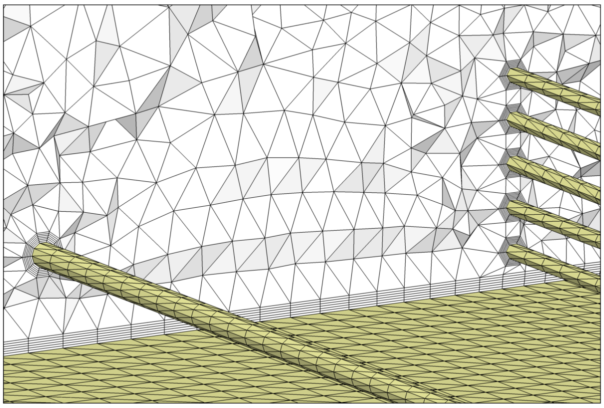


Fig. 56: Zoom on the boundary layer mesh of the top deck and rails.

tom, and lateral boundaries of the computational domain (see Figure 54). The no-slip boundary condition on the bridge deck surface is enforced weakly. The bridge deck is modeled as a rigid object. For the bridge deck mass, moment of inertia tensor, and stiffness and damping matrices (see Eqs. (15) and (16)) the readers are referred to [148]. The deck is allowed to displace vertically, and undergo pitching and rolling motions.

Figure 57 shows drag and lift coefficients for cases with and without the VMDs. The drag and lift coefficients are defined as

$$C_D = \frac{F_D}{\frac{1}{2}\rho U^2 hl}, \tag{27}$$

and

$$C_L = \frac{F_L}{\frac{1}{2}\rho U^2 bl}, \tag{28}$$

where F_D and F_L are the drag and lift forces, respectively, ρ is the air density, and U is the inflow speed. Computational results are compared with the experimental measurements from [148] and reasonable agreement is achieved.

Figure 58 shows the time history of angular displacement of the bridge deck corresponding to the pitching motion. The figure clearly shows that with the added VMDs the bridge deck experiences smaller rotational motions then without, which was also observed in the wind tunnel tests.

To better understand the underlying mechanics, the differences in the air flow with and without VMDs are shown on a planar cut of the bridge deck in Figure 59. The guide vanes keep the flow attached to the underside of the deck, which delays flow separation and precludes formation of large-scale vortical structures that drive the bridge deck response.

Figure 60 shows the 3D view of the deck with guide vanes, where air speed contours at a time instant are plotted on a set of cuts along the deck length. Top and

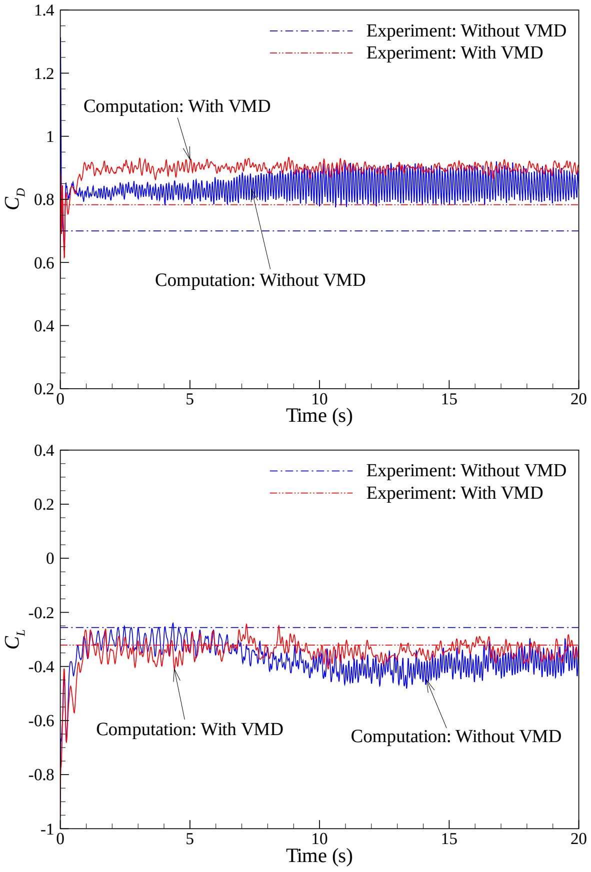


Fig. 57: Time history of the drag (top) and lift (bottom) coefficients for cases with and without VMDs for 2.6 m/s wind speed. Time-averaged experimental measurements from [148] are plotted for comparison.

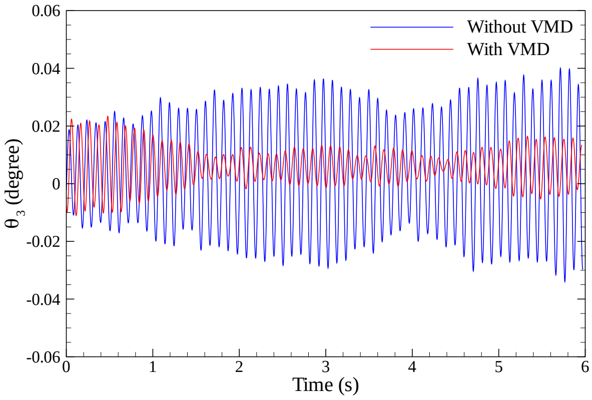


Fig. 58: Time history of the pitching angle for the cases with and without VMDs for 6.0 m/s wind speed.

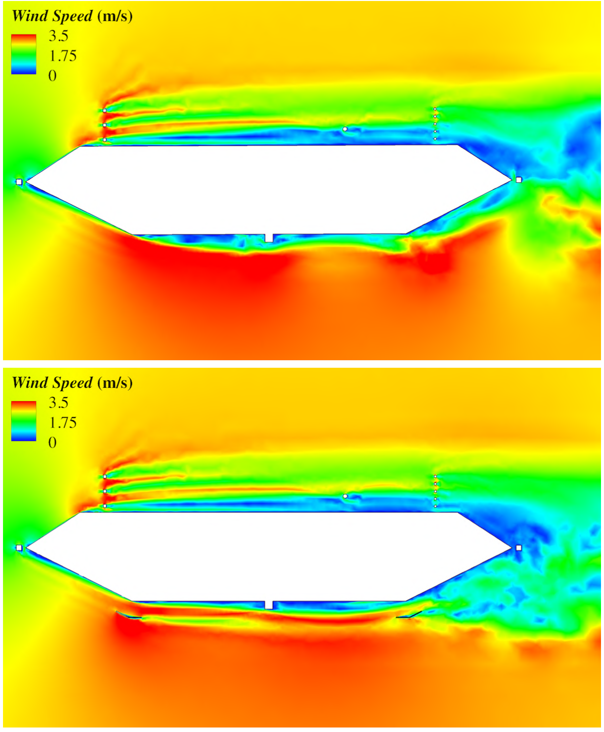


Fig. 59: Instantaneous air speed contours on a planar cut near the bridge deck for 2.6 m/s wind speed. Top: Case without VMDs. Bottom: Case with VMDs.

bottom views are shown. The flow is turbulent and three-dimensional, which underscores the importance of 3D aerodynamics modeling and simulation for this class of problems.

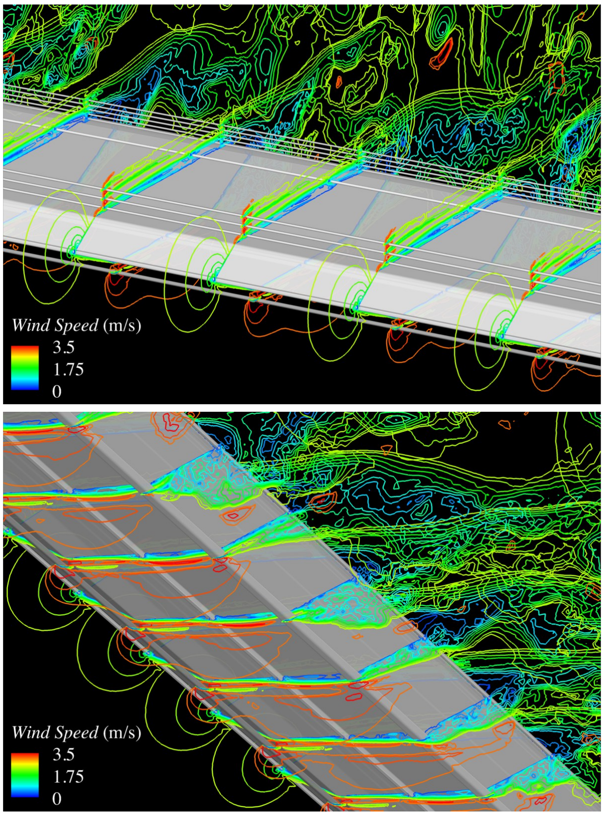


Fig. 60: Instantaneous air speed contours on a set of cuts along the deck length for 2.6 m/s wind speed. Top and bottom deck views are shown.

5 Concluding Remarks

Bringing solution and analysis to specific classes of FSI problems with a common computational technology need motivated the development of our core ALE-VMS and ST FSI techniques, their recent versions, and the special ALE-VMS and ST FSI techniques targeting specific classes of problems, such as parachute FSI, aerodynamics of flapping wings, wind-turbine aerodynamics and FSI, and free-surface flow and FOI for ship hydrodynamics. We presented an overview of the core and special ALE-VMS and ST FSI techniques. We also presented examples of different classes of challenging problems solved: spacecraft parachute FSI, ship hydrodynamics, wind-turbine aerodynamics, patient-specific arterial FSI, aerodynamics of flapping wings of an actual locust and an MAV, and vortex-induced vibrations of a bridge deck section. In some of the problems solved, we included a comparison with the experimental data, and the comparison was always favorable. The examples show that in a diverse set of engineering applications, with the scope and power afforded by the core and special ALE-VMS and ST FSI techniques, we can provide reliable analysis and support the design process.

Acknowledgment

This work was supported in part by NASA JSC grant NNX13AD87G. Method development and evaluation components of the work on aerodynamics of flapping wings and wind-turbine aerodynamics were supported in part by ARO Grant W911NF-12-1-0162 (TT) and Rice–Waseda research agreement (KT). The development and application of FOI techniques for bridge aerodynamics was supported by the program for preferred research areas at the Faculty of Engineering Science and Technology, the Norwegian University of Science and Technology. The research work on free-surface FOI was supported by the ARO Grant W911NF-11-1-0083 (YB). We wish to thank the Texas Advanced Computing Center (TACC) at the University of Texas at Austin, the San Diego Supercomputer Center (SDSC) at the University of California, San Diego, and the Norwegian Metacenter for Computational Science (Notur) for providing some of the HPC resources used. We thank Professor Fabrizio Gabbiani and Dr. Raymond Chan (Baylor College of Medicine) for providing us the digital data extracted from the wind-tunnel videos of the locust.

References

1. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", Advances in Applied Mechanics, **28** (1992) 1–44, doi: 10.1016/S0065-2156(08)70153-4.
2. T.E. Tezduyar, M. Behr, and J. Liou, "A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests", Computer Methods in Applied Mechanics and Engineering, **94** (1992) 339–351, doi: 10.1016/0045-7825(92)90059-S.
3. T.E. Tezduyar, M. Behr, S. Mittal, and J. Liou, "A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders", Computer Methods in Applied Mechanics and Engineering, **94** (1992) 353–371, doi: 10.1016/0045-7825(92)90060-W.
4. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", International Journal for Numerical Methods in Fluids, **43** (2003) 555–575, doi: 10.1002/fld.505.
5. T.E. Tezduyar and S. Sathe, "Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques", International Journal for Numerical Methods in Fluids, **54** (2007) 855–900, doi: 10.1002/fld.1430.
6. K. Takizawa and T.E. Tezduyar, "Multiscale space–time fluid–structure interaction techniques", Computational Mechanics, **48** (2011) 247–267, doi: 10.1007/s00466-011-0571-z.
7. K. Takizawa and T.E. Tezduyar, "Space–time fluid–structure interaction methods", Mathematical Models and Methods in Applied Sciences, **22** (2012) 1230001, doi: 10.1142/S0218202512300013.
8. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, Computational Fluid–Structure Interaction: Methods and Applications. Wiley, February 2013, ISBN 978-0470978771.
9. T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann, "Lagrangian–Eulerian finite element formulation for incompressible viscous flows", Computer Methods in Applied Mechanics and Engineering, **29** (1981) 329–349.
10. R. Ohayon, "Reduced symmetric models for modal analysis of internal structural-acoustic and hydroelastic-sloshing systems", Computer Methods in Applied Mechanics and Engineering, **190** (2001) 3009–3019.
11. E.H. van Brummelen and R. de Borst, "On the nonnormality of subiteration for a fluid-structure interaction problem", SIAM Journal on Scientific Computing, **27** (2005) 599–621.
12. Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes, "Isogeometric fluid–structure interaction analysis with applications to arterial blood flow", Computational Mechanics, **38** (2006) 310–322.
13. R.A. Khurram and A. Masud, "A multiscale/stabilized formulation of the incompressible Navier–Stokes equations for moving boundary flows and fluid–structure interaction", Computational Mechanics, **38** (2006) 403–416.
14. R. Lohner, J.R. Cebral, C. Yang, J.D. Baum, E.L. Mestreau, and O. Soto, "Extending the range of applicability of the loose coupling approach for FSI simulations", in H.-J. Bungartz and M. Schafer, editors, Fluid–Structure Interaction, volume 53 of Lecture Notes in Computational Science and Engineering, 82–100, Springer, 2006.
15. K.-U. Bletzinger, R. Wuchner, and A. Kupzok, "Algorithmic treatment of shells and free form-membranes in FSI", in H.-J. Bungartz and M. Schafer, editors, Fluid–Structure Interaction, volume 53 of Lecture Notes in Computational Science and Engineering, 336–355, Springer, 2006.
16. Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang, "Isogeometric fluid–structure interaction: theory, algorithms, and computations", Computational Mechanics, **43** (2008) 3–37.
17. W.G. Dettmer and D. Peric, "On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction", Computational Mechanics, **43** (2008) 81–90.
18. Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser, and Y. Zhang, "Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device", Computer Methods in Applied Mechanics and Engineering, **198** (2009) 3534–3550.
19. Y. Bazilevs, M.-C. Hsu, D. Benson, S. Sankaran, and A. Marsden, "Computational fluid–structure interaction: Methods and application to a total cavopulmonary connection", Computational Mechanics, **45** (2009) 77–89.
20. R. Calderer and A. Masud, "A multiscale stabilized ALE formulation for incompressible flows with moving boundaries", Computational Mechanics, **46** (2010) 185–197.
21. Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J. Isaksen, "A fully-coupled fluid–structure interaction simulation of cerebral aneurysms", Computational Mechanics, **46** (2010) 3–16.
22. Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen, "Computational fluid–structure interaction: Methods and application to cerebral aneurysms", Biomechanics and Modeling in Mechanobiology, **9** (2010) 481–498.
23. Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T.E. Tezduyar, "3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics", International Journal for Numerical Methods in Fluids, **65** (2011) 207–235, doi: 10.1002/fld.2400.
24. Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger, "3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades", International Journal for Numerical Methods in Fluids, **65** (2011) 236–253.
25. I. Akkerman, Y. Bazilevs, C.E. Kees, and M.W. Farthing, "Isogeometric analysis of free-surface flow", Journal of Computational Physics, **230** (2011) 4137–4152.
26. M.-C. Hsu and Y. Bazilevs, "Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations", Finite Elements in Analysis and Design, **47** (2011) 593–599.
27. S. Nagaoka, Y. Nakabayashi, G. Yagawa, and Y.J. Kim, "Accurate fluid–structure interaction computations using elements without mid-side nodes", Computational Mechanics, **48** (2011) 269–276, doi: 10.1007/s00466-011-0620-7.
28. Y. Bazilevs, M.-C. Hsu, K. Takizawa, and T.E. Tezduyar, "ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction", Mathematical Models and

Methods in Applied Sciences, **22** (2012) 1230002, doi: 10.1142/S0218202512300025.

29. I. Akkerman, Y. Bazilevs, D.J. Benson, M.W. Farthing, and C.E. Kees, “Free-surface flow and fluid–object interaction modeling with emphasis on ship hydrodynamics”, Journal of Applied Mechanics, **79** (2012) 010905.
30. M.-C. Hsu, I. Akkerman, and Y. Bazilevs, “Wind turbine aerodynamics using ALE-VMS: Validation and role of weakly enforced boundary conditions”, Computational Mechanics, **50** (2012) 499–511.
31. M.-C. Hsu and Y. Bazilevs, “Fluid–structure interaction modeling of wind turbines: simulating the full machine”, Computational Mechanics, **50** (2012) 821–833.
32. I. Akkerman, J. Dunaway, J. Kvandal, J. Spinks, and Y. Bazilevs, “Toward free-surface modeling of planing vessels: simulation of the Fridsma hull using ALE-VMS”, Computational Mechanics, **50** (2012) 719–727.
33. S. Minami, H. Kawai, and S. Yoshimura, “Parallel BDD-based monolithic approach for acoustic fluid–structure interaction”, Computational Mechanics, **50** (2012) 707–718.
34. T. Miras, J.-S. Schotte, and R. Ohayon, “Energy approach for static and linearized dynamic studies of elastic structures containing incompressible liquids with capillarity: a theoretical formulation”, Computational Mechanics, **50** (2012) 729–741.
35. T.M. van Opstal, E.H. van Brummelen, R. de Borst, and M.R. Lewis, “A finite-element/boundary-element method for large-displacement fluid–structure interaction”, Computational Mechanics, **50** (2012) 779–788.
36. J.Y. Yao, G.R. Liu, D.A. Narmoneva, R.B. Hinton, and Z.-Q. Zhang, “Immersed smoothed finite element method for fluid–structure interaction simulation of aortic valves”, Computational Mechanics, **50** (2012) 789–804.
37. A. Larese, R. Rossi, E. Onate, and S.R. Idelsohn, “A coupled PFEM–Eulerian approach for the solution of porous FSI problems”, Computational Mechanics, **50** (2012) 805–819.
38. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, “Challenges and directions in computational fluid–structure interaction”, Mathematical Models and Methods in Applied Sciences, **23** (2013) 215–221, doi: 10.1142/S0218202513400010.
39. A. Korobenko, M.-C. Hsu, I. Akkerman, J. Tippmann, and Y. Bazilevs, “Structural mechanics modeling and FSI simulation of wind turbines”, Mathematical Models and Methods in Applied Sciences, **23** (2013) 249–272.
40. J.Y. Yao, G.R. Liu, D. Qian, C.L. Chen, and G.X. Xu, “A moving-mesh gradient smoothing method for compressible CFD problems”, Mathematical Models and Methods in Applied Sciences, **23** (2013) 273–305.
41. K. Kamran, R. Rossi, E. Onate, and S.R. Idelsohn, “A compressible Lagrangian framework for modeling the fluid–structure interaction in the underwater implosion of an aluminum cylinder”, Mathematical Models and Methods in Applied Sciences, **23** (2013) 339–367.
42. M.-C. Hsu, I. Akkerman, and Y. Bazilevs, “Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment”, Wind Energy, **17** (2014) 461–481.
43. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal, “Parallel finite-element computation of 3D flows”, Computer, **26** (1993) 27–36, doi: 10.1109/2.237441.
44. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, and M. Litke, “Flow simulation and high performance computing”, Computational Mechanics, **18** (1996) 397–412, doi: 10.1007/BF00350249.
45. T.E. Tezduyar, “Finite element methods for flow problems with moving boundaries and interfaces”, Archives of Computational Methods in Engineering, **8** (2001) 83–130, doi: 10.1007/BF02897870.
46. J.E. Akin, T.E. Tezduyar, and M. Ungor, “Computation of flow problems with the mixed interface-tracking/interface-capturing technique (MITICT)”, Computers & Fluids, **36** (2007) 2–11, doi: 10.1016/j.compfluid.2005.07.008.
47. S. Mittal and T.E. Tezduyar, “A finite element study of incompressible flows past oscillating cylinders and aerofoils”, International Journal for Numerical Methods in Fluids, **15** (1992) 1073–1118, doi: 10.1002/fld.1650150911.
48. T. Tezduyar and Y. Osawa, “The Multi-Domain Method for computation of the aerodynamics of a parachute crossing the far wake of an aircraft”, Computer Methods in Applied Mechanics and Engineering, **191** (2001) 705–716, doi: 10.1016/S0045-7825(01)00310-3.
49. S. Mittal and T.E. Tezduyar, “Parallel finite element simulation of 3D incompressible flows – Fluid-structure interactions”, International Journal for Numerical Methods in Fluids, **21** (1995) 933–953, doi: 10.1002/fld.1650211011.
50. K. Takizawa, B. Henicke, A. Puntel, T. Spielman, and T.E. Tezduyar, “Space–time computational techniques for the aerodynamics of flapping wings”, Journal of Applied Mechanics, **79** (2012) 010903, doi: 10.1115/1.4005073.
51. K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T.E. Tezduyar, “Space–time techniques for computational aerodynamics modeling of flapping wings of an actual locust”, Computational Mechanics, **50** (2012) 743–760, doi: 10.1007/s00466-012-0759-x.
52. K. Takizawa, N. Kostov, A. Puntel, B. Henicke, and T.E. Tezduyar, “Space–time computational analysis of bio-inspired flapping-wing aerodynamics of a micro aerial vehicle”, Computational Mechanics, **50** (2012) 761–778, doi: 10.1007/s00466-012-0758-y.
53. K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T.E. Tezduyar, “Computer modeling techniques for flapping-wing aerodynamics of a locust”, Computers & Fluids, **85** (2013) 125–134, doi: 10.1016/j.compfluid.2012.11.008.
54. K. Takizawa, B. Henicke, T.E. Tezduyar, M.-C. Hsu, and Y. Bazilevs, “Stabilized space–time computation of wind-turbine rotor aerodynamics”, Computational Mechanics, **48** (2011) 333–344, doi: 10.1007/s00466-011-0589-2.
55. K. Takizawa, B. Henicke, D. Montes, T.E. Tezduyar, M.-C. Hsu, and Y. Bazilevs, “Numerical-performance studies for the stabilized space–time computation of wind-turbine rotor aerodynamics”, Computational Mechanics, **48** (2011) 647–657, doi: 10.1007/s00466-011-0614-5.
56. K. Takizawa, T.E. Tezduyar, S. McIntyre, N. Kostov, R. Kolesar, and C. Habluetzel, “Space–time VMS computation of wind-turbine rotor and tower aerodynamics”, Computational Mechanics, **53** (2014) 1–15, doi: 10.1007/s00466-013-0888-x.
57. S. Takase, K. Kashiyama, S. Tanaka, and T.E. Tezduyar, “Space–time SUPG finite element computation of shallow-water flows with moving shorelines”, Computational Mechanics, **48** (2011) 293–306, doi: 10.1007/s00466-011-0618-1.
58. A.A. Johnson and T.E. Tezduyar, “Simulation of multiple spheres falling in a liquid-filled tube”, Computer Methods in Applied Mechanics and Engineering,

134 (1996) 351–373, doi: 10.1016/0045-7825(95)00988-4.

59. A.A. Johnson and T.E. Tezduyar, "3D simulation of fluid-particle interactions with the number of particles reaching 100", Computer Methods in Applied Mechanics and Engineering, **145** (1997) 301–321, doi: 10.1016/S0045-7825(96)01223-6.
60. A.A. Johnson and T.E. Tezduyar, "Advanced mesh generation and update methods for 3D flow simulations", Computational Mechanics, **23** (1999) 130–143, doi: 10.1007/s004660050393.
61. T.E. Tezduyar, S.K. Aliabadi, M. Behr, and S. Mittal, "Massively parallel finite element simulation of compressible and incompressible flows", Computer Methods in Applied Mechanics and Engineering, **119** (1994) 157–177, doi: 10.1016/0045-7825(94)00082-4.
62. V. Kalro and T.E. Tezduyar, "A parallel 3D computational method for fluid–structure interactions in parachute systems", Computer Methods in Applied Mechanics and Engineering, **190** (2000) 321–332, doi: 10.1016/S0045-7825(00)00204-8.
63. K. Stein, R. Benney, V. Kalro, T.E. Tezduyar, J. Leonard, and M. Accorsi, "Parachute fluid–structure interactions: 3-D Computation", Computer Methods in Applied Mechanics and Engineering, **190** (2000) 373–386, doi: 10.1016/S0045-7825(00)00208-5.
64. T. Tezduyar and Y. Osawa, "Fluid–structure interactions of a parachute crossing the far wake of an aircraft", Computer Methods in Applied Mechanics and Engineering, **191** (2001) 717–726, doi: 10.1016/S0045-7825(01)00311-5.
65. T.E. Tezduyar, S. Sathe, R. Keedy, and K. Stein, "Space–time finite element techniques for computation of fluid–structure interactions", Computer Methods in Applied Mechanics and Engineering, **195** (2006) 2002–2027, doi: 10.1016/j.cma.2004.09.014.
66. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Computer modeling of cardiovascular fluid–structure interactions with the Deforming-Spatial-Domain/Stabilized Space–Time formulation", Computer Methods in Applied Mechanics and Engineering, **195** (2006) 1885–1895, doi: 10.1016/j.cma.2005.05.050.
67. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Fluid–structure interaction modeling of aneurysmal conditions with high and normal blood pressures", Computational Mechanics, **38** (2006) 482–490, doi: 10.1007/s00466-006-0065-6.
68. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", Computers & Fluids, **36** (2007) 191–206, doi: 10.1016/j.compfluid.2005.02.011.
69. T.E. Tezduyar, S. Sathe, T. Cragin, B. Nanna, B.S. Conklin, J. Pausewang, and M. Schwaab, "Modeling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics", International Journal for Numerical Methods in Fluids, **54** (2007) 901–922, doi: 10.1002/fld.1443.
70. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influence of wall elasticity in patient-specific hemodynamic simulations", Computers & Fluids, **36** (2007) 160–168, doi: 10.1016/j.compfluid.2005.07.014.
71. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Numerical investigation of the effect of hypertensive blood pressure on cerebral aneurysm — Dependence of the effect on the aneurysm shape", International Journal for Numerical Methods in Fluids, **54** (2007) 995–1009, doi: 10.1002/fld.1497.
72. M. Manguoglu, A.H. Sameh, T.E. Tezduyar, and S. Sathe, "A nested iterative scheme for computation of incompressible flows in long domains", Computational Mechanics, **43** (2008) 73–80, doi: 10.1007/s00466-008-0276-0.
73. T.E. Tezduyar, S. Sathe, J. Pausewang, M. Schwaab, J. Christopher, and J. Crabtree, "Interface projection techniques for fluid–structure interaction modeling with moving-mesh methods", Computational Mechanics, **43** (2008) 39–49, doi: 10.1007/s00466-008-0261-7.
74. T.E. Tezduyar, S. Sathe, M. Schwaab, J. Pausewang, J. Christopher, and J. Crabtree, "Fluid–structure interaction modeling of ringsail parachutes", Computational Mechanics, **43** (2008) 133–142, doi: 10.1007/s00466-008-0260-8.
75. S. Sathe and T.E. Tezduyar, "Modeling of fluid–structure interactions with the space–time finite elements: Contact problems", Computational Mechanics, **43** (2008) 51–60, doi: 10.1007/s00466-008-0299-6.
76. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Fluid–structure interaction modeling of a patient-specific cerebral aneurysm: Influence of structural modeling", Computational Mechanics, **43** (2008) 151–159, doi: 10.1007/s00466-008-0325-8.
77. T.E. Tezduyar, M. Schwaab, and S. Sathe, "Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique", Computer Methods in Applied Mechanics and Engineering, **198** (2009) 3524–3533, doi: 10.1016/j.cma.2008.05.024.
78. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Fluid–structure interaction modeling of blood flow and cerebral aneurysm: Significance of artery and aneurysm shapes", Computer Methods in Applied Mechanics and Engineering, **198** (2009) 3613–3621, doi: 10.1016/j.cma.2008.08.020.
79. M. Manguoglu, A.H. Sameh, F. Saied, T.E. Tezduyar, and S. Sathe, "Preconditioning techniques for nonsymmetric linear systems in the computation of incompressible flows", Journal of Applied Mechanics, **76** (2009) 021204, doi: 10.1115/1.3059576.
80. K. Takizawa, J. Christopher, T.E. Tezduyar, and S. Sathe, "Space–time finite element computation of arterial fluid–structure interactions with patient-specific data", International Journal for Numerical Methods in Biomedical Engineering, **26** (2010) 101–116, doi: 10.1002/cnm.1241.
81. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Multiscale sequentially-coupled arterial FSI technique", Computational Mechanics, **46** (2010) 17–29, doi: 10.1007/s00466-009-0423-2.
82. K. Takizawa, C. Moorman, S. Wright, J. Christopher, and T.E. Tezduyar, "Wall shear stress calculations in space–time finite element computation of arterial fluid–structure interactions", Computational Mechanics, **46** (2010) 31–41, doi: 10.1007/s00466-009-0425-0.
83. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influence of wall thickness on fluid–structure interaction computations of cerebral aneurysms", International Journal for Numerical Methods in Biomedical Engineering, **26** (2010) 336–347, doi: 10.1002/cnm.1289.
84. M. Manguoglu, K. Takizawa, A.H. Sameh, and T.E. Tezduyar, "Solution of linear systems in arterial fluid mechanics computations with boundary layer mesh refinement", Computational Mechanics, **46** (2010) 83–89, doi: 10.1007/s00466-009-0426-z.
85. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Role of 0D peripheral vasculature model in fluid–structure interaction modeling of

aneurysms", Computational Mechanics, **46** (2010) 43–52, doi: 10.1007/s00466-009-0439-7.

86. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Space–time finite element computation of complex fluid–structure interactions", International Journal for Numerical Methods in Fluids, **64** (2010) 1201–1218, doi: 10.1002/fld.2221.
87. K. Takizawa, C. Moorman, S. Wright, T. Spielman, and T.E. Tezduyar, "Fluid–structure interaction modeling and performance analysis of the Orion spacecraft parachutes", International Journal for Numerical Methods in Fluids, **65** (2011) 271–285, doi: 10.1002/fld.2348.
88. K. Takizawa, C. Moorman, S. Wright, J. Purdue, T. McPhail, P.R. Chen, J. Warren, and T.E. Tezduyar, "Patient-specific arterial fluid–structure interaction modeling of cerebral aneurysms", International Journal for Numerical Methods in Fluids, **65** (2011) 308–323, doi: 10.1002/fld.2360.
89. K. Takizawa, S. Wright, C. Moorman, and T.E. Tezduyar, "Fluid–structure interaction modeling of parachute clusters", International Journal for Numerical Methods in Fluids, **65** (2011) 286–307, doi: 10.1002/fld.2359.
90. M. Manguoglu, K. Takizawa, A.H. Sameh, and T.E. Tezduyar, "Nested and parallel sparse algorithms for arterial fluid mechanics computations with boundary layer mesh refinement", International Journal for Numerical Methods in Fluids, **65** (2011) 135–149, doi: 10.1002/fld.2415.
91. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, "Influencing factors in image-based fluid–structure interaction computation of cerebral aneurysms", International Journal for Numerical Methods in Fluids, **65** (2011) 324–340, doi: 10.1002/fld.2448.
92. T.E. Tezduyar, K. Takizawa, T. Brummer, and P.R. Chen, "Space–time fluid–structure interaction modeling of patient-specific cerebral aneurysms", International Journal for Numerical Methods in Biomedical Engineering, **27** (2011) 1665–1710, doi: 10.1002/cnm.1433.
93. K. Takizawa, T. Spielman, and T.E. Tezduyar, "Space–time FSI modeling and dynamical analysis of spacecraft parachutes and parachute clusters", Computational Mechanics, **48** (2011) 345–364, doi: 10.1007/s00466-011-0590-9.
94. M. Manguoglu, K. Takizawa, A.H. Sameh, and T.E. Tezduyar, "A parallel sparse algorithm targeting arterial fluid mechanics computations", Computational Mechanics, **48** (2011) 377–384, doi: 10.1007/s00466-011-0619-0.
95. K. Takizawa, T. Spielman, C. Moorman, and T.E. Tezduyar, "Fluid–structure interaction modeling of spacecraft parachutes for simulation-based design", Journal of Applied Mechanics, **79** (2012) 010907, doi: 10.1115/1.4005070.
96. K. Takizawa, T. Brummer, T.E. Tezduyar, and P.R. Chen, "A comparative study based on patient-specific fluid–structure interaction modeling of cerebral aneurysms", Journal of Applied Mechanics, **79** (2012) 010908, doi: 10.1115/1.4005071.
97. K. Takizawa and T.E. Tezduyar, "Computational methods for parachute fluid–structure interactions", Archives of Computational Methods in Engineering, **19** (2012) 125–169, doi: 10.1007/s11831-012-9070-4.
98. K. Takizawa, Y. Bazilevs, and T.E. Tezduyar, "Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling", Archives of Computational Methods in Engineering, **19** (2012) 171–225, doi: 10.1007/s11831-012-9071-3.
99. K. Takizawa, M. Fritze, D. Montes, T. Spielman, and T.E. Tezduyar, "Fluid–structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity", Computational Mechanics, **50** (2012) 835–854, doi: 10.1007/s00466-012-0761-3.
100. K. Takizawa, D. Montes, M. Fritze, S. McIntyre, J. Boben, and T.E. Tezduyar, "Methods for FSI modeling of spacecraft parachute dynamics and cover separation", Mathematical Models and Methods in Applied Sciences, **23** (2013) 307–338, doi: 10.1142/S0218202513400058.
101. K. Takizawa and T.E. Tezduyar, "Bringing them down safely", Mechanical Engineering, **134** (2012) 34–37.
102. K. Takizawa, T.E. Tezduyar, J. Boben, N. Kostov, C. Boswell, and A. Buscher, "Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity", Computational Mechanics, **52** (2013) 1351–1364, doi: 10.1007/s00466-013-0880-5.
103. K. Takizawa, T.E. Tezduyar, A. Buscher, and S. Asada, "Space–time interface-tracking with topology change (ST-TC)", Computational Mechanics, **54** (2014) 955–971, doi: 10.1007/s00466-013-0935-7.
104. A.N. Brooks and T.J.R. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", Computer Methods in Applied Mechanics and Engineering, **32** (1982) 199–259.
105. T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih, "Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements", Computer Methods in Applied Mechanics and Engineering, **95** (1992) 221–242, doi: 10.1016/0045-7825(92)90141-6.
106. T.J.R. Hughes, "Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods", Computer Methods in Applied Mechanics and Engineering, **127** (1995) 387–401.
107. T.J.R. Hughes, A.A. Oberai, and L. Mazzei, "Large eddy simulation of turbulent channel flows by the variational multiscale method", Physics of Fluids, **13** (2001) 1784–1799.
108. Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi, "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows", Computer Methods in Applied Mechanics and Engineering, **197** (2007) 173–201.
109. Y. Bazilevs and I. Akkerman, "Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method", Journal of Computational Physics, **229** (2010) 3402–3414.
110. I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and S. Hulshoff, "The role of continuity in residual-based variational multiscale modeling of turbulence", Computational Mechanics, **41** (2008) 371–378.
111. Y. Bazilevs, M.-C. Hsu, J. Kiendl, and D.J. Benson, "A computational procedure for pre-bending of wind turbine blades", International Journal for Numerical Methods in Engineering, **89** (2012) 323–336.
112. A. Korobenko, M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "Aerodynamic simulation of vertical-axis wind turbines", Journal of Applied Mechanics, **81** (2013) 021011, doi: 10.1115/1.4024415.

113. Y. Bazilevs and T.J.R. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics", Computers and Fluids, **36** (2007) 12–26.
114. Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes, "Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes", Computer Methods in Applied Mechanics and Engineering, **199** (2010) 780–790.
115. K. Takizawa, T. Yabe, Y. Tsugawa, T.E. Tezduyar, and H. Mizoe, "Computation of free–surface flows and fluid–object interactions with the CIP method based on adaptive meshless Soroban grids", Computational Mechanics, **40** (2007) 167–183, doi: `10.1007/s00466-006-0093-2`.
116. K. Takizawa, K. Tanizawa, T. Yabe, and T.E. Tezduyar, "Ship hydrodynamics computations with the CIP method based on adaptive Soroban grids", International Journal for Numerical Methods in Fluids, **54** (2007) 1011–1019, doi: `10.1002/fld.1466`.
117. M.A. Cruchaga, D.J. Celentano, and T.E. Tezduyar, "A numerical model based on the Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) for flows with fluid–solid and fluid–fluid interfaces", International Journal for Numerical Methods in Fluids, **54** (2007) 1021–1030, doi: `10.1002/fld.1498`.
118. M. Cruchaga, D. Celentano, and T. Tezduyar, "A moving Lagrangian interface technique for flow computations over fixed meshes", Computer Methods in Applied Mechanics and Engineering, **191** (2001) 525–543, doi: `10.1016/S0045-7825(01)00300-0`.
119. J. Sethian, Level Set Methods and Fast Marching Methods. Cambridge University Press, 1999.
120. C.E. Kees, I. Akkerman, M.W. Farthing, and Y. Bazilevs, "A conservative level set method suitable for variable-order approximations and unstructured meshes", Journal of Computational Physics, **230** (2011) 4536–4558.
121. T.E. Tezduyar, M. Behr, S. Mittal, and A.A. Johnson, "Computation of unsteady incompressible flows with the finite element methods – space–time formulations, iterative strategies and massively parallel implementations", in New Methods in Transient Analysis, PVP-Vol.246/AMD-Vol.143, ASME, New York, (1992) 7–24.
122. A.A. Johnson and T.E. Tezduyar, "Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces", Computer Methods in Applied Mechanics and Engineering, **119** (1994) 73–94, doi: `10.1016/0045-7825(94)00077-8`.
123. T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement", Computer Methods in Applied Mechanics and Engineering, **194** (2005) 4135–4195.
124. Y. Bazilevs and T.J.R. Hughes, "NURBS-based isogeometric analysis for the computation of flows about rotating components", Computational Mechanics, **43** (2008) 143–150.
125. K. Takizawa and T.E. Tezduyar, "Space–time computation techniques with continuous representation in time (ST-C)", Computational Mechanics, **53** (2014) 91–99, doi: `10.1007/s00466-013-0895-y`.
126. T. Belytschko, W.K. Liu, and B. Moran, Nonlinear Finite Elements for Continua and Structures. Wiley, Chichester, 2000.
127. T.J.R. Hughes and J. Winget, "Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis", International Journal for Numerical Methods in Engineering, **15** (1980) 1862–1867.
128. T.E. Tezduyar and M. Senga, "Stabilization and shock-capturing parameters in SUPG formulation of compressible flows", Computer Methods in Applied Mechanics and Engineering, **195** (2006) 1621–1632, doi: `10.1016/j.cma.2005.05.032`.
129. T.E. Tezduyar and M. Senga, "SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing", Computers & Fluids, **36** (2007) 147–159, doi: `10.1016/j.compfluid.2005.07.009`.
130. T.E. Tezduyar, M. Senga, and D. Vicker, "Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing", Computational Mechanics, **38** (2006) 469–481, doi: `10.1007/s00466-005-0025-6`.
131. Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "YZβ discontinuity-capturing for advection-dominated processes with application to arterial drug delivery", International Journal for Numerical Methods in Fluids, **54** (2007) 593–608, doi: `10.1002/fld.1484`.
132. F. Rispoli, R. Saavedra, A. Corsini, and T.E. Tezduyar, "Computation of inviscid compressible flows with the V-SGS stabilization and YZβ shock-capturing", International Journal for Numerical Methods in Fluids, **54** (2007) 695–706, doi: `10.1002/fld.1447`.
133. F. Rispoli, R. Saavedra, F. Menichini, and T.E. Tezduyar, "Computation of inviscid supersonic flows around cylinders and spheres with the V-SGS stabilization and YZβ shock-capturing", Journal of Applied Mechanics, **76** (2009) 021209, doi: `10.1115/1.3057496`.
134. F. Rispoli, A. Corsini, and T.E. Tezduyar, "Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD)", Computers & Fluids, **36** (2007) 121–126, doi: `10.1016/j.compfluid.2005.07.004`.
135. T.E. Tezduyar, S. Ramakrishnan, and S. Sathe, "Stabilized formulations for incompressible flows with thermal coupling", International Journal for Numerical Methods in Fluids, **57** (2008) 1189–1209, doi: `10.1002/fld.1743`.
136. J. Chung and G.M. Hulbert, "A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method", Journal of Applied Mechanics, **60** (1993) 371–75.
137. K.E. Jansen, C.H. Whiting, and G.M. Hulbert, "A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method", Computer Methods in Applied Mechanics and Engineering, **190** (2000) 305–319.
138. T.E. Tezduyar, S. Sathe, and K. Stein, "Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations", Computer Methods in Applied Mechanics and Engineering, **195** (2006) 5743–5753, doi: `10.1016/j.cma.2005.08.023`.
139. M. Cruchaga, D. Celentano, and T. Tezduyar, "Computation of mould filling processes with a moving lagrangian interface technique", Communications in Numerical Methods in Engineering, **18** (2002) 483–493, doi: `10.1002/cnm.506`.
140. M.A. Cruchaga, D.J. Celentano, and T.E. Tezduyar, "Moving-interface computations with the edge-tracked interface locator technique (ETILT)", International Journal for Numerical Methods in Fluids, **47** (2005) 451–469, doi: `10.1002/fld.825`.
141. M.A. Cruchaga, D.J. Celentano, and T.E. Tezduyar, "Collapse of a liquid column: Numerical simulation and experimental validation", Computational Mechanics,

39 (2007) 453–476, doi: `10.1007/s00466-006-0043-z`.

142. K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, and B. Buchner, “A volume-of-fluid based simulation method for wave impact problems”, Journal of Computational Physics, **206** (2005) 363–393.
143. G. Fridsma, “A systematic study of the rough-water performance of planing boats”, Davidson Laboratory Report, **1275** (1968).
144. J. Longo and F. Stern, “Uncertainty assessment for towing tank tests with example for surface combatant DTMB model 5415”, Journal of Ship Research, **49** (2005) 55–68.
145. J. Garcia and E. Oñate, “An unstructured finite element solver for ship hydrodynamics problems”, Journal of Applied Mechanics, **70** (2003).
146. J. Longo, J. Shao, M. Irvine, and F. Stern, “Phase-averaged piv for the nominal wake of a surface ship in regular head waves”, Journal of Fluids Engineering, **129** (2007) 524–541.
147. M.E. McCormick, Ocean Engineering Mechanics. With applications. Cambridge University Press, 2010.
148. Svend Ole Hansen ApS, “The Hardanger bridge: Static and dynamic wind tunnel tests with a section model”, Technical report, Prepared for Norwegian Public Roads Administration, 2006.